Pre-college Computer Science Initiative for Augmented and Virtual Reality Development

Eric Nersesian*, Adam Spryszynski*, Tracy Espiritu†, Michael J. Lee*
*New Jersey Institute of Technology
{eric.nersesian, as2569, mjlee}@njit.edu
†Passaic County Technical Institute
tespiritu@pcti.tec.nj.edu

***Abstract*—New curricula initiatives are growing to meet near-future, industrial demand for computer science (CS) graduates with Augmented and Virtual Reality (AR/VR) development knowledge. Universities are often at the forefront in developing these curricula to help prepare their students for industry jobs. High schools wanting to offer college aligned CS courses for their students typically work with local universities to adapt courses for their students' needs. This paper presents such an effort along with results from a student survey showing the successful implementation of college-level courses through training of high school teachers. The curricula from this study are available for public use at artncoding.com and may be adapted as needed by educational programs to meet the emerging employment needs of their students in the AR/VR field.**

***Index Terms*—STEM Education, Computer Science, Educational Technology, Virtual Reality, Augmented Reality**

Introduction

New technological fields are emerging that will become major job markets for future computer science (CS) graduates. These emerging technologies, including Augmented and Virtual Reality (AR/VR), are appealing subject areas for high schools and colleges to provide their students. Multiple reporting agencies have written reports on the near-term, large growth phase that the AR/VR industries are about to experience. BusinessWire reports that AR is expected to be worth $60 billion by 2020 [1]. Extended Reality report showcases a $209 billion market size for AR/VR by 2022 [2]. Consultancy UK predicts the AR/VR market to boom to $178B by 2022 [3] and the Transparency Market Research advisory stated that, "the worldwide market for AR/VR is estimated to report an exponential CAGR of 92.50% between 2016 and 2024, increasing its overall opportunity to US at $547.20 billion by the end of 2024" [4]. These reports show that a job market for a new series of CS skills will become a vital area of growth for the nation's economy.

To address these emerging industries, the New Jersey Institute of Technology (NJIT) has been developing curricula for undergraduate AR/VR software engineering education for several years. The curricula has been a natural extension to the pre-existing game development program, which shares a large overlap of required skills and knowledge. High school and colleges can take advantage of this, adapting curricula and instructor resources established for game development to foster initial AR/VR curriculum growth.

The Elizabeth School District of New Jersey learned about this growing CS trend while working with NJIT. To take advantage of the efforts of NJIT's growing AR/VR program, the school district decided to enhance their own CS program of study by incorporating college-level AR/VR development courses as electives at one of their high schools, John E. Dwyer Technology Academy (Dywer Academy). The goal was to boost student participation in CS and increase the number of students graduating with both college-level and industry credentials.

Faculty at NJIT developed the curriculum for Dywer Academy, titled "Developing AR/VR Applications." This effort was funded in part by an Advanced CS Competitive Grant from the New Jersey Department of Education, and in collaboration with Oculus VR (a leading AR/VR hardware manufacturer owned by Facebook), and Unity Technologies (a leading game engine company). The curriculum focuses on introduction to 3D and VR development with C# in Unity. Burning Glass Technologies, a job market analytics platform, reports that "Unity is one of the most in-demand tech skills and has one of the highest forecasted growth rates, at over 39% over the next two years" [5]. We tested the course in NJIT's undergraduate game development program. The contents were refined over two iterations/semesters, with student feedback and learning outcomes used to verify success.

High school CS instructors were trained on the curriculum during a four-day workshop hosted by NJIT. We conducted post-workshop to learn about future needs and how to support ongoing high school educational needs to deliver this curricula. These CS instructors taught the course over one academic year (two semesters) at Dywer Academy and were in continuous contact with NJIT's faculty and industry partners to help facilitate a successful course delivery. The curriculum is publicly available via website, artncoding.com, for any interested school systems to review and incorporate into their schools.

