International Journal of Semantic Computing
Vol. 13, No. 3 (2019) 343–366
©World Scientific Publishing Company
DOI: 10.1142/S1793351X19400154

World Scientific
www.worldscientific.com

Learners' Technological Acceptance of VR Content Development: A Sequential 3-Part Use Case Study of Diverse Post-Secondary Students

Vinh T. Nguyen[*,‡], Rebecca Hite[†,§] and Tommy Dang[*,¶]

[*]*Computer Science Department, Texas Tech University*
Lubbock, TX 79409, USA

[†]*Department of Curriculum and Instruction, Texas Tech University*
Lubbock, TX 79409, USA
[‡]*vinh.nguyen@ttu.edu*
[§]*rebecca.hite@ttu.edu*
[¶]*tommy.dang@ttu.edu*

Web-based virtual reality (VR) development tools are in ubiquitous use by software developers, and now, university (undergraduate) students, to move beyond using, to creating new and energizing VR content. Web-based VR (WebVR), among other libraries and frameworks, have risen as a low-cost platform for users to create rich and intuitive VR content and applications. However, the success of WebVR as an instructional tool relies on post-secondary students technological acceptance (TA), the intersectionality of a user's perceived utility (PU) and perceived ease of use (PEOU, or convenience) with said technological tool. Yet, there is a dearth of exploratory studies of students' experiences with the AR/VR development technologies to infer their TA. To ascertain the viability of WebVR tools for software engineering undergraduates in the classroom, this paper presents a 3-case contextual investigation of 38 undergraduate students tasked with creating VR content. In each use case, students were provided increasing freedom in their VR content development parameters. Results indicated that students demonstrated elements of technological acceptance in their selection of webVR and other platforms, and not only successfully creating rich and robust VR content (PU), but also executing these projects in a short period (PEOU). Other positive externalities observed were students exhibitions of soft skills (e.g. creativity, critical thinking) and different modes of demonstrating coding knowledge, which suggest further study. Discussed are the lessons learned from the WebVR and VR/AR interventions and recommendations for WebVR instruction. This work may be helpful for both learners and teachers using VR/AR in selecting, designing, and developing coursework materials, tools, and libraries.

Keywords: A-frame; computer science course design; soft skills; technology acceptance model; undergraduate education; web-based virtual reality.

1. Introduction

Virtual reality (VR) and augmented reality (AR) have become two universally realized technological advancements, utilized over an assortment of learning spaces: exploration [1], medicine [2, 3], business [4], design [5] and the military [6]. What makes these technologies unique is the extent to which the user has control over the environment, for example, *the Camry Web Configurator*[a] won the Visionary Award at the Summit Awards for Emerging Media for flexible car configuration using VR (WebVR) technology. One may infer from the wide variety of VR/AR applications, and this award, that AR/VR technologies allow users to move from a *user* perspective to a *designer/creator* perspective. Thus, there are many studies examining VR/AR from a design point of view, yet mainly focus on the software development of 3D models (code) rather than the utility or ease of use of those models from the user's perspective. In recent online discussions with Earth and Environmental Sciences faculty at California State University, instructors have pointed out that the biggest difficulty in conveying knowledge in this new age of technology is the lack of skilled users to create 3D contents, as well as best practice (recommendations) for teaching and sharing VR/AR content online. Limitations stymie this development of best practice, especially in how to teach new learners how to use AR/VR tools; instruction strategies for building content with extant AR/VR tools; a full understanding of the hardware, software affordances of said tools for those learners; and teaching how to use and create with these tools within a limited time frame like most instructional (classroom) spaces (e.g. within a week, month, or semester).

The issues identified above raise new questions: What VR/AR development framework/library (tool) could meet these needs in user ease of use with little hardware reliance to keep costs low? If such a library exists, what is its perceived utility so that it may meet users' expectations? What are the pros and cons of the use of such existing libraries? To solve a more complex problem, is the library prioritized for use by the user not only being a "user" but rather a "designer/creator?" Addressing these questions plays an essential role in improving the quality of teaching and learning for instructors who wish to train the next generation of VR/AR content developers, especially as designers/creators. Psotka [7] has said, "Teachers and trainers need to be exposed to VR in multiple ways so that they can begin preparing themselves and their institutions for future changes" and "... to give educators ..a better view of the strengths and weaknesses of these environments; ... documenting effectiveness and formative evaluations" (p. 428).

From a teaching and learning point of view, understanding the affordances of VR/AR tools, coupled with strategies to leverage student agency through designer/creator perspectives can enhance students' classroom experiences [8]; which may be useful in the field of AR/VR content development. Therefore, the first step is to explore how students accept new technologies; empower users to not only make their own VR/AR content but also expand their work into new spaces sharing their work with other users/learners across the world. For this exploration, modeling user's perceptions of technology should employ a framework to explore users' perceptions of utility (PU) and ease of use (PEOU), like the Technological Acceptance Model (TAM) [9]. Using the TAM as a lens of quantifying student's experiences (perceptions) with emergent (VR/AR) technologies, we may begin to understand what

[a] https://customise.toyota.com.au/camry/.

knowledge and skills students acquire in VR/AR content development, such to fill the needed pipeline of developers, designers, instructors, markers, artists, and specialists in this space.

To engage in this work, one must select a reasonable framework (tool) for students to utilize while making VR content in the classroom. Previous studies [10–12] concentrated on the utilization of 3D professional software, such as CAD drawing and Unity 3D, that required extensive prior knowledge and some degree of mastery in programming and design. This may be appropriate approach if the learners are advanced software engineering students. Research suggests novice learners benefit from experiences with software development and programming, in enhancing their psychological understandings of substance and inventive skills [13, 14], so the choice of the appropriate tool is paramount for the level of learner. It is unsurprising that WebVR technology has developed rapidly for nascent developers due to the popularity of smart technologies (e.g. mobile phones, internet accessible tablets). As a result, WebVR development libraries (or frameworks) have appeared (i.e. React VR, Three.js, Vizor, Babylon.js, PlayCanVas, etc.) to help users create 3D applications based upon their personal or academic interests. These libraries are continually being updated to patch or add new features, yet the addition of these new features requires users to understand how to use them on the various available platforms. This has led to certain difficulties for users, especially new users when the system becomes cumbersome and difficult to control. To alleviate this problem, programmers have created more libraries (i.e. A-frame) based on the existing framework/s that are easier to use, shortening programming time and thus, more accessible to learn for new users and students.

This paper, based upon a previous study of students' experiences with VR content development [15], explored post-secondary students' experiences of WebVR for VR content development within an three-tiered introductory computer science course. In addition to previously published data (i.e. Use Case 1), there are two additional cases presented as follows:

- **Use Case 1:** Students were tasked to build their own self-described dream house using the WebVR framework/library. This first use case sought to address these questions: What are the WebVR framework/library that students are interested in? Did the framework meet students' expectations? What challenges did students face during the VR content development? What were suggestions for classroom curriculum and instruction when using WebVR?
- **Use Case 2:** Students were tasked to create an application with any VR/AR framework/library in a specified topic. This second use case sought to address these questions: What VR/AR framework/library did the majority of students select? What role did WebVR knowledge play in their selections? If students did not use WebVR library, what were their reasons?
- **Use Case 3:** Students were tasked to create an application with any VR/AR framework/library in *any* topic of the student's choice. This third use case sought

to address these questions: What topic are students most interested? What VR/AR libraries did they use upon experiencing a variety of frameworks?

