KonnectVR: Pioneering an Open-sourced, Educational, Collaborative Virtual Reality Platform

David Anderson[1*], Mark Gill[2]

[1]*Biology Department, St. Cloud Technical & Community College, St. Cloud, MN, 56303, USA*

[2]*Visualization Laboratory, St. Cloud State University, St. Cloud, MN, 56301, USA*

*David.anderson@sctcc.edu

Abstract: Virtual reality (VR) technologies are transforming educational paradigms by enabling highly immersive, interactive simulations that increase engagement and cognition. This paper presents KonnectVR (KVR), an innovative, open-source platform designed to mitigate common barriers in educational VR. Barriers such as high costs, limited content customization, and the absence of real-time assessments. The system, constructed by undergraduate students, facilitates experiential learning, real-time collaboration, and assessment delivery, empowering educators with an unprecedented degree of accessibility and customization. Employing a case study methodology, we provide unique insider perspectives into KVR's architectural design, underlying pedagogical framework, development processes, challenges encountered, and future directions. Findings reveal technical hurdles overcome by undergraduate student teams like VR-specific programming, user interface design, networking, and cybersecurity integration. The analysis uncovers key themes around project-based skill acquisition, problem-solving perseverance, cross-team knowledge gaps, and the benefits of an agile, user-centric approach. Ultimately, KVR demonstrates how emerging technologies and open-sourced solutions can converge to innovate learning, pushing boundaries while serving an egalitarian educational mission. The platform sets the foundation for a new generation of community-driven tools democratizing access to interactive, distributive, collaborative VR experiences that maximize knowledge construction.

Keywords: virtual reality, VR, educational technology, Open Source, collaborative learning, real time assessment, immersive learning

Introduction

The transformative impact of Virtual Reality (VR) technology is being felt across a multitude of industries, from healthcare [1] to real estate [2], from gaming [3] to tourism [4]. Yet, its potential is arguably most transformative in education [5], a realm traditionally fraught with limitations such as rigid curricula, varying learning styles [6], and the lack of real-time assessments [7]. Addressing these limitations, KonnectVR (KVR) emerges as a ground-breaking solution: an open-source, educational, distributive, collaborative VR platform with built-in assessments to revolutionize the very fabric of modern learning experiences.

While VR technologies have been used in higher education since the early 2000s [8], classroom implementation of VR remains a challenge. In addition to the cost, deployment, and maintenance of individual head-mounted displays (HMDs), limited content is available for direct instruction. Existing commercially available VR applications directly focused on delivering higher education content are priced at a premium. Organon is a human anatomy application [9], SimX Patient is a nursing simulation [10], EngageVR is a social VR platform where instructors can develop and hold virtual classes [11], and Virbella is another social VR platform to hold virtual classes [12]. All these applications are limited when considering faculty content and assessments. Estimated costs for one 24-student section range from $3,400 (SimX Patient) to $12,000 (Virbella). As more faculty look to VR for content, these costs scale quickly and become exorbitant. Few colleges or universities, and even fewer students, have funds to support these additional fees.

There is a distinct lack of quality and customizability for more cost-effective VR solutions. Free and open educational content is rarely well-vetted by discipline experts and, therefore, usually unsuitable for educational use. Additionally, most VR applications are made for personal use, so they rarely include assessment tools or collaborative activities. While some applications have assessments, they are seldom customizable without significantly adding to the cost. Finally, few of the currently available VR applications have functionality for instructors to customize content.

Being an open-source platform, KVR democratizes educational technology by offering unprecedented accessibility and customization [13]. Educators, developers, and learners can contribute to the platform's continuous development, thus fostering a community-driven, scalable, adaptable, and ever-evolving approach to education. This feature differentiates KVR from other proprietary educational solutions and sets the stage for limitless pedagogical techniques and content delivery innovations.

KVR is not simply a conceptual vision but a functional platform primed to be community-driven to cohesively integrate immersive educational content, real-time collaboration, and data-backed assessments. With VR's capacity for creating highly realistic simulations, KVR enables instructors to contextualize complex topics interactively and experientially. Imagine a history course where students can virtually explore the streets of Ancient Rome or a chemistry module that allows learners to manipulate atomic structures in real time. These are not mere gimmicks but pedagogical tools designed to enhance cognition and retention.

The platform’s distributive capabilities transcend traditional geographical and institutional boundaries, offering learners worldwide equal opportunities to access high-quality education. KVR's open-source nature further amplifies this by encouraging the contribution of educational modules from diverse cultures and disciplines, thereby enriching the global repository of knowledge and learning methods.

Central to KVR’s mission is the facilitation of real-time collaboration. Traditional learning management systems (LMSs) focus learners on a specific learning flow where the faculty and administrator control the content [14]. While LMS platforms have recently focused on improving user interfaces and incorporating personalized learning, they still promote solitary learning with limited interactivity [15]. In contrast, KVR's collaborative features are designed for collective problem-solving, group experiments, and interactive discourse within a virtual environment. This promotes the acquisition of academic knowledge and the development of essential teamwork and communication skills.

A primary driving force behind KVR’s development has been integrating formative and summative assessments into a VR environment. Real-time evaluation empowers educators to adapt their instructional strategies responsively while students benefit from tailored learning pathways that cater to their unique needs and abilities. Figure 1 visually represents where KVR sits in the educational XR environment.

As we navigate through this article, we will delve into the creation process for KVR. After setting foundational terms, we will begin with the project’s motivational factors behind its conception. We will describe the architectural design structure as well as the project-based learning approach implemented during KVR’s construction. Next, we will explain each team’s contributions to the project, including challenges, successes, and failures. We will finish with KVR’s pedagogical potential and the far-reaching implications of this open-source VR platform.

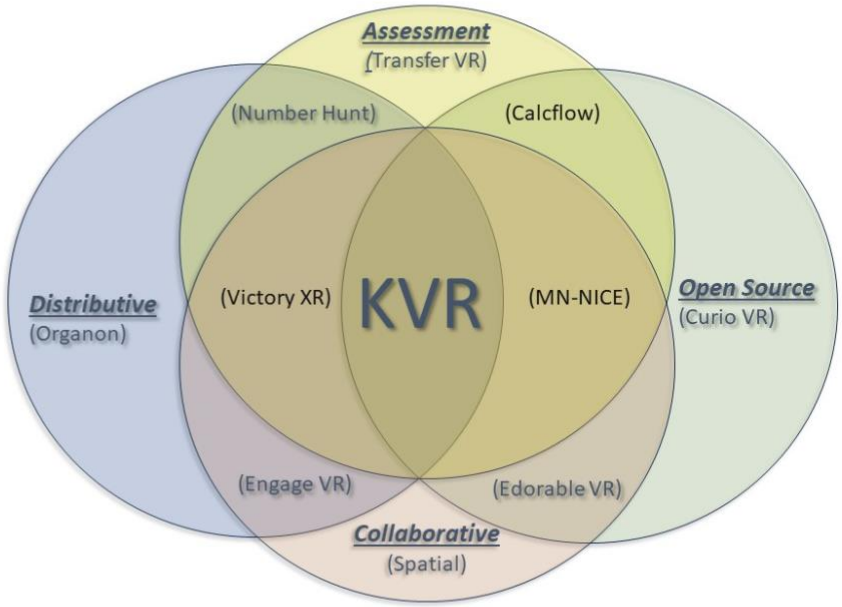


Fig. 1. Ven Diagram showing convergence of Distributive Learning, Collaborative Learning, Assessments, and Open-Source Tools.