Related Works

Recent literature suggests that integration of AR/VR technologies alongside existing teaching methods is feasible. The systematic review by Klimova et al. finds that traditional teaching methods, such as lectures or laboratories, are still predominantly used in AR/VR oriented education. The main distinction in teaching methodologies differentiating 17 courses analyzed by this review is the combination and proportion of

teaching methods used [6]. Developing a course focused on building creativity in a team setting allowed the instructors to use the VR platform to integrate graphics, development, and design. The instructors have observed students paying more attention to interaction design when developing VR applications, as well as investing a significant amount of time in their projects [7]. A 2016 paper examines the usefulness of AR in teaching biology by comparing teaching materials based on 2D graphics, 3D graphics, and 3D objects. Student interviews indicate interest and satisfaction in working with AR. The authors conclude that AR is a valuable tool allowing educators to implement practical hands-on learning in the classroom [8].

High school CS teachers face a variety of challenges. For example, Yadav et al. conducted a qualitative study of 24 teachers uncovered challenges resulting from inadequate preparation and training [9]. They found that most teachers had to learn the CS content on their own, even if they did not have any prior experience in the subject [9]. Professional development workshops have shown success in exposing middle school teachers to technologies and resources used in computer science education [10]. Similarly, Goode et al.'s "Exploring Computer Science" program combines content knowledge with instructional techniques in a professional development program, which makes a strong case for empowering teachers to introduce active learning in settings where traditional curricula have proven to be inadequate [11].

Hands-on programs used to promote and improve student learning have proven effective. A study conducted by Christensen et al. examined positive STEM dispositions of students from three different programs using the STEM Semantics Survey [12]. They concluded that active learning programs can improve or maintain STEM dispositions in high school settings [12]. CS education curriculum revolving around making animations or games have proven to be successful in primary students, resulting in increased student engagement [13]. Game-based learning in after-school programs, such as Lee's work using the Gidget programming game, has shown to increase learners' engagement in with programming, even for members from underrepresented groups in computing [14], [15].

Multidisciplinary approaches that integrate CS and computational thinking with other disciplines have shown a positive impact on student outcomes. A study investigating the effects of the math, dance, and music program suggested that besides the increased CS knowledge, students showed increased interest in STEM [16]. Similarly, a computational thinking courses aimed at applied scientific computation resulted in students reporting interest in pursuing further CS education [17].

Design-based learning is an extension of inquiry-based learning, where activities involve problem-solving using design thinking and design processes. STEM programs using this paradigm have shown promising results in a variety of settings. Introducing engineering design to primary school students improved their ability to apply their math and science skills to evaluate and redesign their work [18]. Duran et al. used this paradigm in a collaborative design-based after-school program combining seminars and workshops to improve IT skills and their use. Results showed a positive impact on students' understanding of roles that IT skills hold in STEM, with some evidence of impact on STEM attitudes among the urban high school students [19].

The literature also supports techniques that are closer to the domain of this study. The Game-Design and Learning initiative (GDL) implements hands-on game design and programming activities for primary students participating in after-school, in-school, and summer programs. Students who participated in the initiative show significant improvement in their problem-solving skills [20].

METHOD

University Curriculum Structure

NJIT's Ying Wu College of Computing offers a course called "Information Design Techniques," that provides a practical overview of interactive design tools, techniques, and principles for 3D development using the Unity creation engine. This course teaches 3D project structure as students create and customize a 3D painting application using feature driven development. Curriculum covers project management, sprint based development, C# scripting, 3D computer graphics, interaction design, user flow diagramming, user interface (UI) design, animation, rendering, physics, and user input.

The course is split into four sections: user input, 3D graphics, animation, and code refactoring, with programming and interaction design being present in all sections (see Table I). Each section has 3-4 weeks of instruction, with each week having three hours of class time (1.5 hours of live instruction and 1.5 hours in online instruction) and three hours of project time. Both live and online instruction are accompanied by exercises to practice the new knowledge. The graded items in the class are the project deliverables due at the end of each of the four sprints and a final exam.

