Interdisciplinary Collaboration Approaches on Undergraduate Virtual Reality Technology Projects

Eric Nersesian, Margarita Vinnikov, Jessica Ross-Nersesian, Michael J. Lee
Department of Informatics
New Jersey Institute of Technology
Newark, NJ USA
{eric.nersesian, margarita.vinnikov, jessica.c.ross, mjlee}@njit.edu

***Abstract*—Educational approaches must keep pace with the rapidly advancing state of technology so that students have the necessary skills for the modern workforce. Computer science (CS) education presents an interesting cross-section of challenges to science, technology, engineering and mathematics (STEM) education to explore the effects of alternative teaching methods. Our undergraduate program has been working on these educational challenges for several years. We have found project-oriented studio classes with computing and design students collaborating on emerging technology projects lead to positive outcomes. This paper presents our current cross-class collaboration method along with student surveys and final presentation results. It is a necessary class structure to successfully educate future developers and designers, and we wish to share our experiences with the larger STEM educational community.**

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

Introduction

Educational approaches must keep pace with the rapidly advancing state of technology so that students have the necessary skill sets for the modern workforce. Recent research supports that innovative methods of teaching, such as educational technologies, have documented improvements over traditional education methods [1], [2]. These improvements can be seen in the student's visual short-term memory, abstract reasoning, spatial cognition, and multitasking abilities [3]. This has influenced the United States Office of Educational Technology to pursue stronger integration of curriculum with immersive and engaging technologies [4]. Teaching methods need to accommodate this reality, as students are likely to be engaged with cutting-edge technology [5]. One approach to update teaching methods is to supplement lecture or textbook activity with technological or collaborative alternatives.

Computer science (CS) education presents an interesting cross-section of challenges to science, technology, engineering and mathematics (STEM) education to explore the effects of alternative teaching methods. When teaching CS concepts, educators often face a complex intersection of STEM disciplines. Instructional challenges of teaching abstract CS concepts can make it difficult to keep students inspired to pursue CS and keep them engaged throughout their academic career. Additionally, many non-STEM fields require basic CS education for employment, but non-STEM students required to take basic CS classes face similar if not increased engagement and comprehension issues. This poses a risk of driving students away from CS, and possibly even their field of interest.

Our program has been working on these educational challenges for several years. We have found project-oriented studio classes integrating STEM and non-STEM students with collaboration on emerging technology projects are leading to positive outcomes. The computing students find it easier to figure out the software engineering solutions when the problem is scoped by the design students. The computing students can talk to the design students when they need more context to continue working on the software solution. The design students find it easier to learn the basics of software development practices by designing small software projects involving emerging technologies that are highly visual in their nature. While this helps both STEM and non-STEM students to have better classroom environments for learning CS, this pedagogical practice is mirroring real-world software engineering practices of multidisciplinary teams of designers and programmers building software together. This has the additional benefit of getting both student populations prepared for real-world job conditions.

Related Works

Previous research has looked at the challenge of increasing the learning and interest of STEM students in CS using immersive technologies [6]–[9]. In a study by Moore et al., the use of augmented reality (AR) and virtual reality (VR) to teach CS showed that technology makes the learning material less abstract using a more approachable technology [6]. They argued that the application of AR/VR techniques to teaching software engineering in higher education may alleviate learning obstacles attributed to its abstract and complex nature. Of particular interest, they point out that virtual simulations of system processes may appear more tangible to students. Gokhale demonstrated a teaching style that used the inherent students' interest in technology to teach and connect them to CS concepts [7]. Their approach utilized evidence-based question answering for students to understand abstract CS concepts by exploring the ideas through the lens of how the concept is applied to their favorite technologies. Both research approaches found favorable methods for making abstract CS concepts for STEM students more concrete.