Background & Foundation

We will use terminologies and constructs in the following sections, which are fully defined in Supplemental A.

Motivational Factors

1. Democratizing Education. Open education resources (OER) have been increasingly adopted over the past decade, yet a small percentage of textbooks and resources are used in higher education [16]. They are broadly defined as "the open provision of educational resources, enabled by information and communication technologies, for consultation, use, and adaptation by a community of users for non-commercial purposes" [17]. These resources have a significant impact on reducing student financial risk and debt while also improving student course completion [16].

Open source takes OER to a deeper level. Beyond releasing a resource for use free of charge, open source releases the code that also creates the product [13]. Formed by UNIX, GNU, and Linux development, the open-source movement has been closely linked with academia since its inception [13]. To be open source, software must meet the ten requirements relating to author rights, distribution policies, and restrictions set forth by the Open Source Initiative [13]. As an open-source platform, KVR is free to use and provides opportunities for others to extend the platform, creating modules or adding new features.

2. XReality (XR) Educational Technologies. XReality technologies (XR), including VR, are a variety of digitally simulated environments [18] that provide users with a sense of immersion and presence [19]. VR can be both immersive and non-immersive. In immersive VR (IVR), the user is immersed in a digitally simulated environment created by a head-mounted display (HMD), which disconnects them from the physical world [19–22]. In non-immersive VR (NiVR), the digitally simulated environment is displayed on a computer screen and manipulated by a computer mouse or joystick [23, 24].

Two hallmarks of VR are its ability to create a sense of immersion and presence. Immersion relates to the user's disconnection between the real world and the immersive environment, and presence is a sense of "being there" [25–27]. Both immersion and presence directly contribute to the believability and relatability of a virtual situation. IVR has been consistent in its ability to induce physiological and emotional reactions more than screen-based or NiVR learning experiences [28–30]. The unconscious reactions stimulate feelings of immersion and presence, which lead to autobiographical memory encoding like real experiences rather than laboratory memory encoding, which most traditional, screen-based learning experiences (including NiVR) produce [26,27,31]. However, NiVR allows for increased accessibility by using PCs and tablets, which are ubiquitous in education.

Existing technology-enhanced teaching methods often fail to provide the believability and relatability possible using VR, especially IVR. These new, immersive technologies can provide students with collaborative, inquiry-based, problem- and case-based, and discovery learning experiences in visually rich, personalized environments [23, 31–34]. Research has shown that students have significantly higher motivation, satisfaction, and performance scores after IVR experiences [24, 35]. In higher education settings, IVR technology can increase knowledge retention[36, 37], improve abstract and spatial understanding of complex objects such as the heart [38], as well as enhance student motivation and interest in the course material [39, 40]. Additionally, studies have demonstrated that students feel IVR enhances both motivation and enjoyment for learning, as well as finding the course content more engaging [39, 41]. The significance of this aspect lies in previous studies demonstrating that students across various age groups when motivated, tend to achieve higher levels of academic success [42–44]. Similar results on motivation (although not memory encoding) have also been achieved using NiVR [24, 35].

3. Project-based Learning. KVR implemented a project-based learning approach during its construction process, engaging undergraduate computer science and software engineering students to develop the platform. Each year, a new team of students was recruited to add new functions to KVR, building upon previous teams' work. The students met with the stakeholders bi-weekly during the academic year to discuss progress, challenges, and next steps. While the stakeholders gave each student capstone team direction and support, they were also given autonomy to explore potential solutions as they encountered challenges.

Project-based learning (PBL) is increasingly recognized as a pedagogical approach that is well-suited for college-level computer science and software engineering student learning [45, 46]. In traditional lecture-based settings, theoretical concepts are often taught in isolation, limiting students' ability to grasp their application in real-world scenarios. However, PBL promotes the practical application of theory by developing software projects that simulate industry challenges [45, 46]. Students are typically assigned to small groups responsible for conceptualizing, designing, implementing, and testing a software solution to a given problem [45, 46]. This process enables them to gain hands-on experience in various phases of software development, including requirement analysis, system design, coding, testing, and project management [45, 46]. Moreover, students are encouraged to integrate and apply concepts from different areas of their curriculum by engaging in projects that often require multidisciplinary knowledge – such as machine learning applications, mobile app development, or cloud computing solutions.

The collaborative nature of PBL is particularly beneficial in the context of computer science and software engineering, given that teamwork, communication, and cross-functional collaboration are critical skills in the tech industry [47]. Working in groups simulates the collaborative dynamics of real-world software development, thereby giving students a glimpse into what professional work entails, including the need to coordinate with team members, troubleshoot issues collectively, and manage version control. Additionally, the iterative process of PBL, where review and refinement are ongoing, mirrors methodologies commonly used in the software industry, thereby familiarizing students with prevalent best practices [45, 46]. Students are assessed not only based on their technical deliverables but also on their ability to work in teams, communicate their ideas clearly, and adapt to changing requirements or feedback, thus creating a more holistic educational experience beyond mere coding skills.

Collaboration in VR in the context of learning is not a widely implemented concept. While many multiplayer games exist [48], collaborative VR learning experiences are much less common. Social VR use did increase significantly during the COVID-19 pandemic [49]. Yet, these platforms offer little in the way of hands-on learning opportunities. Instead, they aim to create a sense of community in VR. Instructors can use social VR in a remote classroom, but the educational value is limited to verbal interactions. EngageVR does provide opportunities for experiential learning as well as customization and is used by content developers such as VictoryXR [50]. However, as is common with proprietary platforms, EngageVR is priced at a premium, making it cost-prohibitive for most institutions.

Methods

This is a case study undertaken by KVR's primary stakeholders to explore its creation process, architectural nuances, and pedagogical potential. Our unique insider perspective provides insights into the multifaceted challenges and solutions that encapsulated the journey of KVR.