University Curriculum Weekly Learning Objectives

Week 1's learning objectives involve basics of interaction design, flowcharts, Unity interface, and scripting. Students will be able to apply interaction design principles to user flow charting, and function as a user of the Unity platform.

Week 2's learning objectives involve user input, camera, raycasting, screen- and world -space, hierarchies, objects, and build requirements. Students will be able to create, instantiate and destroy objects in hierarchies, and understand screen, local, and world coordinate systems.

Week 3's learning objectives involve random number generation, position, scale, rotation, color theory, attributes, and UI elements. Students will be able to control transforms of 3D primitives, generate world position from screen-space mouse position, and organize UI elements.

Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Sprint	1				2				3				4			
Topic	User Input				3D Graphics				3D Animation				Refactoring			

TABLE I
CURRICULUM SCHEDULE FOR SEMESTER 3D DEVELOPMENT CLASS

Week 4's learning objectives involve the math library, scripting documentation, algorithms, loops, and switch statements. Students will be able to implementing basic math equations in their code, prototype out new feature ideas, and use scripting documentation to learn more about code libraries.

Week 5's learning objectives involve creating complex user flows, diagrams, object representations, and UI functionality. Students will be able to generate a higher level of detail in their user flows, UI functionality, and organization.

Week 6's learning objectives involve deciphering user input, increasing content complexity, and expanding UI features. Students will be able to create cleaner hierarchies for better navigation, edit data sent back to UI for better usability, and organize UI elements into clear delineations.

Week 7's learning objectives involve UI usability decisions for grouping, organizing, and naming schemes, looping structures and logical operators. Students will be able to condense UI information, use loops to manage unknown amounts of objects, and stress testing features.

Week 8's learning objectives involve basics of motion design, procedural animation, and the animation system. Students will be able to build and control animations with code and with an animation system.

Week 9's learning objectives involve creating objects with multiple components, and animating UI elements.

Week 10's learning objectives involve problem solving techniques, animation creation, and breaking down complex parts of code into simpler parts.

Week 11's learning objectives involve organization of complicated UI interfaces, rendering orders for 2D and 3D graphics, multi-threaded code, and bug fixing.

Week 12's learning objectives involve testing procedures, external assets, data collections, and standardization practices.

Week 13's learning objectives involve naming, commenting, organization and refactoring processes for clean, legible code.

High School Curriculum Requirements

The Elizabeth School District's motivations to push for this program redesign was due to enrollment drops after the first and second year in their high schools. Administration wanted to include more opportunities for students to apply their programming skills in an application that is trending in the CS industry. In addition, students felt that the current CS scope and sequence was not interesting, resulting in low engagement and reduced enrollment in the advanced courses. When surveyed, students communicated that they selected CS because of their interest in the gaming industry. The belief by administration was that offering an exciting program at Dywer Academy, such as an AR/VR programming course, would expand student opportunities that are aligned with their interests but also enhance their career choices in technology. The curriculum for this AR/VR programming course was based on NJIT's "Information Design Techniques" undergraduate class (see Figure 1).

The one-semester college curriculum was converted to a two-semester year long, high school curriculum as a joint effort between the college faculty, career and technical education (CTE) supervisors, and the high school CS teachers. Once the high school CS teachers were trained on the curriculum in a training workshop, they were able to determine the complexity of the curriculum and how to pace it appropriately for high school CS students. A series of weekly meetings was held between the college faculty and the high school CS teachers as the curriculum and its related exercises were slowed down for a younger student demographic. It was agreed from both sides, that each of the 13 weeks of college curriculum presented in above section should be expanded into two weeks of curriculum with a week dedicated to direct instruction and a week dedicated to project-based learning for the students.