Pancratz and Diethelm focused on the students' thought process to help underrepresented STEM minorities, such as females, to improve their enrollment and retention in a CS degree [8]. They demonstrated that teaching problem solving techniques by breaking down large problems into parts not only teaches good life skills but helps a larger student audience learn fundamental CS concepts, such as object oriented programming, modularity, and divide and conquer techniques. The research team used rapid prototyping techniques for small Internet-of-Things as the teaching medium. Zhu and Panorkou looked at the whole of simulation technologies to help students comprehend STEM concepts [9]. They explored use cases of computer simulations in K-12 environment science involving the modeling environment, Netlogo, and the visual programming language, Scratch. Students were able to manipulate system parameters and interpret the results of environmental changes. These teaching techniques present novel approaches to teaching complex CS and STEM concepts.

Other researchers have also looked at a similar challenge of inspiring non-STEM students to become interested in STEM and CS learning using immersive technologies [10]–[13]. Wang and Frye used VR to focus on engaging art aspects of STEAM with experiential learning to recruit underrepresented females into STEM [10]. Surveys indicated that students had minimal STEM backgrounds and found the arts and crafts style hands-on activities were engaging and helped them acclimate to abstract concepts and STEM career potentials. Seo, and Lawrence exposed design students, especially females, to software engineering with VR as an approachable technology [11]. They used design focused students inherent interest in visual technologies to teach them that CS concepts are an approachable academic discipline. 3D printing, CAD software, and VR technologies were a variety of teaching aids used in this project driven workshop. These studies showed the importance of visualizing STEM topics when teaching them to non-STEM students.

Shamir et al taught and increased interest in CS to non-CS students with VR, art, and animation [12]. They developed a teaching paradigm using interlacing structures in dancing, body poses, song melodies, and painting composition that increasing student interest in pursuing CS careers. Shibata, and Kasiwagi focused on building a learning model to teach CS to non-CS liberal arts students in Japan [13]. They taught basic computer operation and system skills by focusing on the organization of scientific knowledge between connected lecture topics, and developed a learning style specific to this organization. Teaching styles presented by these researchers revolved around structure of STEM knowledge and making connections to similar structures in non-STEM knowledge.

The interdisciplinary collaboration approach we are presenting in this paper has been shown by previous research to bridge these two similar challenges in two different student demographics [14], [15]. Kim and Stogdill combined teachers from different disciplines in one class to inspire students and make the STEM learning material more interesting [14]. A physician and an English teacher taught a secondary course reviewing contemporary issues in science and medicine. Students responded positively to learning dense medical topics in a student-driven, open conversation structure. Frydenberg, and Andone found that having students collaborate on VR projects increased their interest in technology [15].The project paired students from universities in the United States and Romania to remotely visit and learn about each other's region and culture through shared VR development projects. Students reported broadening their understanding of applications for future technologies. These exploratory studies of using emerging technologies to increase integration of STEM education leads the direction for our own collaboration efforts of computing and design undergraduate education.

METHOD

At the New Jersey Institute of Technology (NJIT), our Informatics program within the College of Computing teaches applied CS concepts to information technology (IT) students. Our teaching environment has led us to be sensitive to situations where students can get overwhelmed by the science and math fundamentals as we progress through the software engineering curriculum. We have found that exposing students to CS topics through applied learning principles in software engineering and design can have beneficial results to their engagement and retention factors. Therefore, we focus on the engineering and design concepts of software development and teach CS and math concepts on an as-needed basis. We have found that choosing emerging technologies like AR/VR can help inspire IT students to try harder with CS and math fundamentals by catering to their fascination with new technologies. This applies especially to project-based learning using technologies they are engaged with for entertainment purposes.

Our undergraduate program has a sister program in our university's College of Architecture and Design, called Digital Design. Students in that program are taught graphic design, computer graphics, and software design. Both programs share teaching responsibilities for design practices pertaining to software since its a multidisciplinary field, although each school has a different educational agenda in software design. The students in the Digital Design program has a core curriculum in design and need an auxiliary curriculum in CS as dictated by the requirements of the job roles that these students wish to pursue post-graduation. Teaching CS skills to non-STEM students has proven to be a difficult educational task that requires time and patience to find approaches that work.