This case study method provides an in-depth, contextually rich examination of KVR, allowing us to unravel the complexities of building an integrated educational platform. We dissect the platform's genesis, from initial conceptualization and motivation to minimum viable product (MVP), exploring the technical decisions and design considerations that guided its development and the pitfalls and setbacks the project has encountered. The aim is to provide insights into the functionalities of KVR and the systemic thought processes and collaborative efforts that contributed to its inception and refinement.

Given our first-hand experience with KVR, our data collection was retrospective and iterative. We first contacted each capstone team and collected the working and final documents they had submitted for their capstone courses. We then reviewed these documents to analyze each team's initial goals versus completed outcomes, barriers encountered, solutions employed, and results. Beyond document analysis, we reviewed different code iterations to identify reusability, compatibility with current versions, and effectiveness.

It is critical to acknowledge the potential for research team bias as the primary stakeholders of KVR. We aimed to mitigate this by continually seeking external reviews and feedback and grounding our analysis in empirical evidence. Furthermore, the dynamic nature of XR technologies signifies continuous evolution of the platform, and as such, future iterations might exhibit differences from the version explored in this study.

Results

KVR was constructed over three years by senior computer science and software engineering students at St. Cloud State University (SCSU). Each team member was enrolled in a capstone course that spanned two standard academic semesters. The capstone course aims to teach students how to manage software engineering projects with real-world stakeholders.

The capstone teams employed a two-week sprint cycle, a component of the Agile methodology designed to accelerate product development and improve responsiveness to stakeholder feedback. In a sprint cycle, the team meets initially to review the backlog, a list of tasks and features prioritized by the stakeholders, and plan the sprint. The students work collaboratively over the next 14 days to develop, test, and iterate software features, aiming for specific deliverables by the sprint's end. The cycle concludes with a sprint review, where developed features are showcased to stakeholders, including the course instructor, who provide feedback [51]. The cycle is repeated, with the team planning new tasks and working to complete them in the next two weeks. Each standard academic semester provides space for up to seven sprints, for a maximum of 14 sprints for each capstone team. Each team completed a total of 12 sprints, five in each fall term and seven each spring.

St. Cloud State University Visualization Laboratory (SCSU VizLab)

Founded in 2014, the SCSU VizLab's primary mission is to explore visualization techniques and how they can be used to improve learning experiences, data analysis, and industry workflows, emphasizing student collaboration, experiential learning, and innovation. The lab actively engages with industry and educational stakeholders, allowing undergraduate students to learn how visualization can be leveraged. It hosts cross-disciplinary projects with local communities and private industries, enriching students' learning experiences in a rapidly evolving field. KVR has been an active project at the VizLab since 2020, providing undergraduate students with opportunities to learn on a real-world project.

2019-2020: Minnesota (State) Networked Immersive Collaborative Experience (MN-NICE)

The spark that launched KVR began in an SCSU VizLab public demonstration of Minnesota (State) Networked Immersive Collaborative Experience (MN-NICE) (Figure 2). As a leader of XR technology construction and integration in the upper Midwest, the SCSU VizLab has held many different demonstrations since its inception to raise awareness of the usefulness of these technologies in education. These have included two workshops related to a grant awarded to SCSU by Minnesota State Colleges and Universities (Minnesota State) called MN-NICE. One demonstration targeted educators interested in XR adoption, and the other was a deliverable specific to the grant award for the public. It was at one of these workshops that Anderson conceptualized KVR.

The Minnesota (State) Networked Immersive Collaborative Experience (MN-NICE) was designed as a collaborative, collocated, educational VR platform where students could engage with virtual environments and their classmates in a shared experience. Virtual objects and settings were mapped to and aligned with the real world, which enabled a blending of the different realities. Unfortunately, because it required people to be in the same physical space, MN-NICE was untenable during the COVID-19 pandemic. The collaborative portions of MN-NICE were repurposed into the first iterations of KVR.

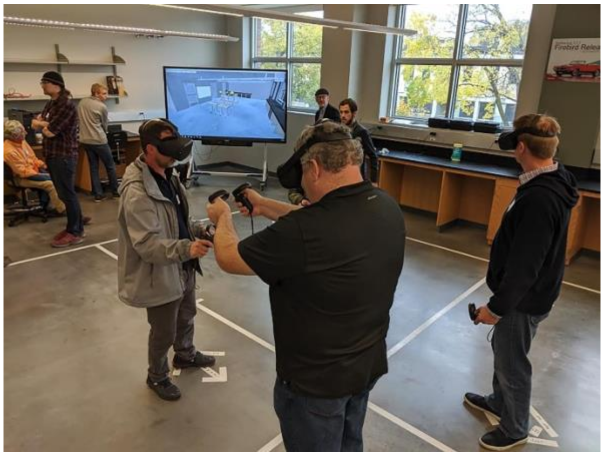


Fig. 2. Public demonstration of MN-NICE in the SCSU VizLab.

2020-21: Integrating OER Anatomy & Physiology Texts with VR

In 2020, Anderson secured another Minnesota State grant award to create OER anatomy & physiology experiences in VR. Anderson and Gill explored options to deploy these laboratory experiences and, after finding none, engaged undergraduate software engineering students to begin developing one. All six members of this first KVR capstone team were an integral part of the development of MN-NICE, so they had experience working with Unity, the primary VR creation software. The team was provided with a scope of work that included 12 requirements (Figure 3). The students chose to break into three sub-teams of two, each tackling one aspect of the requirements, focusing on the VR input system, interaction design, avatars, and multiplayer networking.

User Management	User data for assessment & reporting captured
Networking	Multiple users in shared VR experience
Avatars	Visual representation of users
Modular	Supports 3rd party content
Assessment	Track user progress
Feedback	Notify user of success & progress
Reporting	User data stored
UI/UX	User interface & user experience
Documentation	Hand-off to next team
FERPA compliant	Secure log-in & reporting
Unit testing	Discrete testing of code units
Lab builder	Instructor created experiences

Fig. 3. KVR Scope of Work Requirements

1. VR Input System. Sequencing and connecting objects were identified as primary functions of an educational platform. The students implemented a system so objects can be connected at a given point. Using a skeleton assembly lab as a use case, they created identifying display spheres where objects could be connected that change color based on whether it was correctly or incorrectly connected (Figure 4). They also worked to implement User Interfaces (UIs) in the form of a VR keyboard and tablet (Figure 5). The keyboard was for users to input usernames or question answers, whereas the tablet, which could be free-standing or worn on the arm, served as a system navigation interface.