The series of weekly meetings also addressed classroom management techniques and curriculum alternatives for students who may have issues using the technologies due to disabilities, as all students should have a role in the AR/VR programming course. Similar to other curriculum and learning standards, accommodations can be made based on the students' 504 plans and/or Individualized Educational Plans (IEPs). This may include research-based best practices, instructional strategies, in-class support, special equipment or modifications, and/or student pairing. In some cases, space and location of the class may need to be considered. Workstation physical layout and equipment use cases were discussed for various disabilities and contingency plans were developed. Just as diversity demands universal design, students with disabilities involved in the design and programming of virtual environments can provide unique perspectives and considerations to applications of AR/VR. In turn, with their participation, AR/VR applications can expand to service those with disabilities.

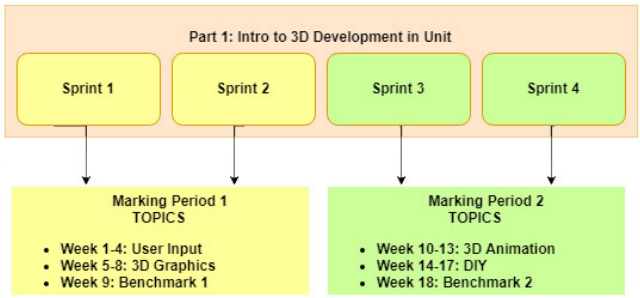

Fig. 1. Conversion of the university curriculum for 3D development

Fig. 2. University faculty training high school educators

Collegiate and Industry Advisory Board

We created an advisory board that consisted of a CTE supervisor from the Elizabeth School District, two faculty members from NJIT, and representatives from the education division of Unity Technologies and Facebook. The collaboration with NJIT, Unity, and Facebook in the development of the curricula and proposed program of study helped ensure alignment with NJIT's CS course, providing students with the opportunity to earn college credits. Additionally, the program was designed to prepare students for the Unity Certified Developer Certification Exam and earn industry credentials.

Pre-college Agreement of University and High School Partners

NJIT has a Center for Pre-College Programs (CPCP), which partners with K-12 schools across the state. The CPCP allows high schools to provide college-level courses at their schools. Using the college course syllabi, CPCP-approved high school teachers teach these courses where students can earn college course credits. Our courses were approved by the CPCP to be taught in the Elizabeth School District.

High School CS Teacher Training Workshop

We ran a workshop to train the high school CS teachers in using Unity to develop 3D and VR programs during the summer before the high school AR/VR class was first implemented. Weekly meetings were held throughout the summer leading up to a four day workshop. The weekly meetings involved NJIT faculty working with high school curriculum writers and CS teachers to develop the high school curriculum based on the college curriculum. Once the high school curriculum was established, the workshop was organized by NJIT faculty to train the high school CS teachers.

The first three days were full day training on Unity, C# programming, 3D computer graphics, user interfaces, and animation. The training continued by going through the college level curriculum in a condensed format. The learning objectives were to ensure the teachers could follow the college class and training videos independently to prepare for the coming academic year. The last day was hands-on training on the VR equipment (see Figure 2). Teachers were trained on how to use the equipment and common issues that they will face when using the equipment in a classroom setting with students. Training ended with learning how to develop in Unity specifically for the VR equipment to be used in the class, the Oculus Rift VR headset.

Question number	What is your knowledge level on	1st Semester Start of Semester	2nd Semester Start of Semester	Start of Semester Differences	1st Semester End of Semester	2nd Semester End of Semester	End of Semester Differences
1	Unity	1.35	1.373	0.023	3.7	3.655	-0.045
2	C#	1.65	1.638	-0.045	3.35	3.448	0.098
3	Programming	3.05	2.948	-0.102	3.75	3.845	0.095
4	Interaction Design	1.35	1.828	0.478	3.5	3.724	0.224
5	3D Development	1.75	1.621	-0.129	3.5	3.724	0.224
6	Animation	1.35	1.672	0.322	3.545	3.517	-0.028
7	User Interfaces	1.579	2.052	0.473	3.571	3.862	0.291
8	Visual Studio	2.0	2.017	0.017	3.737	3.759	0.022
AVG	**The average of the eight responses**	**1.76**	**1.894**	**0.134**	**3.582**	**3.692**	**0.11**