The collaboration discussed in this paper involved an instructor from a VR computing course in our university's College of Computing and an instructor from a VR interaction design course in our university's College of Architecture and Design. The collaboration was set up for students from both classes to work with each other, while both instructors were able to meet their objectives. Specifically, students worked together on different project deadlines and due dates. For example, the computing course required most of the project to be completed by the third month of the semester, while the interaction design students were not expected to be done with their work until the first week of the fourth month of the semester, due to the differing end goals of each course. The computing students had to user test their games, collect feedback, and present the feedback for the final presentation at the end of their semester. In contrast, the interaction design students' final consisted of an exhibit at the end of the semester, where all groups were required to present their interaction designs to their college's student body, faculty, administrators, and family members.

Cross-collaboration between the computing and design courses produced highly rewarding results for both students and instructors. Although these collaborations are beneficial, they require extensive planning and compromise from both instructors. The first step for the two instructors was to understand what the priorities were for each of their courses. This provided a frame work for what priorities had to be met for both courses and how the students from each group can aid in those outcomes. For example, the computing professor expressed the importance of conducting user testing and having original art assets to be implemented into her students' work. This gave the design instructor valuable insight into how to schedule her course so that her students were available for user testing sessions with the other class and also how to manage her student's time so they could provide completed art assets for the computing students' projects. The collaboration was not mandatory but was encouraged, and some student groups in both courses were happy to exchange art assets for code, even if they were not working on the same project.

Class Structure and Goals

Computing Class: We structured the class around AR/VR projects that had various phases throughout the semester. Additional in-class activities were designed to support a deeper understanding of issues related to AR/VR in general and to assist students with their projects. Besides development, two additional projects required students to concentrate on reading course literature, class presentations and discussions, and written reports. The goals of the course were consistent with the general major's objectives, which are A) To be able to communicate effectively and critically, both verbally and in writing, on various topics in AR/VR; B) Be able to design, implement, and evaluate AR/VR application; C) Be able to identify user needs in the context of AR/VR applications.

Interaction Design Class: Students were given the task of creating an interaction design that could be presented within an interactive format at the end of the semester. Students were not required to have code for their designs, but they were required to have their work presented in a way so that the functionality could be easily understood. Options for the deliverable included paper prototypes, animations, videos or storyboards. Students were required to present concept sketches, flow charts, and 3D models of assets used in their applications and a PDF of their project work over the semester. The goal for the final exhibit for each student group was to explain their interaction design to attendees in a clear, and engaging way. They were instructed to pitch their ideas for further development.

Curriculum

Computing Class: The curriculum was based on understanding AR/VR technologies in relation to human physiology and perception. Class discussions were based on an overview of human sensorimotor systems and the related computer peripheral systems of visual, auditory, haptic, smell, and taste displays. Input devices and tracking techniques were also discussed. The second half of the course facilitated learning activities about system evaluation, definition of contingency and concurrency, human factors studies and user evaluation. Finally students had to discuss and understand the different levels of immersion, presence, and fidelity, and how users can be impacted by different artifacts such as simulation sickness.

Interaction Design Class: The curriculum was structured around the industry practice known as a design sprint. The design sprint process consisted of five phases: understand (market research and consumer needs), diverge (explore, develop ideas, iterate), converge (identify concepts that are relevant for the next iteration cycle and create storyboards of how application is used), prototype (create rapid low fidelity prototypes of the digital application with paper and invite classmates to test them out), and test (conduct more in depth user testing on the product or application).

To teach these design sprint phases, student groups were chosen to present their progress, including current stage of development. Observing students would offer advice to get to the next design sprint phase, to speed up the current iteration phase, and provided feedback on the design itself. Participation during these feedback sessions was mandatory for all students and part of their final grade. It was an important process in training the students to think like a designer, so they understood what to look for when evaluating a software application. Once they understood user experience (UX) principles and the design sprint, they were able to spot flaws not only in their classmates' work, but in their own. These feedback sessions where students expressed and exchanged information was central to forming meaningful class discussions and collaboration.