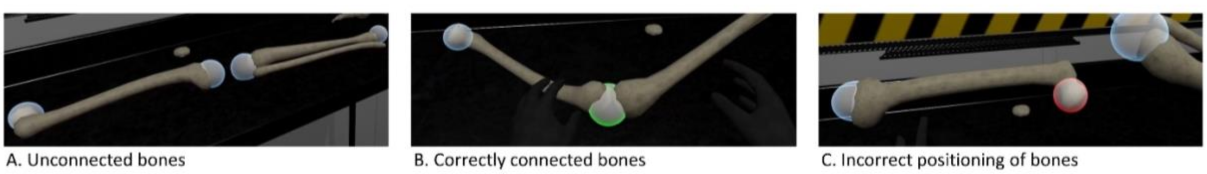


Fig. 4. Bones with spherical connectors at the ends that were used as a use case throughout development: A. Bones are unconnected, and spheres are clear. B. Bones are correctly connected, and spheres glow green. C. Bones are being placed incorrectly, and the sphere glows red.

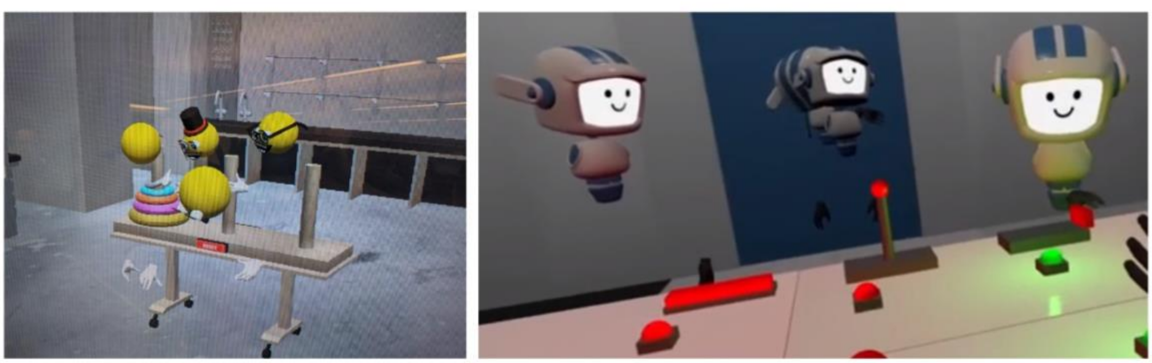
A. Original Tablet B. Keyboard

Fig. 5. Original tablet and keyboard created as user interfaces (UIs) by Team 1.

2. Interaction Design. The students used the first several sprints to brainstorm and plan how to construct the interaction and quiz system, finally deciding the platform would need an online database and cloud server access. While they could begin with student-level Microsoft accounts, they quickly determined more robust access would be required for long-term platform sustainability.

The team encountered significant challenges while connecting the Unity project with the online database. While resolved in the end, it was a significant impediment to progress during the early spring sprints. The performance assessment functionality development proved challenging, as this foundational work needed significant time to explore different avenues and options. By the end of the spring term, Team One had completed an assessment framework. It was exported to a spreadsheet file stored locally on the headset that an instructor could access and download.

3. Avatars. Initially, the work on the avatars was slow due to the delayed procurement of VR headsets. While the students worked on the development, it was difficult to assess the work accurately without access to VR headsets to test. They began with rudimentary avatars that had hands and a sphere as a head. By the spring semester, they improved functionality enough to have facial and hand animations and more realistic bodies. Name tags were created so collaborators could easily identify each other and a camera feature so an individual could see how their avatar presented. They also implemented haptic feedback and sounds when certain actions were performed. The final avatar configuration for Team One was a robot-like figure (Figure 6).



A. Spheres & hands avatars B. Robot avatars

Fig. 6. Team 1 Avatars: A. Initial avatars were only hands and spherical heads. The final avatars were robots with smiley faces.

4. Multiplayer Networking. A primary requirement for the platform was that it had the capacity for several users to interact and collaborate in the same experience. Over the year, the team evaluated several options to incorporate this feature. While Unity had a new solution for multiplayer networking, it was not stable and ready for implementation. The students explored a third-party solution, but determined it was not feasible due to cost and timeline. By the end of the third sprint, they had found a potential solution; however, this system required standing up an online server, which became a significant barrier to overcome over the remainder of the academic year. It was finally solved mid-way through the spring term when the stakeholders could secure an online multiplayer subscription.

Stable networking provided significant challenges to the students throughout the project. At different points in the spring term, they encountered various issues with how objects behaved in a collaborative environment. By the end of spring term, though, the students had resolved enough issues to have four users in the same virtual environment, experiencing the same object actions and outcomes.

5. Challenges & Problem Solving. Over the academic year, two significant challenges outside the team's control became evident. First was the lack of VR headsets available for testing. While the students could test on a flat screen, it does not always accurately display the issues that need to be remedied. Confounding this issue was that the VizLab was shut down due to COVID-19, preventing students from utilizing headsets that would normally have been available. The stakeholders finally provided each team member with a headset by mid-fall. The second significant challenge was robust online server access. Early on in the fall term, the team identified this as a sustainability requirement, but there were significant complexities encountered by the stakeholders securing this access. In the end, the stakeholders navigated the processes and secured the required access by mid-spring.

The students noted two significant challenges within their control over the course of the year: underestimating time to complete tasks and the complexity involved with laying foundational work that others would continue. Beyond seeking stakeholder input, they used mostly online resources to solve problems, including general search engines, YouTube tutorials, documentation found in an online software repository, and Unity Discord communities.

6. Products. The first capstone team was able to complete a functional, collaborative, educational VR platform with built-in assessment capabilities they dubbed KonnectVR (KVR). The platform did have glitches that needed refining, but the foundation was laid for future work. Team One also constructed a *Content Creation Manual* for content creators to understand how to efficiently create and upload modules for the platform.

2021-22: Expanding Access for a VR Learning Platform

The second capstone team was comprised of five members, all of which worked as student workers in the VizLab during the summer of 2021 and so had a working knowledge of KVR and its existing capabilities and shortcomings. As Team Two had five members, it was not as easily split into pairs, so at times they worked individually, at times in pairs, and sometimes as a trio. They regularly noted the importance of comprehensive documentation, which was not always provided by Team One.

Initially, Team Two was to be tasked with creating a two-dimensional interface (2D Client) for KVR so one could access the modules on a standard PC rather than a VR headset. However, one of the team members had taken the initiative over the summer to begin adapting the input and output functions, allowing the scope for Team Two to broaden. The final scope included adding general platform functionality improvement, making avatars more life-like, and expanding capabilities for instructors to create experiences without the need to code.