The central focus and contribution of this paper is the exploration and communication of post-secondary students' experiences in VR development, from new to experienced user, throughout a sequenced three-experience case study [16]. Therefore, this paper is organized as follows: summarizing related research in Sec. 2; presenting the study design in Sec. 3; Sec. 4 highlights the main points in each use case. Section 5 analyzes the results from students' perspectives in three use cases to address the research questions. Conclusion and future research direction are presented in Sec. 6.

2. Related Work

The concept of VR is not new; it first was introduced in the early nineteenth century by a French general and painter Louis-Francois Lejeune [17]. Since this time, and only very recently, has the idea of VR become a (virtual) reality. VR has been hampered by the absence of VR-ready technologies (e.g. adequate hardware and software) and early equipment costs. This paper does acknowledge the interesting and rich history of VR development, yet does not intend to review all current VR research, rather concentrating on instructive methodologies using VR technologies for content development, specifically.

In education, several efforts have been made to bring VR into classroom, from understanding concepts [18], teaching the content [19], and providing hands-on learning experiences [20, 21], to integrating VR content development into a computer graphics course [22], focusing on students using existing libraries and frameworks [12]. To explore more deeply the affordances of these experiences, Miyata *et al.* [21] examined skill development and group work performance when students created VR applications cooperatively. Throughout the course, this study showed that students' creativity and knowledge gradually increased. A similar study was performed by Häfner *et al.* [23] where students were engaged in developing a 3-year industrial VR project, where task specification played the vital role for students' successes in creating a final VR product. A more in-depth study was conducted by Takala *et al.* [24] where the authors shared their teaching experience of VR development courses throughout 5 years. They emphasized that face-to-face meeting between students and teaching assistant played a vital role in the students' performance. Stansfield [25] presented a study of a VR course that combined both traditional teachings (lecture) supplemented with hands-on experiences in VR development. The author argued that the format of this VR course provided more and varied experiences in communication skills, research skills, and presentation skills, the oft-described "soft skills" needed for student success beyond graduation.

Yet there remain challenges and issues among the research, such as the accessibility and cost of HMDs hardware and the multifaceted programming knowledge and specialized libraries needed for VR content development. Therefore, this paper endeavors to mitigate these current issues when aiding learners (as designers) in creating VR content in a classroom setting. In particular, this paper provides insight into this process by relating novel pedagogical strategies (which will be introduced in Sec. 3.2).

3. Methodology

This research study sought to answer one primary research question (research question 1) and sub-questions (2, 3, and 4), informed by elements of the presented three cases:

- **R1:** What VR/AR development framework/library best allowed novice to expert users to stimulate their interest in creating and sharing VR/AR content in both perceived utility and ease of use?
- **R2:** Was the library/framework that students favored afford students the ability to solve more complex problems?
- **R3:** When given the choice, which library/framework did students employ to develop a VR/AR application (based upon their interests)?
- **R4:** Based upon reported learners' perspectives, what were the pros and cons of WebVR compared to other app-based VR/AR tools?

3.1. *Goal and objectives*

Based on the previous research questions, the goal of this study was to evaluate the feasibility of learning VR development, employing a WebVR application by pinpointing the most suitable framework/library for new users among inter-disciplinary fields, as well as compare the pros and cons of WebVR to other app-based VR/AR applications. Therefore, our instructional goals and objectives are (1) to help students actively engage in the learning process when developing AR/VR applications, (2) to understand students preferences of current typical VR/AR libraries, and (3) to collect and analyze students' experiences when developing VR/AR applications.

3.2. *Study design*

In order to meet these objectives, this study was designed such that students could complete their projects within a given time frame to meet the aforementioned goals and objectives. This was a challenging task since the majority of students did not have prior knowledge of or background in 3D computer graphics and designs. It means that students were at all levels of experience and expertise, sourced from multiple fields and different graduate levels. This was taken into account when devising instructional methods. In our approach, when structuring the course, student engagement was an intentional element for their classroom experiences.

Of the available teaching methods [26] (e.g. inquiry-based, situated-based, task-based, project-based, studio-based), studio-based learning informed the course's teaching strategy, where learners are actively engaged in three-projects development. We investigate each project through a *use case*, whose results were used in total to answer the research questions.

The course activity is shown in Fig. 1 whereas the detail on each activity is presented in the following section.

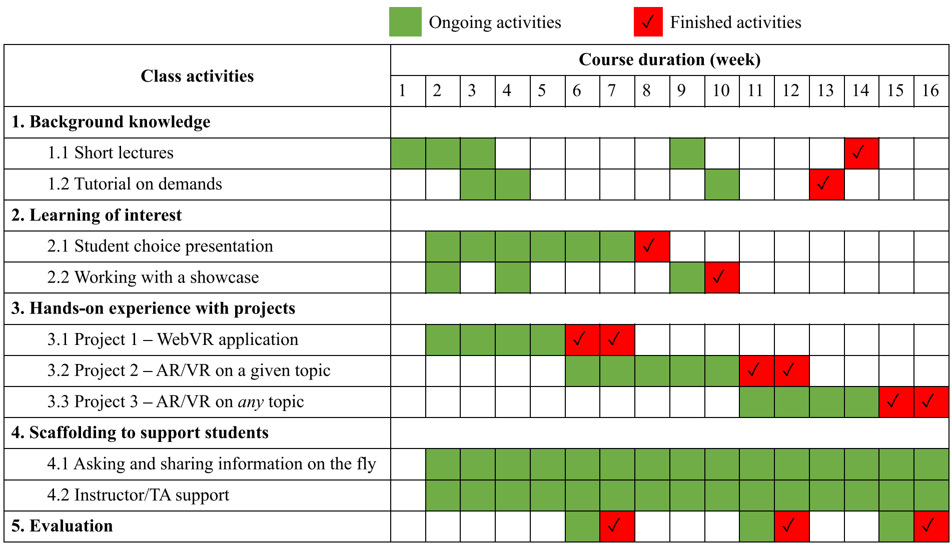

Fig. 1. The study design: 16-week activities.

3.2.1. *Background knowledge*

In-class lectures/tutorials were given at the start of the course to provide an overview of VR/AR, the history and current available VR/AR development tools (e.g. ThreeJS [27], A-Frame [28], Unity3D, Vuforia []), and principles of visual design. Tutorials were given on demand (e.g. read, process and display data or showcase of an application) at students' requests.

3.2.2. *Learning of interest*

Each student was required to present a 5-min in-class talk on a given VR topic of their choosing. The topics for the presentation were the student's choice or suggested by the course instructor in the following domains: AR/VR hardware, software, or research publications. The reason for this talk was for all students to independently acquaint themselves with VR-based new advances, and imparting this information to their peers. The discussed content included but was not limited to: information such as an overview of a given hardware/software/library; pros and cons of VR technology, and VR's impact on emotional sickness, etc.

Along with the exploration of new emerging VR/AR topics, learners had an opportunity to experience and immerse themselves in a VR environment by playing with a small VR/AR showcase (i.e. a virtual reality room). This sample project allowed users to build a basic understanding of HTML and JavaScript from scratch.

3.2.3. *Hands-on experience with projects*

Working with VR projects was the main component of the course activity. The first project case, students built a fantasy or a "dream" house on the web-based VR framework. The purpose of this project was to help students understand the theoretical underpinnings of 3D modeling. In the second project case, a topic was given (i.e. water management) in which learners started going more in-depth, beyond a simple 3D model, to explore a suitable library/framework to help them solve the problem. When students gathered enough knowledge on VR/AR, the third project case enabled them to develop a VR/AR application on any topic, fostering student agency, as well as critical thinking and creativity.