Student Groups

Computing Class: The primary development project required all students to find a group partner from the same computing class but some students were advanced and preferred to work alone. Many students formed groups with students from the interaction design class, which was encouraged but not mandatory. One student decided to work on an old project and nobody else wanted to participate in his project.

Interaction Design Class: Students had an initial meeting at the beginning of the semester where they pitched project ideas and exchanged feedback. Neither design or computing student groups voluntarily started a dialogue after ideas were presented, but after we paired up a few computing and design groups, the conversations started flowing. Three groups worked with each other over the course of the next few weeks. One of the groups remained in collaboration with each other until the final exhibition. The other groups worked with each other more sporadically, exchanging assets and code when needed.

Students Collaboration

Computing Class: Students had to share their workload and exchange capabilities with design students by helping them solve technical challenges in return for computer graphic assets. Communication happened through Canvas, Trello, GitLab, Slack, and Discord, although most communication was verbal and email. Successful groups spent a lot of time in the lab together, while one group that struggled did not communicate effectively, often failed to submit on time, and when probed could not answer whose responsibility it was to submit each time. Eventually, one student decided to drop the course and did not notify his partner on the decision. The entire class, a designated TA and the instructor tried to help the group to succeed but unfortunately, the outcome was not promising.

Interaction Design Class: Groups used Discord and Slack to communicate with messaging. When members were not were responsive in checking or answering, the teams found the most effective communication was in-person meetings. After observing group dynamics and their successes or failures, the amount of regular in-person communication was paramount in the successful collaborations. Students created shared Google Drives where either team could access the work when needed. This was more relevant for the design students because they were creating assets for the computing students to implement into their games. Some issues did arise from this sharing technique because design students were not always up to date with the latest versions of these assets, so the computing students would not know which assets to use.

Learning Management Software

Canvas was used as a primary tool to highlight and emphasizes various deadlines. Canvas modules were used to post a description of each major assessment component with a list of all important deadlines associated with each module. These two steps allowed students to plan ahead and know what to expect in terms of the workload throughout the course. Students were often asked to refer to these pages on Canvas. All assignments had to be submitted through Canvas as well and so the student could benefit from automatic reminders implemented in Canvas. In-class slides always started with several slides that showed upcoming deadlines for a week and what to expect in next few weeks. The slides were regularly posted on Canvas. Finally, Canvas was used to send course announcements so students will them via email.

Summative and Cumulative Assessments

Computing Class: Weekly discussions accounted for 20% of class grade. For weekly discussions, students had to present an article, facilitate class discussion, and write a summary on a Canvas board. The topic of interest also accounted to 20% of class grade. Students had to do a referenced research on a topic of interest, do a 7-minute presentation, facilitate class discussion, write a proper 3-page report. The development project accounted for 40% of class grade. For their project, students had to develop an AR/VR application in weekly or biweekly increments. They started with ideation and submitting the project's objectives document, then they had to submit and showcase their progress by supplementing video, pictures, and progress reports. Finally, students had to design, conduct, and analyze a usability study. At the end of the semester they can to summarize their work in a full report and presentation accounting for 20% of class grade.

Interaction Design Class: Rubrics were provided to students for each major milestone. Monthly grades were broken down into weekly increments that were progressively larger components of their class grade. For example, the first monthly sum was 10% of their class grade, the second monthly sum was 28% of the class grade, and the third monthly sum was 32% of the class grade. This grading method was meant to reflect the difficulty and expectations of students as the work became more demanding. The third month was the most intense of the semester because it led to the first week of forth month, when the demo hall exhibit took place. Other portions of grade included participation which was 10% of the class grade and presentation which was 20% of their class grade. The 20% presentation grade counted as a final exam for the class.