1. 2D Client. The initial application as a 2D client was ready by mid-fall. However, the stduents found many different areas that needed addressing over the rest of the year with moving to a PC version. General inputs and outputs were straightforward. Integrating grabbing, manipulating, and general hand movements were more complex problems to resolve. Spawning location as well as maintaining object border integrity were also issues the students found and resolved by early spring.

2. Avatar Improvement. Team One created a placeholder avatar with the expectation that future teams would improve upon the design. Team Two implemented a new, more life-like customizable avatar structure that could enhance user agency. They created a new UI that included a color palette selector that changed an avatar's hair, skin, and clothing color (Figure 7). They also created animations for facial expressions and mouth movements when talking. Finally, hand and arm animations were added to indicate when someone was talking.

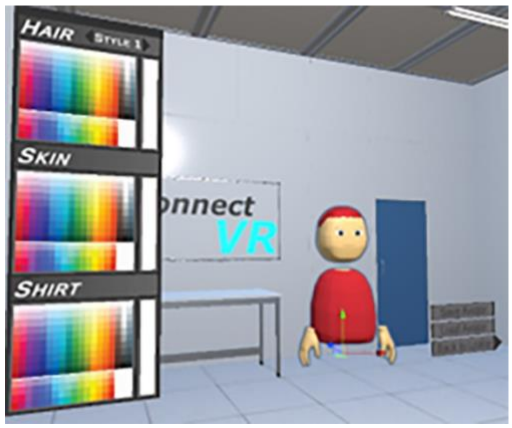


Fig. 7. Customizable Avatar UI: Users can modify their avatar's appearance.

3. Lab Builder. Creating an experience while in a virtual environment added a new degree of complexity to the project, including object manipulation; new UIs; and Upload, Load, Save, Delete, and Search functions for objects and assessments. Each summer, the VizLab engages student interns, which are predominantly SCSU computer science and software engineering students. Expanding on work done by these interns over the summer, Team Two initially focused on creating a process where an instructor could upload, spawn, and manipulate an object while in the VR environment. Object movement was accomplished by selecting the arrow that moved the object in the specific cardinal direction as well as rotation (Figure 8). Additionally, they included a function to resize objects.

The complexity of lab development required a full revision of KVR's existing UIs, which were built more for navigation and basic input. This new UI needed to have a file explorer framework which would allow the user to spawn objects; upload, delete, and search for files; and load, save, and search for lab modules. Additionally, the UI needed options to switch between different modes of operation. To reduce the instructor time burden, the students explored and implemented the concept of asset bundles, which are groupings of objects that can be uploaded into the environment all at once, rather than individually.

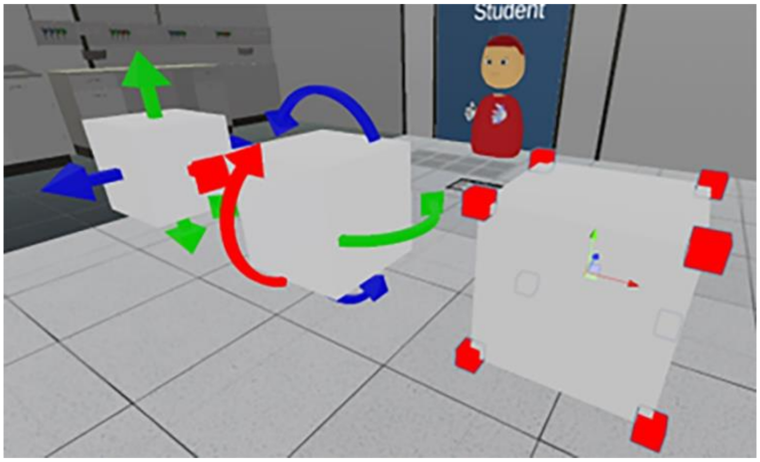


Fig. 8. Lab Builder: The blocks are placeholders for 3D models. The instructor can spawn a model into an environment and manipulate the object using the arrows for location and position as well as the blocks for size.

4. Challenges & Problem Solving. As with Team One, Team Two found unexpected challenges throughout the project, with three seen consistently: version control, understanding complexity, and artistry. Both teams used the free version of GitHub as a version control system. Where Team One did not seem to have issues, Team Two consistently had problems with GitHub losing new versions or creating merging errors. In fact, the team was unable to showcase their work in their final presentation due to breakage when merging everyone's final versions. Understanding just how complex a project will be is challenging for experienced developers, so it is not surprising that senior students would struggle with forecasting how complex a project would be. Finally, all members of Team Two were senior software engineer students that were tasked with creating 3D models, a new avatar, and user interfaces. None of the students considered themselves well versed in art or UI techniques and they all struggled with this aspect. A final challenge for Team Two was implementing networking capabilities. While they searched through KVR's documentation and online resources, they were never able to replicate Team One's networking capabilities.

5. Products. The second team was able to complete their required scope, although some core functionalities (like networking) broke and were not able to be fixed. Modules in KVR could now be experienced in either a VR headset or on a PC and the user could customize their avatar. Additionally, instructors could experiment with creating their own laboratory experiences and not rely solely on developers.

2022-23: Unique User IDs and Security

At the end of the summer of 2022, stakeholder Gill took a sabbatical from the SCSU VizLab to work with Meta Reality Labs (Facebook). This created significant reservations in the next potential capstone team. There were discussions regarding the continuation of the project and in what direction it should move. In the end, the third capstone team moved forward, comprised of four members which all worked as interns in the VizLab during the summer of 2022. With only four members, Team Three loosely split into two pairs that changed based on who had most confidence tackling specific tasks.

Team Three was tasked with implementing a secure log-in system that would be compatible with learning management systems (LMSs), stabilizing the networking capabilities, and improving the avatar customization feature. To begin the year, stakeholder Anderson connected the students with the Minnesota State College and Universities' IT Department (MN State) to provide the team with insights into how an LMS sends and receives secure data. While the process was in transition at the time, the students learned that the new standard for software to connect with an LMS was to implement Learning Tools Interoperability Core Specification 1.3

(LTI 1.3). MN State IT also created an LMS shell so the development team could test processes against a live course.

1. LMS Cybersecurity. To secure data packets over the Internet, public and private key cryptography is employed, where a public key is used to encrypt data, making it accessible only to the holder of the matched private key, who can decrypt it (Figure 9). In LTI 1.3, these public and private keys are pivotal for establishing trust between the tool and the platform. The tool's public key is shared with the LMS, which uses it to verify messages from the tool signed by its private key. Conversely, the LMS has a private key to sign messages and a public key that the tool uses to verify. This ensures secure message exchange.