TABLE II
SURVEY RESULTS TABLE - START AND END OF SEMESTER RESPONSES

RESULTS

University Class Surveys

The Introduction to 3D Development college course ran over two semesters. The first semester had two sections totaling 48 students, and the second semester had three sections totaling 76 students. We conducted the survey as a non-graded, voluntary survey during the first day and last day of class. The survey was designed as a quick questionnaire with eight questions ranked on a five-point Likert scale. An open-ended response question was added to the end of the semester survey, asking for additional comments. All questions involved asking for the students' perceived knowledge level of a curriculum area from the class. The students were asked to rank their knowledge of each area on a five-point Likert scale from little (1), below average (2), average (3), above average (4), and a lot (5). The questions asked, "What is your knowledge level on..."

1) Unity
2) C#
3) Programming
4) Interaction Design
5) 3D Development
6) Animation
7) User Interfaces
8) Visual Studio

22 students from the first semester and 59 students from the second semester responded to the survey (see Table II). Overall responses from the first semester had a mean of 1.760 and 3.582 at the start and end of the semester, respectively (for an increase of 1.822). Overall responses from the second semester had a mean of 1.894 and 3.692 at the start and end of the semester, respectively (for an increase of 1.798).

All responses for both semesters had lower start of semester self-reported knowledge levels than their end of semester counterparts (see Figure 3). Start of semester knowledge levels for the first semester had a minimum of 1.35 for Unity, interaction design, and animation, and a maximum of 3.05 for

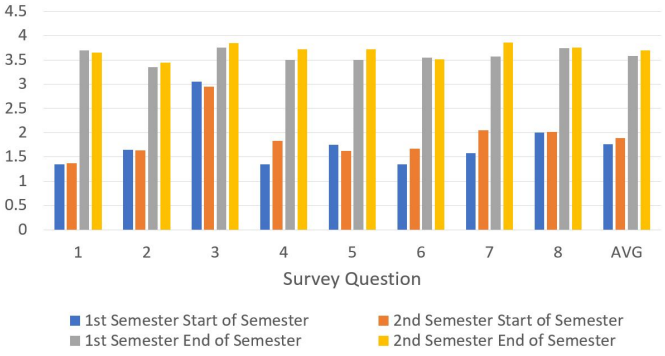

Fig. 3. Survey results graph - start and end of semester responses

programming. Start of semester knowledge levels for the second semester had a minimum of 1.373 for Unity, and a maximum of 2.98 for programming. End of semester knowledge levels for first semester had a minimum of 3.35 for C#, and a maximum of 3.75 for programming. End of semester knowledge levels for second semester had a minimum of 3.448 for C#, and a maximum of 3.862 for user interfaces.

The survey itself had eight questions grouped around two central learning objectives of programming and design. Four questions (1-3, and 8) were grouped around programming, "What is your knowledge level on...Unity, C#, Programming, and Visual Studio. The other four questions (4-7) were grouped around design, "What is your knowledge level on...interaction design, 3D development, animation, and user interfaces.

Start of semester knowledge levels for first semester had a mean of 2.013 for the programming questions and a mean of 1.507 for the design questions. Start of semester knowledge levels for second semester had a mean of 1.994 for the programming questions and a mean of 1.793 for the design questions. End of semester knowledge levels for first semester had a mean of 3.634 for the programming questions and a mean of 3.529 for the design questions. Start of semester knowledge levels for second semester had a mean of 3.677 for the programming questions and a mean of 3.707 for the design questions.