To ensure that students understood how their presentation would be graded, a rubric was provided that explained the three main areas of evaluation. These included final deliverables, participation, and presentation. The rubric was divided into quadrants which each explained a level of quality, from highest to lowest. These quadrants included exceptional, proficient, satisfactory, and unsatisfactory. Final deliverables consisted of all components of the final; visuals, research and demonstration of functionality. Students were instructed to complete their work in each of these categories for their grade. Much of this portion was visual work that had been done throughout the semester during the design phase.

RESULTS

At the end of the semester, the interaction design students demonstrated the results of their interdisciplinary collaboration in a demo hall exhibit. The exhibit was hosted on campus with affiliated faculty, administrators, students, and friends and family members of the students (see figure 1). Each student team had a designated area including table and wall space to properly articulate their project design processes and final deliverable. At the same time, computing students were given 5 minutes to present the overall journey of their development project, emphasizing what they have discovered during the user study phase and how the project would have been developed if they were given a phase two opportunity. They also had to submit their final written report and code via Canvas. Students from both classes were also given a survey of questions centered on the effects of the collaboration based on a five-point Likert scale. Survey results were quantified for analysis and additional comments gathered for review.

Computing Class Final Presentations

Although final marks for development projects were given at the end of the semester, students had to complete their games or applications one month before the end of the semester. This

Fig. 1. Attendee, testing class project at exhibit

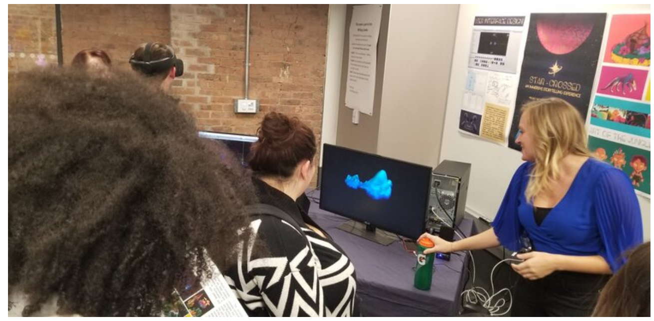

Fig. 2. Student showcasing class project at exhibit

was done so they will have enough time to test their projects on real users, get various feedback, analyze the feedback, write a reflection, and then finally present their entire process to the class. One group had to split into two groups because one of the students felt that he was doing all the work, yet at the end of the semester, the weaker student delivered a more complete project than the stronger student. One group failed to complete the project and did not deliver a final presentation. Overall, there were eight successfully completed projects. All students who delivered the final presentation spoke about their findings during the testing phase and how they would change the project if they were to continue to the next phase. One group worked over winter break to significantly improve their project. This group also planned to continue development during the spring semester and write a short paper describing their work.

Interaction Design Class Demo Hall Exhibit

Preparation for the Demo Hall Exhibit began during the second week of the semester. Students were informed of what their final project would be and how they were expected to present it. To help students in visualizing what this exhibit would entail, they were shown photos of similar events from previous years' exhibits. This helped the students solidify their ideas of how and what they would present at the end of the semester. Students were inspired by the photos of other student exhibits and had a visual goal to work towards throughout the semester. Students were instructed to have four forms of presentation: display, visuals, PDF, and video. These four forms of presentation made up the final presentation deliverables so that students could effectively communicate their work with concept sketches, paper prototypes, summaries of their research, and a short video of their application in use (see figure 2).

The visual portion included concept artwork, UX sketches, storyboards, and archived materials such as photos of previous versions or during user testing sessions in class. This helped explain what the process was like throughout development through visual artifacts. Paper prototypes were an essential part of development throughout the semester. Students made prototypes of their application with paper and cardboard mockups, which allowed them to rapidly test their projects with classmates. These prototypes were integral in the final products and served as great conversation starters between students and attendees. It also added a unique element of interaction, as attendees enjoyed playing with the paper prototypes almost as much as they did playing the functioning game demos.