3.2.4. *Scaffolding to support students*

The instructor and the teaching assistant worked collaboratively as mentors to support the students in learning. Course communication consisted of face-to-face interactions in the classroom, online discussion forums (i.e. Piazza), or by email. The studio-based learning approach focuses more on self-teaching and learning, yet the instructor and teaching assistant wished to provide avenues for students to seek help or assistance, as needed.

3.2.5. *Evaluation*

In this expanded study, we conducted three additional assessments (two evaluations for Project Case two, three respectively, and one for the whole course), to gather student feedback of their experiences. Data collection was performed via an online survey tool (i.e. Google Form), and all data was consolidated and saved in Excel. Each assessment was conducted immediately after project completion to best capture students' experiences from the activity. In the first project case (termed "learning from an individual element"), survey questions explored students choices regarding available library/framework for a Web-based VR application, and how they solved a simple problem. In the second project case, a peer-review survey was conducted for each project by each student. Because this case sought to examine VR development from students perspectives, securing their opinion on each part of the project was essential. The survey the third project case was similar to the survey in project case two. An evaluation survey was conducted on the final day of the course, where information on the entire course was collected. The face validity of the questionnaire was evaluated and revised by the instructor and the teaching assistant, to ensure that all questions were easy to understand and inclusive of students' current knowledge of VR development. Survey items and open-ended response coding themes were based upon perceived utility (PU) and perceived ease of use (PEOU) of using the technology, both central constructs of the validated Technological Acceptance Model (TAM) theory [9]. The reliability of the questionnaire was performed by both the instructor and the TA to maintain all responses are inside the scope of conceivable answers.

Ethical concerns in survey data collection: Before any data collection began, students were informed about the research study and were assured that their private information would remain confidential in both data collection and analysis. After students submitted surveys, data were de-identified and copied into Excel for analysis. Data was then coded by converting categorical data into numerical values for analysis, as well as coded using TAM constructs among the open-ended responses.

3.3. *Participants*

Participants in this study were mainly undergraduate ($N = 33$, five graduate) students enrolled in the virtual reality course offered by the Computer Science department at a tier 1 university in the Southwestern United States. There were a total of 38 students (as shown in Table 1) who participated in the study. The average general programming experience among participants was 3.5 years. This computer science course was comprised of university students who held general programming skills as an expectation for enrolling in the course. This course was advertised for students, in other majors other than computer science, so sampled students represented cross-disciplinary subjects.

Table 1. Participant distribution by *gender* vs *graduate level*.

	Undergraduate	Master	PhD
Male	28	2	2
Female	5	1	0
Sum	33	3	2

4. Use Case Study

4.1. *Use case 1: Developing a WebVR application*

4.1.1. *Project description and deliveries*

In order to answer the first research question (R1), students garnered experiences in virtual reality by exploring current WebVR libraries and created a simple web-based VR application. The project's topic was to build their VR "dream house" in which a user can experience and interact with 3D objects in a scene. Students had the freedom to develop the application in any programming language, using any library or software. The project required to navigate and perform simple interactions with several object models by using both computers and hand-held devices (tablet or smartphone) via an internet browser.

Project delivery: Completed VR applications were presented in the classroom. Students showed their VR dream houses using a desktop computer connecting to a projector and WebVR. Interactions in the scene were performed by using a mouse and keyboard. Students could also view the VR house with their laptops, tablets, smartphones, or immerse themselves in the virtual world with a Google Cardboard HMD. If students chose the latter VR option, they moved around by pressing the button at the top right of the Google Cardboard, whereas other interactions (e.g. turning on/off lights or triggering blinds) were triggered by gazing at the objects within a certain amount of time (i.e. 3 s).

4.1.2. *Project assessment*

The project was developed in one month and students presented their work, to one another, in the classroom. A report of how they carried out their tasks, created models or sourced models (obtained from the open sources) was also submitted for instructor evaluation. To evaluate students' projects, three criteria/requirements were used in a rubric fashion, where each level gradually increased in difficulty. The purpose of categorizing the requirements into three levels was to evaluate the completeness of the tasks within a short time. Overall, the levels are summarized as follows:

- **Level 1:** The student was able to create a house model from scratch or customize an existing model; make a 3D simple object or use at least 10 free 3D models, and light up the house.
- **Level 2:** The student was able to import more five of 3D models to the scene without affecting the performance of the application, make some simple adjustments such as to control the light of the house (for example turn on/off, increase/decrease the intensity of the light, or switch to different color themes), and navigate around the house.
- **Level 3:** The student was able to interact with a particular object (e.g. drag/move/scale, run animation, or close/open doors), or display dynamic/movable objects in the house.

4.1.3. *Survey*

To gather data for answering the research question, a post-activity survey was conducted to ask students their opinions about the WebVR application. The questions were informed by the two constructs of the Technology Acceptance

Table 2. Research questions for the survey.

Question number	Question	Question type	Research question (s)
Q1	Which library/framework did you use to do your project? (e.g. Three.js, Unity, Babylon...).	Open-ended	R1
Q2	How did you find your library/framework?	Single choice	R1
Q3	To what extend did you find your library to use on the scale 1 (easy)–10 (very difficult)?	Likert scale	R2
Q4	Did you find your chosen library meet your expectation to finish the project on the scale from 1 (No) to 10 (fully met the expectation)?	Likert scale	R2
Q5	To what minimum required level of programming (you think) to learn and use your chosen library?	Single choice	R2
Q6	How did you get the models for your project?	Multiple choice	R1, R2
Q7	How did you learn about the library that you used for project?	Multiple choice	R2
Q8	How many days that you spent for your project (including time to study your library and programming language)?	Open-ended	R2, R3
Q9	How did you rate the difficulty of the project on the scale 1–10?	Likert scale	R2, R3, R4
Q10	How did you rate the usefulness (for your learning of building WebVR) of project on the scale 1–10?	Likert scale	R2

Model: perceived ease of use (PEOU) and perceived usefulness (PU). The survey consisted of 10 items which are shown in Table 2.

4.1.4. *Project results and reports*

Along with presenting the projects in the classroom, students were required to submit a report describing their WebVR applications with references (for downloaded assets), recording any challenges they experiences and lessons learned. At the end of the one-month project period, 37 houses were created by students. Figure 2 provides a

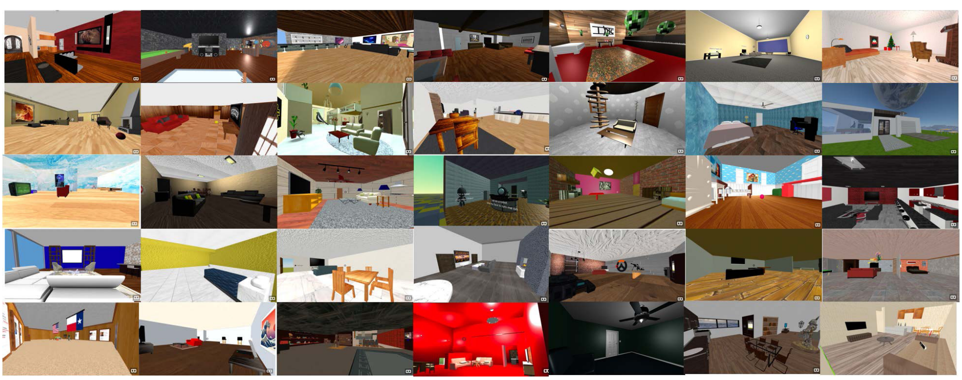

Fig. 2. A collage of WebVR applications created by sampled students.

snapshot of 35 WebVR applications created by students. We show an enlarged example of a WebVR house on the moon in Fig. 3(a), a relaxing dream house in Fig. 3(b), and a bar dream house in Fig. 3(c).