The mean of the four questions grouped around programming showed a 1.621 difference in first semester with a starting mean of 2.013 and an ending mean of 3.634, and a 1.683 difference in the second semester with a starting mean of 1.994 and an ending mean of 3.677. The mean of the four questions grouped around design showed a 2.022 difference in first semester with a starting mean of 1.507 and an ending mean of 3.529, and a 1.914 difference in the second semester with a starting mean of 1.793 and an ending mean of 3.707 (see Table III).

14 students from the first semester, and 40 students from the second semester gave additional comments. Student comments ranged from negative and positive views of the class curriculum but were mostly positive reinforcing the quantitative differences between the start and end semester surveys. The comments that were critical of the curriculum usually had a constructive feedback element incorporated into it to help the instructors improve the class experience.

Positive comments involved student appreciation of curriculum areas, completing project deliverables successfully and being emotional content with the class results. Curriculum areas pointed out by students were user flow diagramming, introduction to C# programming, creative problem solving, and basic art and design principles into a computing class. Students writing positive comments were content with instructional flow of the class, especially the inclusion of online training videos and project code. These supplemental online materials were mentioned as productive self-guided learning time for the students. Some sample comments from both semesters:

- "What I really liked about the class was thinking out the user flow process and drawing those steps. I had never programmed with C# before taking this course, but now after 15 weeks I feel more comfortable with the language."
- "I really enjoyed the class. I had only taken a Java and a Python class prior to this class. It is good to have a few weeks on the fundamentals of C# before starting Unity. This is useful for people like me who didn't have prior C# knowledge. The flow-charting was an excellent learning experience. It helped me to do the main project for my Introduction to CS course."
- "I went into this class not knowing anything about C# and programming. Even though I don't feel like I could recreate what I have done in the sprints on my own without watching the videos, it still got me very interested in learning more about the material that was taught."
- "My favorite aspect of this class is how much it taught me that I CAN [*sic*] do animation and 3D design. This class taught me that I don't need to know how to draw with a pencil to be good at animation and game design. I just need a passion, a vision, and patience. In the end, many animations are math-based anyway. This class has been an inspiration for me."

Negative comments involved misalignment of the class curriculum, and development environment technical issues. There are a large amount of curriculum areas to cover in this survey style class, and students felt that not enough attention was paid to the more important curriculum areas. Some students felt that user experience and interaction design warranted stronger focus in the class and to minimize time spent on animation and code use. Other students expressed some frustration over technical issues with the development environment. Particularly, some versions of Unity had broken UI elements like sliders and drop downs that required the student to port over their project to another version of Unity. Some sample comments from both semesters:

- "Too much focus on supplementary aspects (code and animation), instead of user experience and interaction."
- "The class itself was informative and did help me learn Unity. However, there were a few problems with Unity that was addressed in the class that were never fixed by Unity. Aside from that, the class was good."

What is your knowledge level on	1st Semester Start of Semester	1st Semester End of Semester	1st Semester Delta	2nd Semester Start of Semester	2nd Semester End of Semester	2nd Semester Delta
Programming	2.013	3.634	1.621	1.994	3.677	1.683
Design	1.507	3.529	2.022	1.793	3.707	1.914

TABLE III
PROGRAMMING AND DESIGN KNOWLEDGE RESULTS TABLE - START AND END OF SEMESTER RESPONSES

High School Teacher Workshop Interviews

A post workshop semi-structured group interview with the high school CS teachers was conducted to understand their level of comprehension of the curriculum and comfortability factors with teaching it the coming academic year. They found the material interesting not only for themselves but as a new way to approach CS topics for their students. They found

the workshop through in its material, but tiring in its pace. Although the teachers had a CS background and proficient with several programming languages, the ideas of 3D computer graphics and development were too novel for the teachers to fully absorb in one week of full time training. They were generally overwhelmed in learning 3D computer graphics, 3D math topics, the Unity development environment, and C# programming all at once.