Research was a major component of this course, as students had to develop user personas, case scenarios, and market research. Students were required to run frequent user testing and feedback sessions where they would take notes of observations and feedback. Students were directed to form a Design Document from the beginning stages of development. Students presented this portion of their project through slide presentations or printed booklets for attendees to browse through during the exhibit. The video portion of the final presentation was meant to serve as a way to demonstrate the functionality of the final product. Because students were not required to have a functional demo, they needed an animation of the functionality so that the interaction design was accurately represented. The video portion would also serve as a strong portfolio piece for students to provide a quick, concrete demo of their interaction design project.

The display portion of their exhibit included the overall branding of their project. Students were told to present as though pitching to potential investors or studios, so how they present their project was an important part of their final. They were expected to have figured out who their target audience was during the UX phase of the semester, where personas and user scenarios were outlined, then apply this to the marketing portion of their project. For example, one group decided that their product was for children but also parents, so the look and feel of their product needed to appeal to both user groups. Another group realized that their users were most likely teenagers and young adults, so the look and feel of their product needed to appeal to these user groups, without being too childish or too mature. The branding materials needed to reflect these carefully thought out decisions. These materials included banners for the tables, posters, stickers, and business cards. Once the branding materials were designed later in the semester, students were instructed to design invitations for the event. One group even had 3D prints made of the two main characters in their VR experience, which they had in their final exhibit display.

End of Semester Collaboration Survey

The survey was designed as a quick questionnaire with ten questions ranked on a five-point Likert scale and an open-

Question number	What impact did collaboration have on	Design Class Average	Computing Class Average	Class Average Difference	Design Class Std Dev	Computing Class Std Dev	Class Std Dev Difference
1	Your learning	4.23	3.5	0.73	0.44	0.85	0.41
2	Exposing you to new ideas	4.38	3.4	0.98	0.51	1.17	0.67
3	Expanding your knowledge base	4.31	3.5	0.81	0.63	0.97	0.34
4	The quality of the class	4.15	3.6	0.55	0.69	1.07	0.39
5	Your preference for future collaboration	4.31	3.9	0.41	0.63	1.29	0.66
6	Your interest in tech and engineering	3.92	3.9	0.02	0.95	0.74	-0.22
7	Your knowledge in tech and engineering	3.46	3.8	-0.34	0.88	0.79	-0.09
8	Your interest in design and art	4.46	3.6	0.86	0.66	1.17	0.51
9	Your knowledge in design and art	4.38	3.5	0.88	0.65	0.97	0.32
10	The quality of your project	4.17	3.75	0.42	0.58	0.89	0.33
AVG	**The average of the 10 responses**	**4.18**	**3.65**	**0.53**	**0.66**	**0.99**	**0.33**

TABLE I
SURVEY RESULTS TABLE - CLASS AVERAGES AND STANDARD DEVIATIONS

ended response at the end, asking for additional comments. All questions involved the student's perspective on the impact level of an aspect of collaboration between the classes. The students were asked to rank the impact of each area from strongly negative, negative, neutral, positive, and strongly positive. Question responses were ranked on a five point Likert scale from strongly negative (1), negative (2), neutral (3), positive (4), and strongly positive (5). The questions asked, "What impact did collaborating with students from another discipline have on..."

1) Your learning
2) Exposing you to new ideas
3) Expanding your knowledge base
4) The quality of the class
5) Your preference for future class collaboration
6) Your interest in technology and engineering
7) Your knowledge base in technology and engineering
8) Your interest in design and art
9) Your knowledge base in design and art
10) The quality of your project, if you did collaborate

13 design students and 10 engineering students responded to the survey (see table I). Four design students and six engineering students gave additional comments. Responses from the design class averaged 4.18 with a minimum of 3.46 and a maximum of 4.46. Responses from the engineering class averaged 3.65 with a minimum of 3.4 and a maximum of 3.9 (see figure 3). The differences between the population response averages were 0.53, with the design class having higher, more positive, responses.