(a) (b) (c)

Fig. 3. Three good WebVR project examples: (a) A moon dream house (b) A relaxing dream house (c) A bar dream house.

The performance results (grades) of each student's project are illustrated in Fig. 4; the majority of students who finished their projects were awarded a Levels 2 and 3. Further analysis of grades is provided in the results section.

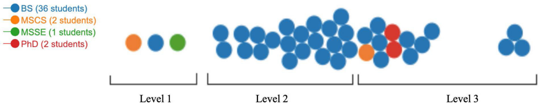

Fig. 4. Students' grade distribution of the WebVR dream house project: Level 1 is equivalent to a C while Level 3 is equivalent to a A.

4.2. *Use case 2: Developing a VR/AR application on a given topic*

4.2.1. *Project description and deliveries*

Unlike Project case one which focused on web-based VR application, in Project case two, students had the freedom to choose any application and hardware platform (e.g. web VR/AR, iOS VR/AR, Android VR/AR, or Oculus app). This project required students to collaboratively create (in groups) a VR/AR application that focused on a given topic, which was water education.

Project delivery: The VR/AR application was presented in the classroom by the student's chosen device or Oculus Rift (upon request), where the running application was shown on the screen with a projector. Along with the delivery of the developed applications, students were again required to submit a report that addresses all the issues, successes, and challenges faced during project development. These reports were used in part of the data analysis.

4.2.2. *Project assessment*

In the second project case, students' performance was evaluated by both instructor, teaching assistant, and their peers via a Google Form. The peer evaluation is shown in Table 3.

Table 3. Survey questions for peer evaluation in project case two.

Number	Question	Question type
Q1	How do you rate the difficulty (technical challenging) of this project on the scale from 1 (easy) to 10 (very difficult)?	Likert scale
Q2	How do you rate the usefulness of this project on educating water/ environment on the scale from 1 (useless) to 10 (very useful)?	Likert scale
Q3	How do you rate the interestingness of this project on the scale from 1 (boring, don't want to try) to 10 (very much like to play with this VR/ AR game)?	Likert scale
Q4	How do you rate the Innovation of this project idea on the scale from 1 (not at all) to 10 (very creative)?	Likert scale
Q5	How do you rate the visual design of this project on the scale from 1 (not appealing at all) to 10 (very visual appealing)?	Likert scale
Q6	How do you rate the sound effect of this project on the scale from 1 (distractive, or no sound track at all) to 10 (very effective)?	Likert scale
Q7	How do you rate the usability of this project on the scale from 1 (no guidance or very difficult to use) to 10 (very intuitive to use)?	Likert scale
Q8	How do you rate the effort of this group on the scale from 1 (no effort) to 10 (best effort)?	Likert scale
Q9	How do you rate the chemistry of this group on the scale from 1 (no teamwork at all) to 10 (best teamwork)?	Likert scale
Q10	How is your vote (score) for this group on the scale from 1 (bad) to 10 (very good)?	Likert scale
Q11	Your questions/comments for this group	Open-ended

4.2.3. *Project results*

In the second project case, a total of 12 VR/AR applications were produced from groups. Figure 6 provides a snapshot of visual design layouts for each group's project.

4.3. *Use case 3: Developing a VR/AR application on any topic*

4.3.1. *Project description and deliveries*

In this final project case, students were again released from any restriction on a tool but now were allowed to select their topic. The purpose of this project had two intentions: first, to determine which framework/library students chose and second, what research direction they preferred regarding topics.

4.3.2. *Project assessment*

Similar to Project case two, student performance assessments were conducted via an online survey tool. The question set was reused from the previous case, Table 3 except for question number 2 as the topic requirement was not prescribed as it were in Project case two.

4.3.3. *Project results*

At the end of project case three, a total of 14 group VR/AR applications were developed. A snapshot of visual design layout is shown in Fig. 7.

5. Results and Discussion

A large portion of this section addresses the concepts related to research question 1, whereas research questions 2, 3, 4 were positioned as supporting queries for research question 1.

5.1. *Research question 1: What VR/AR development framework/library best-allowed novice to expert users to stimulate their interest in creating and sharing VR/AR content in both perceived utility and ease of use?*

For this research question, we analyzed Use Case 1 in more detail using descriptive and correlational analysis to garner a deeper understanding of students' reported experiences from the TAM-based post-project survey ($N = 36$).

Descriptive Analysis. As depicted in Fig. 5(1), the majority (92%) of sampled students chose *A-frame* as their WebVR library to complete their first project; two students used both *A-frame* and *Three.js*, whereas only one student used *Unity3D*. Reasoning included suggestions provided by this instructor Fig. 5(2), accounting for 77.78% and searching on the internet (13.89% — due to student absence or being unfocused in class). Although *A-Frame* claims to be one of the easiest libraries to create WebVR content for any user [28], 9% of students reported that it was difficult to use (only three students rated its *A-Frame* difficulty from 8 to 10). Anecdotally, a part of this problem may have stemmed from the fact that some students had no experience with web development (when asked in the classroom and conversations with the teaching assistant). In general, the chosen library (largely *A-Frame*) was considered by students to be of moderate difficulty regarding ease of use ($mean = 5.09$). Overall, students' expectations of the chosen library were highly acceptable in terms of utility (overall mean score is greater than 7.0) as in Fig. 5(4). In terms of programming skills required to learn the library, a majority of students (97%) felt that they did not need advanced knowledge in completing their project, whereas only 3% reported that they needed to learn more coding. In the class during this project, students spent most of their time in the project customizing objects (e.g. shape, size, color, position, rotation) rather than creating new objects. These results suggest that students with a moderate level of programming skills were more able to creatively design and execute (accept the use of) a WebVR project within a short period. Students also reported that the most time-consuming part of the project was in content creation as creating 3D models can be an arduous task, especially for users who are new to the 3D model design process. Consequently, 84.21% of students decided to use (i.e. download) existing models from the internet as depicted in Fig. 5(6). In evaluating students projects, researchers found that students often made 3D models from the basic shapes (e.g. a cube, a sphere, etc.); and these low poly models increased the performance (rendering time and improved interactivity) of the application significantly compared to those downloaded from the internet. Only a few students (7 out of 38) challenged themselves in using third party software to