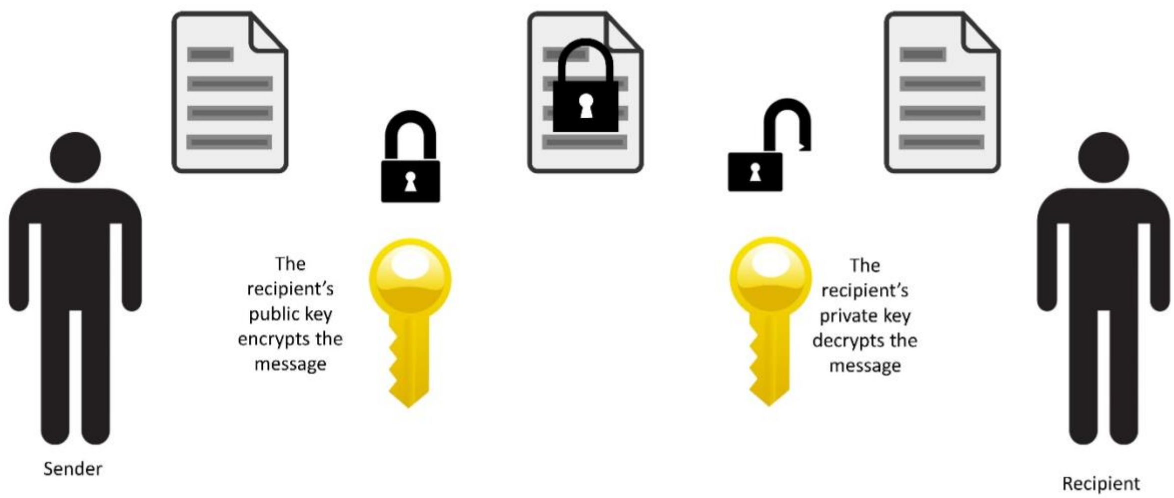


Fig. 9. Cybersecurity Basics: The sender encrypts the data with a public key, and the recipient decrypts the data with a matched private key.

Implementing a secure log-in system that was LTI 1.3 compliant was by far the most challenging task for Team Three. As none had cybersecurity training, they needed to learn the basic concepts and apply them in a new arena. While VR games have secure log-in capabilities, this is handled differently than LTI 1.3 requires. At the beginning of the term, the stakeholders were unaware of any VR application that had successfully implemented LTI 1.3.

Over the year, Team Three was able to successfully implement a log-in system that securely sent information to an online database and received verification of whether that information matched a user account. When a user enters KVR, they land in a lobby scene. Team Three created a UI requiring users to enter an ID, password, and access code to move into any module. If the user does not have an account, they can select "Create an account" and do so. Theyalso successfully implemented a password recovery system that sends an email to a user if requested so they can reset their password. Additionally, Team Three added error messages that display incorrect field inputs (Figure 10) and a reporting function that sends results to an instructor via email tied to the access code and includes the student ID.

Fig. 10. Secure Log-in: UI requiring User ID, Password, and Access Code with password error message.

2. Networking. The team restored basic networking functionality and fixed issues with how the tablet UI displayed when in a collaborative environment.

3. Avatars. Over the summer, VizLab gained access to a Unity package called *AutoHands*, which provides for easy implementation of various interactions. Team Three succeeded in replacing the traditional Unity commands with *AutoHands* to enhance the user experience and reduce the burdens on module creators.

4. UI/UX. While not in the scope of Team Three, they created a whiteboard that allows the user to write in a variety of colors and highlighters in the VR space.

5. Unit Testing. Unit tests verify individual software components in isolation, ensuring functionality and reliability. Essential to software development, they aid in maintaining code, reducing bugs, and facilitating test-driven development. These tests act as checks and documentation, improving design and long-term maintenance. Through Team Three's fall term, no one had constructed any unit tests for KVR. Once Team Three understood the importance of unit testing, especially to cybersecurity, they began to implement a unit testing process to validate the software design.

6. Challenges & Problem Solving. Team Three's scope significantly differed from the previous two, moving squarely into cybersecurity and LMS integration. As stated, none of the students had experience with this aspect of software engineering, which significantly slowed progress. Although MN State IT helped by setting up a course shell and providing input, they were not versed on how to create a new system that implemented LTI 1.3, only how to determine whether a system met the standards after being built. The students reached out to SCSU faculty but received little relevant help. After fall, stakeholder Anderson connected the students with a former MN State IT cybersecurity specialist who was able to help identify issues and potential solutions.

Working in both immersive and 2D environments also creates challenges. As a game development engine, Unity is not built with tools to easily implement standards such as LTI 1.3. Where standard desktop operating systems provide a wide array of tools (e.g., LTI 1.3), immersive environments do not. The students found no resources to guide them in implementing LTI 1.3 with Unity, and the resources were tangential systems that would not work in both environments. The team worked diligently to find online resources and tested several iterations, but in the end, KVR was not quite ready for LMS integration.

While Team Three did have some challenges with revision control like Teams One and Two, the project was moved to Unity's PlasticSCM revision control system over the summer, which significantly decreased conflict issues.

7. Products. As the smallest team with one significant challenge, Team Three had fewer deliverables than Teams One or Two. By the end of the year, the structure to implement LTI 1.3 was in place, including a log-in UI connected to an online database that could verify the validity of log-in credentials. KVR also now sends results via email to an instructor rather than having to access the file directly on the headset. Scope creep involved the team developing a functional whiteboard and implementing the *AutoHands* feature. Team Three also created a unit test framework for future teams to implement.

Table 1 summarizes the efforts and accomplishments of each team over the three academic years inspected.

Table 1. KVR Scope of Work Requirements and Team Completion

Requirement	Team 1 2020-21	Team 2 2021-22	Team 3 2022-23
User Management	Implemented		
Networking	Implemented		Stabilized
Avatars	Implemented	Improved	Improved
Modular	Implemented		Improved
Assessment	Implemented	Extended	
Feedback	Implemented		
Reporting	Implemented		Extended
UI/UX	Implemented	Extended	Extended
Documentation	Minimal	Extended	Extended
FERPA compliant			Initiated
Unit testing			Implemented
Lab Builder		Implemented	

Discussion

The discussion and analysis of creating the KVR platform over three academic years reveals a trajectory of increasing complexity and enhancement of educational technology. Several key themes emerged regarding the student learning and development process:

Collaborative Development Process

The capstone teams exemplify collaborative project-based learning, working in small groups with distinct responsibilities to construct components of KVR. This cooperative approach mirrors real-world engineering teams and enables skill-building in communication, task coordination, and teamwork. However, collaboration was sometimes hindered by documentation gaps between teams. More robust cross-team knowledge sharing could improve project continuity.