The teachers stated that the workshop was sufficient enough for them to continue by themselves with the training videos to prepare for the coming academic year. They had wished that the training was spread out over a few weeks instead of one week and to revisit topics as needed. They found that many new domain areas were exposed to them in a short amount of time. The foundational knowledge needed to teach an introductory class in just 3D development, not even approaching AR/VR topics, was too vast to absorb in one week. This feeling of being overwhelmed was alleviated by the fact that the college class was video recorded and available to them on artncoding.com.

DISCUSSION

University Class Survey Interpretations

Both semesters of surveying have large enough populations to extrapolate meaning from the results. We can see from the graph that both start of semester questionnaires are similar in their responses and are lower than their end of semester counterparts, which is to be expected if the course was designed and taught well for students to learn and feel comfortable in their knowledge levels. Having both semester surveys show similar trends in overall increase in student knowledge levels in all areas of curriculum gives the university educators confidence that the curriculum is refined, serves its educational objectives, and ready for conversion to high school level education. This was all verified before the major efforts between the university and high school partners to convert the tested curriculum for high school education.

We can see that the start of semester questions are around the 1-2 point area (a little to below average knowledge) for both semesters, with a mean of 1.76 for the first semester and 1.894 for the second semester. The exception being the programming knowledge question where both start of semesters are the maximum response compared to the other start of semester questions at 3.05 and 2.948 respectively. The start of semester knowledge level for programming is still below the end of semester for both semesters, showing an increase in knowledge but not as big of a change over the course of a semester compared to the other knowledge areas. We believe this is due to the fact that the term "programming" is vague, and most computing students have had previous programming classes so self-reporting of start of semester knowledge is high.

When the questions are grouped into their categories of programming and design, it becomes apparent that the computing students do have inherent knowledge stronger in programming (2.013 first semester, and 1.994 second semester) compared to design (1.507 first semester, and 1.793 in second semester). The knowledge areas do come to similar levels in the end of semester surveys with programming (3.634 first semester, and 3.677 second semester) knowledge close to design (3.529 first semester, and 3.707 second semester).

The additional comments show in general that the students do enjoy the unique combination of programming and design principles in this curriculum. Most of their classes are solely on programming principles, with little user experience or general design training. The students find this curriculum to not only be a pleasant change from their core curriculum but also as a new approach to programming that helps strengthen their creative problem solving and gives new light to a repeated curriculum area.

Study Limitations

As with other studies examining multidisciplinary and novel educational approaches, there are no standardized assessments for evaluating students in this domain. Some of our work relies on data that was self-reported by students. Finally, given the ethical requirements for working in education settings with minors, the survey was voluntary, which introduces the potential for self-selection bias.

Future Work

We plan to build out and test new curriculum areas introducing the topics of AR/VR software design and engineering at the college level. We plan to work with our high school partners to convert the materials for their use and also to release all educational material publicly, free for all to use.

CONCLUSION

This paper presents a curricula initiative to meet industrial demand for CS graduates with AR/VR development knowledge. Universities are often at the forefront in developing these curricula to help prepare their students for industry jobs. High schools wanting to offer college-level courses for their students typically work with local universities to adapt courses for their students' needs. In this initiative, student surveys from the university classes showed successful implementation of college-level courses in AR/VR development. College faculty worked with CTE supervisors of a local school district to convert the curricula for high school classroom use. The curricula from this study are available for public use at artncoding.com and may be adapted as needed by educational programs to meet the emerging employment needs of their students pursuing academic and industry jobs in the AR/VR field.

ACKNOWLEDGMENT

We thank the Elizabeth School District, Dywer Academy, the New Jersey Department of Education, Facebook, and Unity for their support of this work. Any opinions, findings, conclusions, or recommendations are those of the authors and may not reflect the views of any of these parties.