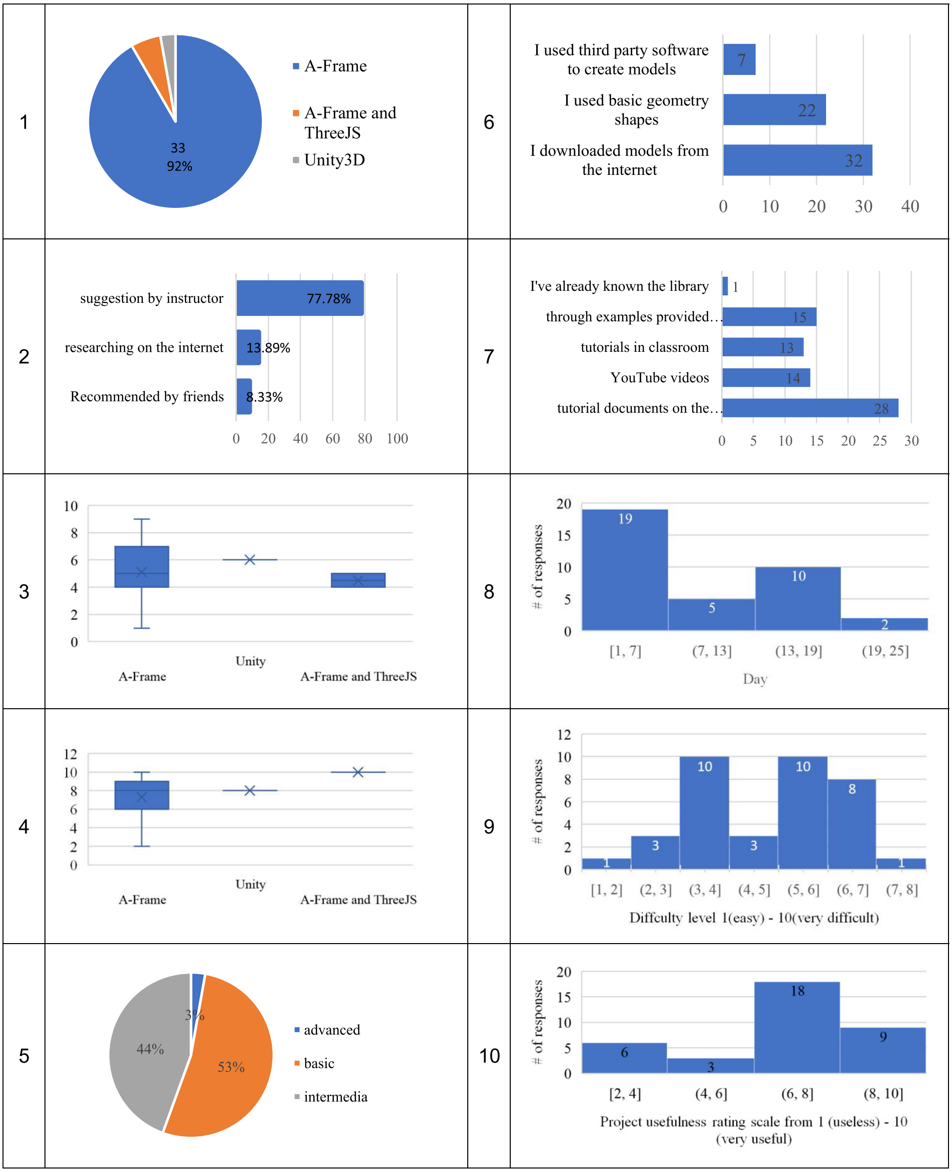

Fig. 5. Survey results from students in Project 1 from question 1 to 10.

create models. Although a sample project was provided to students at the beginning of their project to provide context to the learning activity, students' feedback in Fig. 5(7) showed that they actively referenced tutorial documents on the chosen library website (73.68%). Overall, half of the class (50&) was able to finish the project in one week, as shown in Fig. 5(8). This estimation would have been more precise had we measured students hourly progress. Meaning, some students

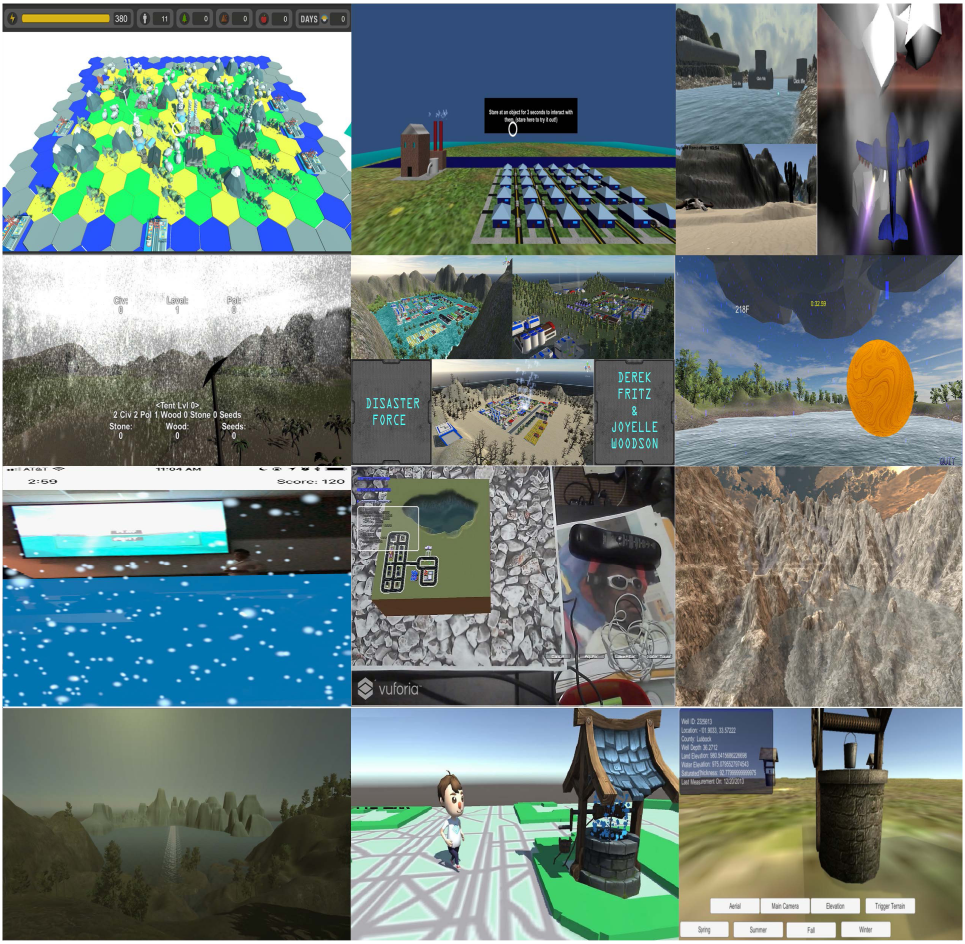

Fig. 6. A collage of VR/AR applications created by students.

spent whole class days working on their project, while others allocated multiple class days to their project. We found that this duration was reasonable for the majority of the students who wanted to experience VR.

Figure 5(9) illustrates students' feedback in terms of the difficulty of the project. Students felt that this project was neither too easy nor too difficult (mean = 5.25 and Std. Deviation = 1.57). This was expected when introducing a WebVR project for students from disparate disciplines. On the other hand, the usefulness of the project (i.e. learning VR) did not yield expected results (mean = 7.08 and Std. Deviation = 2.28) as seen in Fig. 5(10) since it only focused on WebVR application. Yet, this result may be a good indicator to help instructors in preparing similar classes using a student learning-centered model [29].