Overcoming Technical Complexity

The students faced substantial technical challenges inherent in crafting an innovative VR platform, including networking, UI/UX design, VR-specific development, and cybersecurity integration. Students gained first-hand experience grappling with complex systems. Each team persisted in problem-solving, seeking creative solutions through self-directed research and stakeholder support. However, underestimating task complexity frequently leads to delays.

Bridging Theory and Practice

The project provided an authentic context for students to bridge classroom knowledge with practical application. Students combined their expertise in areas like databases, software design, and security to construct a functional product. This reinforced technical learning and preparedness for industry. However, some skill gaps like cybersecurity and UI design were apparent; targeted supplemental training could be valuable.

Agile, Responsive Development

The scope evolved over the three years based on accomplishments, challenges, and emerging priorities. This agility allowed the platform to expand meaningfully each year. However, changing goals also caused fragmentation. Explicitly mapping long-term objectives at project onset may have improved continuity. Formal user testing and feedback loops could have also helped guide responsive development.

Future Work

The KVR platform, while functional at the end of each academic year, still requires significant work before it is ready for release and widespread adoption. Capstone Team Four is currently advancing the platform forward. In light of the experiences of past capstone teams, future work should focus on several key areas to fully realize the platform's potential.

- Enhanced documentation and knowledge transfer. It is imperative to establish a robust knowledge transfer system to mitigate the collaboration issues due to documentation gaps. This system would ensure team continuity and preserve institutional knowledge for future developers.
- User-centric development. Moving forward, incorporating regular user testing and feedback will be crucial. This will help create a responsive development environment where user experience drives the platform's evolution.
- Advancing LMS integration. Given the complexities of integrating with learning management systems, especially with LTI 1.3, further work should focus on achieving seamless interoperability and ensuring the platform meets educational standards for data security and privacy.
- Unit testing and quality assurance. Establishing a comprehensive unit testing framework will be vital for future development. This framework will guarantee that new features function as intended and maintain the overall integrity of the platform.
- Scalability and performance optimization. As the platform grows, optimizing performance and ensuring that KVR can scale to support an increasing number of users and complex simulations will be important.
- Enhanced accessibility. VR both restricts and expands user access to learning experiences. While a 2D client is a significant advancement in ensuring all students can access the content, exploring additional accessibility challenges like voice interaction and screen-reading functionalities will create an even larger educational impact of KVR.

Conclusion

Overall, KVR demonstrates how student-led project teams can construct complex software systems collaboratively, learning immensely in the process. It points to project-based learning's capacity for instilling professional skills and complementary knowledge. While challenges emerged, they also offered formative experiences in analytic troubleshooting and perseverance. With enhancements to cross-team continuity, intentionally responsive design, and supplemental training on emerging skills, this instructional approach shows immense potential to prepare future technologists.

The construction of the KVR platform presents a multifaceted narrative of growth, challenge, and learning, showcasing the potent blend of collaboration, technical acumen, and problem-solving capabilities fostered through project-based learning. The platform is a testament to the efficacy of hands-on educational experiences in preparing students for real-world technology landscapes.

By converging the domains of immersive technologies and open-sourced educational solutions, KVR provides educators with learning experiences that are interactive, collaborative, and accessible. While the journey was fraught with hurdles, each was a steppingstone towards significant learning outcomes, culminating in a platform that serves educational purposes and empowers its creators with a profound understanding of the software development lifecycle. This case study provides unique insider perspectives into the realities of crafting an ambitious platform like KVR. Future enhancements, guided by the lessons learned, are set to propel KVR into a new echelon of educational technology tools, bridging the gap between theoretical knowledge and practical applications and molding adept future technologists equipped to navigate and shape the technological frontiers of tomorrow.

Acknowledgements. This work was supported by the Minnesota State System of Colleges and Universities Innovation Funding Program and the National Science Foundation (NSF) under awards #2202206 and #1902574.

Disclosures. The authors declare no conflicts of interest.

References

[1] M. M. T. E. Kouijzer, H. Kip, Y. H. A. Bouman, and S. M. Kelders, "Implementation of virtual reality in healthcare: a scoping review on the implementation process of virtual reality in various healthcare settings," Implementation Science Communications **4**, 67 (2023).

[2] G. Pleyers and I. Poncin, "Non-immersive virtual reality technologies in real estate: How customer experience drives attitudes toward properties and the service provider," Journal of Retailing and Consumer Services **57**, 102175 (2020).

[3] F. Wallace, "VR Gaming and the Technology's Impact on the World," HeadStuff (2018). Accessed 8/31/2023.

[4] M. Melo, H. Coelho, G. Gonçalves, N. Losada, F. Jorge, M. S. Teixeira, and M. Bessa, "Immersive multisensory virtual reality technologies for virtual tourism," Multimedia Systems **28**, 1027–1037 (2022).

[5] U. Dakeev and F. Yildiz, "Design and Development of Mixed Reality (MR) Laboratory Tools to Improve Spatial Cognition, Student Engagement, and Employee Safety," (2022).

[6] R. Gentry, A. P. Sallie, and C. A. Sanders, *Differentiated Instructional Strategies to Accommodate Students with Varying Needs and Learning Styles* (2013).

[7] E. Gustafsson-Wright, S. Osborne, and A. Sharma, "How can real-time performance data lead to better education outcomes?," Brookings (2021).

[8] E. Drake-Bridges, A. Strelzoff, and T. Sulbaran, "Teaching Marketing Through a Micro-Economy in Virtual Reality," Journal of Marketing Education **33**, 295–311 (2011).

[9] "The Leading XR Medical Anatomy Platform," https://www.3dorganon.com/. Accessed 8/31/2023.

[10] "SimX Virtual Reality Medical Simulation," https://www.simxvr.com/. Accessed 8/31/2023.

[11] "EngageVR," https://engagevr.io/. Accessed 8/31/2023.

[12] "Virbela: A Virtual World for Work, Education & Events," https://www.virbela.com/. Accessed 8/31/2023.

[13] S. E. Lakhan and K. Jhunjhunwala, "Open Source Software in Education," https://er.educause.edu/articles/2008/5/open-source-software-in-education. Accessed 8/31/2023.

[14] A. Terehin, "How to Create a Learning Experience Platform [LMS to LXP] | Agente," (2021). Accessed 10/3/2023.

[15] A. Terehin, "How to Design and Redesign a LMS," (2021). Accessed 10/3/2023.

[16] N. B. Colvard, C. E. Watson, and H. Park, "The impact of open educational resources on various student success metrics," International Journal of Teaching and Learning in Higher Education **30**, 262–276 (2018).