REFERENCES

[1] "Global Augmented Reality (AR) and Virtual Reality (VR) Market Worth $60.55 Billion and $34.08 Billion by 2023." [Online]. Available: https://www.businesswire.com/news/home/20180508005963/en/Global-Augmented-Reality-AR-Virtual-Reality-VR

[2] J. Scribani, "Infographic: What is Extended Reality (XR)? - Visual Capitalist," 2019. [Online]. Available: https://www.visualcapitalist.com/extended-reality-xr/

[3] Consultancy.uk, "Virtual and Augmented Reality market to boom to $170 billion by 2022," 2018. [Online]. Available: https://www.consultancy.uk/news/17876/virtual-and-augmented-reality-market-to-boom-to-170-billion-by-2022

[4] "Virtual and Augmented Reality Market worth US$547.20 bn by 2024 | CAGR of 92.50%." [Online]. Available: https://www.transparencymarketresearch.com/virtual-augmented-reality-market.html

[5] "Visualizing the Future Demand for 3D Graphics and Real-time 3D Across the Economy," Tech. Rep., 2019.

[6] "Existing Teaching Practices in Augmented Reality," in *Procedia Comput. Sci.*, vol. 136. Elsevier B.V., jan 2018, pp. 5–15.

[7] K. Vasylevska, I. Podkosova, and H. Kaufmann, "Teaching virtual reality with HTC Vive and leap motion," in *SIGGRAPH Asia 2017 Symp. Educ. SA 2017*. Association for Computing Machinery, Inc, nov 2017.

[8] Y.-H. Hung, C.-H. Chen, and S.-W. Huang, "Applying augmented reality to enhance learning: a study of different teaching materials," *Journal of Computer Assisted Learning*, vol. 33, no. 3, pp. 252–266, 2017.

[9] A. Yadav, S. Gretter, S. Hambrusch, and P. Sands, "Expanding computer science education in schools: understanding teacher experiences and challenges," *Computer Science Education*, vol. 26, no. 4, pp. 235–254, 2016.

[10] I. Morais and M. S. Bachrach, "Analyzing the impact of computer science workshops on middle school teachers," in *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2019, pp. 57–61.

[11] J. Goode, J. Margolis, and G. Chapman, "Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model," in *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, pp. 493–498.

[12] R. Christensen, G. Knezek, and T. Tyler-Wood, "Alignment of hands-on stem engagement activities with positive stem dispositions in secondary school students," *Journal of Science Education and Technology*, vol. 24, no. 6, pp. 898–909, 2015.

[13] L. Werner, S. Campe, and J. Denner, "Children learning computer science concepts via alice game-programming," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, 2012, pp. 427–432.

[14] M. J. Lee, "Increasing minority youths' participation in computing through near-peer mentorship," *Journal of computing sciences in colleges*, vol. 35, no. 3, 2019.

[15] ——, "Exploring differences in minority students' attitudes towards computing after a one-day coding workshop," in *ACM ITiCSE*, 2019, pp. 409–415.

[16] M. Shamir, M. Kocherovsky, and C. Chung, "A paradigm for teaching math and computer science concepts in k-12 learning environment by integrating coding, animation, dance, music and art," in *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2019, pp. 62–68.

[17] S. Hambrusch, C. Hoffmann, J. T. Korb, M. Haugan, and A. L. Hosking, "A multidisciplinary approach towards computational thinking for science majors," *ACM SIGCSE Bulletin*, vol. 41, no. 1, pp. 183–187, 2009.

[18] L. D. English and D. T. King, "Stem learning through engineering design: fourth-grade students' investigations in aerospace," *International Journal of STEM Education*, vol. 2, no. 1, p. 14, 2015.

[19] M. Duran, M. Höft, D. B. Lawson, B. Medjahed, and E. A. Orady, "Urban high school students' it/stem learning: Findings from a collaborative inquiry-and design-based afterschool program," *Journal of Science Education and Technology*, vol. 23, no. 1, pp. 116–137, 2014.

[20] M. Akcaoglu and M. J. Koehler, "Cognitive outcomes from the game-design and learning (gdl) after-school program," *Computers & Education*, vol. 75, pp. 72–81, 2014.