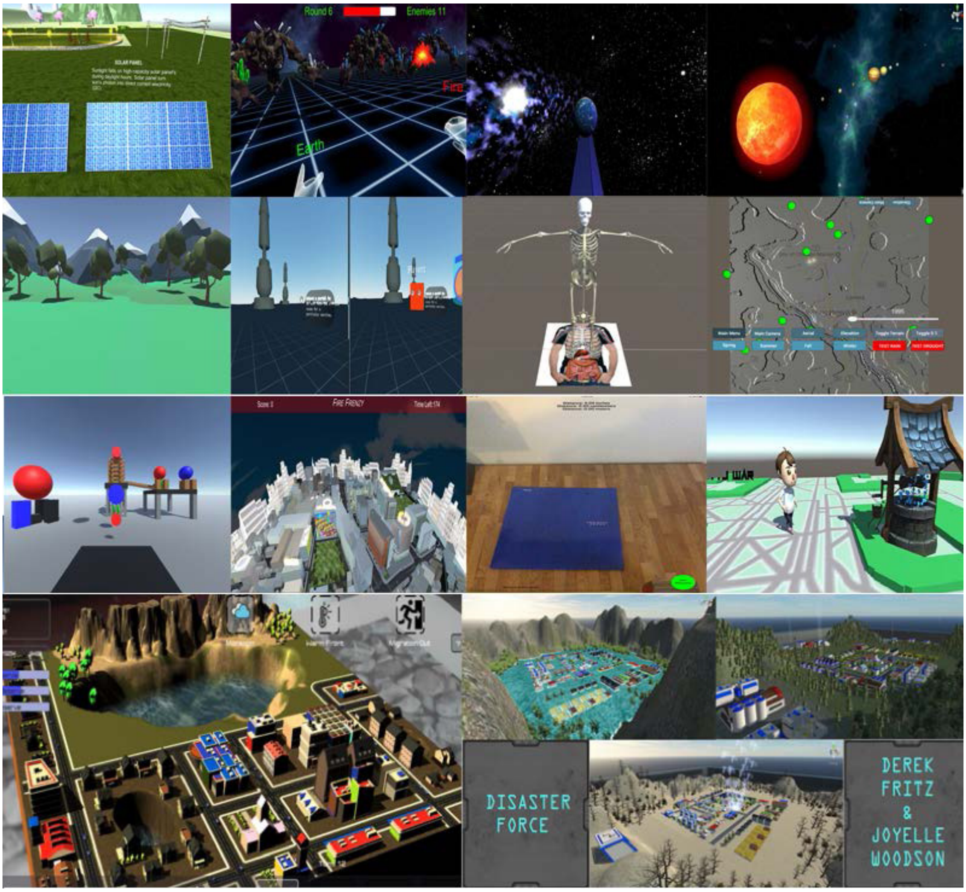

Fig. 7. A collage of VR/AR applications created by students in Project 3.

Correlation analysis: To further investigate the use of the chosen library, we conducted a correlation analysis between the chosen library with reported project expectation. The Pearson correlation was used as the metric for evaluation via SPSS software. Table 4 shows the correlation values among survey questions (items) Q3, Q4, Q9, Q10. It can be seen that there is a moderate negative correlation ($r = -0.401, p = 0.015$) between Q4 (library expectation) and Q9 (project difficulty), indicating a relationship between high library expectation and lower difficulty of the project (and vice versa). While exploring the correlation between the ease of use of the chosen library with library expectation to finish the project, the Pearson correlation output ($r = -0.441$) was only moderately negative and significant only at the 0.01 confidence level. A weak correlation can also be found between library ease of use and perceived project difficulty ($r = 0.620$ and $p = 0.000$).

Table 4. Pearson correlation test scores produced by SPSS software [15].

		Q3	Q4	Q9	Q10
Library ease of use (Q3)	Pearson Correlation	1	−0.441	0.620	0.06
	Sig. (2-tailed)		0.007	0.000	0.713
Library expectation (Q4)	Pearson Correlation	−0.441	1	−0.401	0.272
	Sig. (2-tailed)	0.007		0.015	0.108
Project difficulty (Q9)	Pearson Correlation	0.620	−0.401	1	−0.109
	Sig. (2-tailed)	0.000	0.015		0.526
Project Usefulness (Q10)	Pearson Correlation	0.064	0.272	−0.109	1
	Sig. (2-tailed)	0.713	0.108	0.526	

Correlation is significant at the 0.01 level (2-tailed)
Correlation is significant at the 0.05 level (2-tailed)

5.2. *Research question 2: Was the library/framework that students favored afford students the ability to solve more complex problems?*

In Project case two (summarized in Table 5), a large portion of applications was developed in Unity3D (11 out of 12 apps) and only one application was carried out in Xcode (an integrated development environment (IDE) for MacOS user). None of the applications were created using A-Frame as introduced and practiced in Project case one. This was due to several reasons: (1) the curiosity of using a new hardware device (i.e. Oculus Rift), although not all student's laptops were able to meet the minimal hardware requirement to run Oculus software; (2) not all requirement were available in the current WebVR library, students came up with "how to use" components in Unity3D rather than creating their component in JavaScript; (3) the new IDE tool (Unity3D and Xcode) provided a more intuitive way to generate and mange programming codes; (4) team dynamics directed groups to a unified or common framework/library; and (5) finally, the fear of being alone on the project facilitated students to go with 'the crowd', that is, deferring to peers for advice. This could be the reason why other development tools (e.g. Unreal Engine) were not taken into students' considerations.

Table 5. A summary of the VR/AR application types and hardware.

	Tool/library	Application type	Hardware	Count
Project 2				**12**
	Unity3D	VR	Oculus Rift	3
	Unity3D	VR	Desktop/Laptop	6
	Unity3D	AR	Smartphones	2
	Xcode	AR	Smartphone	1
Project 3				**14**
	Unity3D	VR	Oculus Rift	5
	Unity3D	VR	Desktop/Laptop	2
	Unity3D	VR	Smartphones	2
	Unity3D	AR	Smartphones	3
	Xcode	AR	Smartphone	1
	A-Frame	VR	Smartphone, Computers	1

5.3. *Research question 3: When given the choice, which library/framework did students employ to develop a VR/AR application (based upon their interests)?*

Unlike Project case two, where half of the VR/AR applications were deployed on students personal laptops, in the third and last project case, only two groups used this strategy. In this case, students mainly used Oculus Rift and their own smartphones for their VR applications. The final survey showed that when students had the freedom of developing their interest (through topic selection), they reported that the majority of students wanted to explore and experience some "fun stuff" on their devices (smartphone). Other groups continued developing and improving their existing (second project case) project because they wanted to "learn more about it". Only one group continued to use the first learn library of A-Frame; their reasoning for developing in A-Frame was for "easy testing" on students' devices.

5.4. *Research question 4: Based upon reported learners' perspectives, What were the pros and cons of WebVR compared to other app-based VR/AR tools?*

The analyses presented in the following section does not try to address all WebVR issues compared to other related app-based VR/AR applications, instead, this is a summary of students' reported experiences during the development of the three projects which were given in the students' reports, questions posed in classroom and survey responses at the end of the semester. They have been aggregated into binary categories (i.e. of pro and con) of their experiences.

5.4.1. *WebVR application development*

Pros

- *Fast development*: With a little knowledge on HTML and JavaScript, users were able to create and enjoy a VR environment with ease, even with new users that said "Web VR is not that difficult to develop".
- *Responsive community*: The community provided responsive feedback to users once a question was asked. Named applications were Slack, Github, Twitter, and Stack Overflow.
- *Compatible devices*: Since WebVR runs on an internet browser that supported WebGL, it was compatible with most modern smartphones.
- *Open-source*: Written in vanilla JavaScript and HTML, the majority of WebVR library provided an open-source code for enthusiasts.
- *Easy sharing*: The VR content was easily shared, with multiple users, by providing a link to a webpage. In addition, users were able to enjoy being with each other in the same VR environment without purchasing additional specialized hardware (e.g. social VR room as presented by one group in Project case 3).