[17] *Forum on the Impact of Open Courseware for Higher Education in Developing Countries. Final Report.* (2002).

[18] P. A. Rauschnabel, R. Felix, C. Hinsch, H. Shahab, and F. Alt, "What is XR? Towards a Framework for Augmented and Virtual Reality," Computers in Human Behavior **133**, 107289 (2022).

[19] P. Rubin, *Future Presence: How Virtual Reality Is Changing Human Connection, Intimacy, and the Limits of Ordinary Life* (HARPERONE, 2020).

[20] M. J. W. Lee, M. Georgieva, E. Craig, B. Alexander, and J. Richter, "State of XR & immersive learning outlook report 2021," (2021).

[21] A. K. Noor, "The Hololens revolution," Mechanical Engineering **138**, 30–35 (2016).

[22] S. Lessick and M. Kraft, "Facing reality: The growth of virtual reality and health sciences libraries," jmla **105**, (2017).

[23] D. F. Angel-Urdinola, C. Castillo-Castro, and A. Hoyos, *Meta-Analysis Assessing the Effects of Virtual Reality Training on Student Learning and Skills Development* (World Bank, 2021).

[24] N. R. Dyrberg, A. H. Treusch, and C. Wiegand, "Virtual laboratories in science education: Students' motivation and experiences in two tertiary biology courses," Journal of Biological Education **51**, 358–374 (2017).

[25] J. J. Cummings and J. N. Bailenson, "How immersive is enough? A meta-analysis of the effect of immersive technology on user presence," Media Psychology **19**, 272–309 (2016).

[26] A. Suh and J. Prophet, "The state of immersive technology research: A literature analysis," Computers in Human Behavior **86**, 77–90 (2018).

[27] J. Parong and R. E. Mayer, "Cognitive and affective processes for learning science in immersive virtual reality," J Comput Assist Learn **37**, 226–241 (2021).

[28] J. Kisker, T. Gruber, and B. Schöne, "Behavioral realism and lifelike psychophysiological responses in virtual reality by the example of a height exposure," Psychological Research **85**, 68–81 (2021).

[29] L. B. Cadet and H. Chainay, "Memory of virtual experiences: Role of immersion, emotion and sense of presence," International Journal of Human-Computer Studies **144**, 102506 (2020).

[30] G. A. Wilson and M. McGill, "Violent video games in virtual reality: Re-evaluating the impact and rating of interactive experiences," CHI PLAY (2018).

[31] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda," Computers & Education **147**, 103778 (2020).

[32] B. Wu, X. Yu, and X. Gu, "Effectiveness of immersive virtual reality using head-mounted displays on learning performance: A meta-analysis," British Journal of Educational Technology **51**, 1991–2005 (2020).

[33] Y. Zhao and K. A. Frank, "Factors affecting technology uses in schools: An ecological perspective," American Educational Research Journal **40**, 807–840 (2003).

[34] H. Luo, G. Li, Q. Feng, Y. Yang, and M. Zuo, "Virtual reality in K-12 and higher education: A systematic review of the literature from 2000 to 2019," Journal of Computer Assisted Learning **37**, 887–901 (2021).

[35] C. Koh, H. S. Tan, K. C. Tan, L. Fang, F. M. Fong, D. Kan, S. L. Lye, and M. L. Wee, "Investigating the effect of 3D simulation based learning on the motivation and performance of engineering students," Journal of Engineering Education **99**, 237–251 (2010).

[36] K.-T. Huang, C. Ball, J. Francis, R. Ratan, J. Boumis, and J. Fordham, "Augmented Versus Virtual Reality in Education: An Exploratory Study Examining Science Knowledge Retention When Using Augmented Reality/Virtual Reality Mobile Applications," Cyberpsychology, Behavior, and Social Networking **22**, 105–110 (2019).

[37] E. Krokos, C. Plaisant, and A. Varshney, "Virtual memory palaces: immersion aids recall," Virtual Reality **23**, 1–15 (2019).

[38] H.S. Maresky, A. Oikonomou, I. Ali, N. Ditkofsky, M. Pakkal, and B. Ballyk, "Virtual reality and cardiac anatomy: Exploring immersive three-dimensional cardiac imaging, a pilot study in undergraduate medical anatomy education," Clin. Anat. **32**, 238–243 (2019).

[39] S. Kavanagh, A. Luxton-Reilly, B. Wuensche, and B. Plimmer, "A systematic review of Virtual Reality in education," Themes in Science and Technology Education **10**, 85–119 (2017).

[40] G. Makransky and L. Lilleholt, "A structural equation modeling investigation of the emotional value of immersive virtual reality in education," Education Tech Research Dev **66**, 1141–1164 (2018).

[41] H.-M. Huang, U. Rauch, and S.-S. Liaw, "Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach," Computers & Education **55**, 1171–1182 (2010).

[42] M. Boekaerts, "Successful Schooling," in *Motivation to Learn*, D. Bhaskara Rao, ed. (Discovery Publishing House, 2002), pp. 101–120.

[43] J. Hattie, *Visible Learning*, 0 ed. (Routledge, 2008).

[44] Y.-G. Lin, W. J. McKeachie, and Y. C. Kim, "College student intrinsic and/or extrinsic motivation and learning," Learning and Individual Differences **13**, 251–258 (2003).

[45] E. Ceh-Varela, C. Canto-Bonilla, and D. Duni, "Application of Project-Based Learning to a Software Engineering course in a hybrid class environment," Information and Software Technology **158**, 107189 (2023).

[46] W. Intayoad, "PBL Framework for Enhancing Software Development Skills: An Empirical Study for Information Technology Students," Wireless Personal Communications **76**, 419–433 (2014).

[47] "51 In-Demand Tech Skills for Technology Careers | Indeed.com," https://www.indeed.com/career-advice/career-development/in-demand-tech-skills. Accessed 9/1/2023.

[48] "Steam Store," https://store.steampowered.com/. Accessed 8/31/2023.

[49] B. Kelley, "The Rise of the 'Quarantine Bar Simulator': The Uses and Gratifications of Social VR During the COVID-19 Pandemic," in *2021 4th International Conference on Information and Computer Technologies (ICICT)* (2021), pp. 216–221.

[50] "Home Virtual Reality VR Education Software & Augmented Reality Learning - VictoryXR," https://www.victoryxr.com/. Accessed 8/31/2023.

[51] M. Frydenberg, D. Yates, and J. Kukesh, "Sprint, then Fly: Teaching Agile Methodologies with Paper Airplanes," Information Systems Education Journal **16**, 22 (2018).