- *Easy integrate available VR contents*: There were a lot of available components that allowed users to bring VR content and 3D models to life with ease using a single line of code.
- *Available components*: Although still in development, continuously developing components allowed creators to speed up their work (e.g. Aframe-extras and Networked-aframe)

Cons

- *Performance*: One of the reported downsides of WebVR performance was due to its lower frame-rate per second (fps) compared to native VR/AR applications. (The fps sometimes drops in a non-native support WebVR browser (i.e. 25–30 fps)).
- *Loading time*: Since WebVR uses an internet connection to load VR content, loading time is dependent on current network traffic and bandwidth. Hence, VR content varied in the time it took to load and retrieve.
- *Coding control*: It was challenging for users to manage and control a certain part of the application when its size continued growing (with the addition of more codes and components). Reporting that this became especially problematic with asynchronous issues, as users did not know which part of the programming ran first, leading to difficulty in debugging.
- *No standardized WebVR*: Because WebVR is still in the early development stage, there were many libraries/frameworks being developed. Thus, student faced issues with a lack of standardization.
- *Unstable libraries and components*: The developing libraries/components kept updating every two to five months to patch or add more functions. In addition, some components did not work with the legacy version leading to unstable systems or vice versa.
- *Available components*: Compared to other VR/AR development tools, the availability of webVR components still behind. This was one of the reason why none of students used WebVR for a more complex problem.

5.4.2. *App-based VR/AR application development*

Pros

- *Well developed IDEs*: Existing development tools and IDEs were more stable and well developed.
- *Good and stable performance*: The unexpected fps dropping in WebVR was rarely found in app-base VR/AR applications. Users had smooth experiences in VR due to the availability of graphic cards.
- *Rich community*: Unlike the younger WebVR community, a more established community can provide and share a lot of information for new learners.
- *Available VR contents/components*: There is plenty of free VR content that can be found on Assets stores with rating information from reviewers. Users were able to shorten development time by using free developed components with ease.

Cons

- *Specialized hardware*: The first issue was the lack of HMDs headsets for testing VR experiences. The second problem came from the development device hardware. In our study, none of the participants had laptops that met the minimum requirement of Oculus Rift SDK installation. Students had to create a VR/AR for testing on desktop/laptop first, and then go to the instructor's lab to test the final version as said *no one on the team had a supported GPU and travelling to the lab to debug, attempt to set up VR, and debug again wasn't viable, this was not possible.*
- *Sharing experience*: It was difficult to share the same experience with multiple users at the same time due to the cost of HMDs and graphic card requirements.
- *Huge space for development*: Compared to WebVR, app-based VR/AR applications took more space, making it difficult to share source code for testing and demonstration. This was especially relevant with online repositories that does not upload large files (e.g. Github restricts file to 100Mb).
- *API incompatibility*: One of the biggest issues when dealing with app-based VR/AR development was the compatible API with legacy system on deployed devices. The first problem was due to differences in software versions, so When students shared the application for testing and demonstrations, one component may not have worked well on the other computers. The other issue related to Operating System (OS) problems, where VR/AR applications would halt, quit or go to a black screen when the OS was upgraded.

5.5. *Lessons learned and discussion*

The results of the three project cases provided a unique insight in students' technological acceptance, with a focus on the use of WebVR, compared to other app-based VR/AR approaches, for individual and collaborative development of VR content among learners of various programming skill levels in an undergraduate computer science course. Although the *Three.js* library was introduced first and was highly recommended to students for its utility and ability to understand the basic ideas of how WebGL works, only two students used this library for their projects. This is likely due to the complexity of the code and the time needed to set up the basic VR environment and interactions, suggesting a great deal of prior knowledge needed to maintain and control the program, so it would run correctly when a new component was imported. The second library (*A-frame*), on the other hand, was more attractive to students because of the ease of use, namely its simplicity and easy VR setup compared to *Three.js*. Besides, instead of spending time on programming efforts, students focused on logical design and performance of the application. For example, while working with the high fidelity 3D models (i.e. obj or fbx), the size of these models was big, heavily affecting the performance of the application. Students overcame these problems either by (1) creating a similar low poly model, or (2) converting these heavy models to another format (gltf — GL Transmission Format [30]). Therefore, students' perceptions of ease of use became paramount with equivalent platforms in utility, mediating their technological acceptance and overall use.

Regarding utility, students found novel ways to utilize the software to meet their needs. One strategy was in 3D object interactions; while some libraries did not allow direct interaction with 3D models (meaning they would only interact with some primitives such as cube, rectangle, cylinder, torus with pre-defined Box Collider), one student created an invisible big cube that cover the whole 3D models in the same position, functioning as a *Box Collider,* although was not. From the user's perspective, it appeared to interact with the 3D model, but in fact, it only interacted with the invisible cube. Another interesting trick was in creating scroll blinds on house windows, an animation made by merely creating two cubes on top of the windows and moving to the bottom windows, once the user clicked on the switch. When the cubes moved down, they blocked the light from entering into the house, creating an illusion of window blinds.

Although all students had some prior experience with programming, not all of them possessed troubleshooting skills, or the ability to find out the reasons why application encountered an error. Such errors are unavoidable, especially when the size of the project gets bigger with complex VR applications. For example, it is easy to fix a mistake if there is an indicator of an error, however, what happens if the application shows no error? Several students faced this problem and one common strategy was to remove/comment parts of the application (*leave-one-out* approach) until it ran successfully. Then, students incrementally added back the code/s to locate the errors.

Besides the advantages of WebVR framework, there were some difficulties in developing WebVR applications. The first challenge was how to handle complex 3D models. Although these models could have been converted to another format to reduce their size, they lost fidelity, reducing the apparent realism unique to (Web) VR environments. The second challenge was to reduce latency when users changed their orientations suddenly, that often causes dizziness or nausea. Known as VR sickness, these challenges are common, yet crucial in maintaining a virtual presence in 3D, VR environments [31].

The advantages of WebVR in terms of "ease of use" and "online sharing" would have not become apparent without considering Use case 2 and Use case 3, where students faced a issues in *Collaborative design and issues with version control for certain aspects of the project* or *Dealing with library/hardware incompatibilities and fixing problems with the tools that we are using.* These challenges did not arise when WebVR was required as a text editor for the development.

Recommendations: Based on this research, we present four recommendations for educators, instructors, and course designers when teaching undergraduates VR content development, especially WebVR, in the classroom:

- Before project development, short presentations on WebVR applications should be introduced to and by students. First, it engages students in the process by researching VR content and securing materials for self-guided learning. Second, it motivates and inspires other students by listening to peers showcasing the diversity of extant VR assets. And third, it provides a snapshot of the knowledge and interests of peers as indicated in the course feedback *I didn't realize all of the different applications for VR, but learned a lot from the student choice presentations.* Each presentation should only take around five to seven minutes as longer presentations may lead to information overload.

- To motivate students to start the projects earlier, a basic sample project should be provided. It can work as a "skeleton" project (or scaffold) as a baseline from where students can start, especially for those who are new to programming to *Learning how to use it in a very short time.* By having a sample project, students may conceptualize what an exemplar of WebVR project development is as well as frame expectations of the course.
- For agile project development, use of an online discussion forum and moderation can play an essential role in helping students throughout the development process. Any question that emerges during project development can be posted, receiving feedback quickly, without having to wait until the next face to face class. In this online environment, exciting findings may be shared among students, providing opportunities for students to the co-construct meaning of the process and products [16].
- Regarding technological acceptance, *A-frame* was found to be a good starting point for undergraduate students and new learners to VR content development. From survey responses and project deliveries, students preferred *A-frame* as a skeleton structure and gradually developed projects based upon utility and ease of use, as predicated by the TAM. Complex codes may be embedded later on through careful scaffolding of information and guiding students through advanced programming.

6. Conclusion and Future Work

This paper presents three consecutive cases of 38 post-secondary students' experience and perceptions of technological acceptance in developing VR content using current WebVR platforms, among other app-based applications and topics, with diverse programming knowledge backgrounds. Results suggest that WebVR, still in the early stages of development, held high perceived usability and ease of use on a wide range of hardware devices, independent of any operating system; which facilitated technological acceptance among even the most novice of users. Although issues of hardware access remains an ongoing challenge for classrooms across the world (i.e. ineffective or obsolete hardware, restrictions to third-party software, and so on), WebVR has broad appeal due to the ubiquitous presence of web-based technologies. Leveraging the TAM in examining students' perceptions in three case projects, researchers found students' skills gradually increased in VR content development. Although the scope of the study was on students' perceptions of their VR content development experiences, this empirical contribution provides unique insight and guidance for instructors who wish to instruct eager, yet diverse, learners in VR content development. A limitation of this study was a lack of consideration of alternative emerging libraries/tools (e.g. Babylonjs or Unreal Engine). Despite these platforms being mentioned and made available, the student was discouraged due to hardware and software limitations.

In future work, we hope to continue our research by exploring in more detail in Use case 2 and Use case 3 in terms of students' learning in new frameworks/libraries. Moreover, the availability of more hardware such as different types of VR headsets, gloves, and treadmill integration to give students hand-on experiences and haptic feedback are critical to the success of future VR classes.

References

[1] J. A. Barceló, M. Forte and D. H. Sanders, *Virtual Reality in Archaeology* (Archaeo-Press, Oxford, 2000).

[2] B. Atzori, H. G. Hoffman, L. Vagnoli, A. Messeri and R. L. Grotto, Virtual reality as distraction technique for pain management in children and adolescents, in *Advanced Methodologies and Technologies in Medicine and Healthcare* (IGI Global, 2019), pp. 483–494.

[3] M. Rus-Calafell, P. Garety, E. Sason, T. J. Craig and L. R. Valmaggia, Virtual reality in the assessment and treatment of psychosis: A systematic review of its utility, acceptability and effectiveness, *Psychol. Medi.* **48**(3) (2018) 362–391.

[4] Y. C. Huang, K. F. Backman, S. J. Backman and L. L. Chang, Exploring the implications of virtual reality technology in tourism marketing: An integrated research framework, *Int. J. Tour. Res.* **18**(2) (2016) 116–128.

[5] B. L. Ludlow, Virtual reality: Emerging applications and future directions, *Rural Special Education Quart.* **34**(3) (2015) 3–10.

[6] N. Robitaille, P. L. Jackson, L. J. Hébert, C. Mercier, L. J. Bouyer, S. Fecteau, C. L. Richards and B. J. McFadyen, A virtual reality avatar interaction (vrai) platform to assess residual executive dysfunction in active military personnel with previous mild traumatic brain injury: Proof of concept, *Disability Rehabilitation: Assistive Technol.* **12**(7) (2017) 758–764.

[7] J. Psotka, Immersive training systems: Virtual reality and education and training, *Instruct. Sci.* **23**(5–6) (1995) 405–431.

[8] C. Deed, P. Cox, J. Dorman, D. Edwards, C. Farrelly, M. Keeffe, V. Lovejoy, L. Mow, P. Sellings, V. Prain *et al.*, Personalised learning in the open classroom: The mutuality of teacher and student agency, *Int. J. Pedagogies Learn.* **9**(1) (2014) 66–75.

[9] F. D. Davis, R. P. Bagozzi and P. R. Warshaw, User acceptance of computer technology: A comparison of two theoretical models, *Manage. Sci.* **35**(8) (1989) 982–1003.

[10] F. M. Dinis, A. S. Guimarães, B. R. Carvalho and J. P. P. Martins, Virtual and augmented reality game-based applications to civil engineering education, in *Global Engineering Education Conference*, 2017, pp. 1683–1688.

[11] A. Z. Sampaio, M. M. Ferreira, D. P. Rosário and O. P. Martins, 3d and vr models in civil engineering education: Construction, rehabilitation and maintenance, *Auto. Construct.* **19**(7) (2010) 819–828.

[12] V. T. Nguyen and T. Dang, Setting up virtual reality and augmented reality learning environment in unity, in *IEEE Int. Symp. Mixed and Augmented Reality*, 2017, pp. 315–320.

[13] F. Kalelioğlu, A new way of teaching programming skills to k-12 students: Code. org, *Comput. Human Behav.* **52** (2015) 200–210.
[14] E. Wang, Teaching freshmen design, creativity and programming with legos and labview, in *31st Annual Frontiers in Education Conference*, 2001, pp. F3G–11.
[15] V. T. Nguyen, R. Hite and T. Dang, Web-based virtual reality development in classroom: From learner's perspectives, in *IEEE Int. Conf. Artificial Intelligence and Virtual Reality*, 2018, pp. 11–18.
[16] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes* (Harvard University Press, 1980).
[17] C. Duffy, *Borodino and the War of 1812* (Seeley Service & Co., 1972).
[18] D. Bell, Teaching virtual reality, *ACM SIGCSE Bullet.* **28**(2) (1996) 56–61.
[19] J. Zara, Virtual reality coursea natural enrichment of computer graphics classes, in *Computer Graphics Forum*, Vol. 25(1) (Wiley Online Library, 2006), pp. 105–112.
[20] D. Kamińska, T. Sapiński, N. Aitken, A. Della Rocca, M. Barańska and R. Wietsma, Virtual reality as a new trend in mechanical and electrical engineering education, *Open Phys.* **15**(1) (2017) 936–941.
[21] An educational framework for creating vr application through groupwork, *Comput. Graph.* **34**(6) (2010) 811–819.
[22] D. C. Cliburn, J. R. Miller and M. E. Doherty, The design and evaluation of online lesson units for teaching virtual reality to undergraduates, in *IEEE Frontiers in Education Conference*, 2010, pp. F3F–1.
[23] P. Häfner, V. Häfner and J. Ovtcharova, Teaching methodology for virtual reality practical course in engineering education, *Procedia Computer Science* **25** (2013) 251–260.
[24] T. M. Takala, L. Malmi, R. Pugliese and T. Takala, Empowering students to create better virtual reality applications: A longitudinal study of a vr capstone course, *Informa. Educ.* **15**(2) (2016) p. 287.
[25] S. Stansfield, An introductory vr course for undergraduates incorporating foundation, experience and capstone, *SIGCSE Bull.* **37** (2005) 197–200.
[26] A. S. Carter and C. D. Hundhausen, A review of studio-based learning in computer science, *J. Comput. Sci. Colleges* **27**(1) (2011) 105–111.
[27] R. Cabello, Javascript 3d library https://threejs.org/.
[28] Mozilla, A-frame school https://aframe.io/aframe-school/.
[29] J. G. O'Brien, B. J. Millis and M. W. Cohen, *The Course Syllabus: A Learning-Centered Approach* (John Wiley & Sons, 2009).
[30] Khronos, Gl transmission format https://www.khronos.org/gltf/.
[31] R. Hite, G. Childers and M. G. Jones, Hardware affordances and challenges to produce presence and learning in k-20 science virtual reality environments, in *Handbook of Mobile Teaching and Learning*, 2019, pp. 1–12.