The two class's response rates for the comments sections were similar. Students' free-response comments were mostly positive, reinforcing the results from the Likert scale questions. Comments that were critical of the collaboration usually

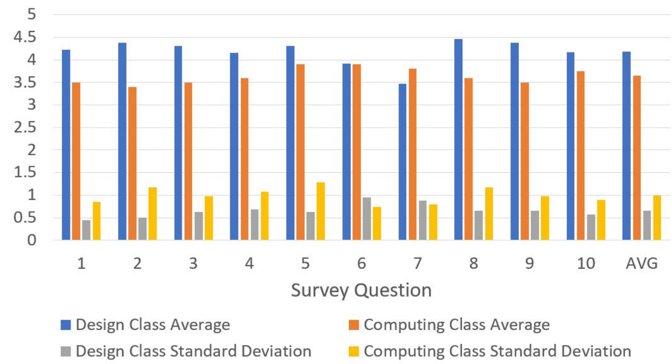

Fig. 3. Survey Results Graph - Class Averages and Standard Deviations

included constructive feedback to help the instructors improve the collaborative experience. Positive comments involved completing project deliverables successfully and being emotional content with the results. Students writing positive comments were content with how the collaboration impacting their educational experience. They found it interesting to work with students that have a different perspective on the shared project goals. Some even from friendships across the colleges that hopefully will result in future collaborations and stronger student projects that will increase their employability as they graduate from college. Some sample comments from both classes:

- "It was interesting to learn alongside IT students. They helped us implement a working demo at the demo hall, which we would not have had without them."
- "The DD students were amazing to work with. I absolutely loved working with them!"
- "I acknowledge that some groups didn't have the best common interests as a team, but my group was a blast to work with, so I am looking forward to have more collaboration with IT students."

Negative comments involved missed deliveries, miscommunication, lack of work execution and unclear expectations. Students writing negative comments were not content with how the collaboration impacted their project performance, yet did not talk about the negative collaboration experience having a negative impact on their educational experience. They found that the major obstacles to successful collaboration were communication between team members, clear definition of goals and milestones, and the structure around the collaboration between the classes. Most indicated that the process of exchanging fair pieces of work between students with different backgrounds was the hardest part of the collaboration. Some sample comments from both classes:

- "Very difficult to communicate with the DD students, and limited how much control we had over our finished project, which added confusion."
- "Only do collaboration if students work only one project. Doing portions of work in exchange didn't work out."
- "Have students collaborate with other dedicated students

to finish set requirements near the end of the semester. This way they will not back out in the end."

- "It would be helpful to have a more structured collaboration. Clear indication of expectations and schedules."

Discussion

Survey Interpretation & Observations

Design student responses were clustered around four out of five points, which is a positive response to the collaboration structure of the class projects. Computing student responses were clustering a half-point lower, around 3.5, which is still positive but getting close to a neutral response to the collaboration's impact on their class experience. Class averages per question stay close to these total question average indicating that these total averages are accurate of each class population's experience to the collaboration aspect of the class structure.

The differences column of the results table reinforced this interpretation. Each question has the design class average about half to a whole point above the computing class average. Exceptions to this pattern are seen in the two questions about students' interest and knowledge base in technology and engineering. The two-class averages are equal in regards to interests in technology and engineering, while the computing class shows a small 0.3 point increase to the design class in the knowledge base of technology and engineering. That should not be surprising since one would expect the computing students to feel that they have a stronger knowledge base in technology and engineering over design students.

It is interesting to see that the design students showed a stronger interest in their primary field of design and art based on collaborating with computing students. The design students' responses showed that their own interest and knowledge base in design and art was impacted more than their own interest and knowledge base in technology and engineering when collaborating with computing students. At first glance, one might think that the opposite should be true, but upon further reflection, that pattern does reflect the reality of the situation since the design students would have focused their efforts on the projects more on the design side if they collaborated with the computing students successfully.

Teacher Observations

Computing Class: The class had achieved all of its objectives. There was significant improvement in students' understanding of the course material. Students has demonstrated very high level of mastery of the course material in their Topic of interest presentations, discussions, and report. They also had very through discussions on their exams. In terms of their development projects, all students have delivered a stand alone applications that they can proudly showcase in their portfolio. Over the week, the instructor has observer also growth in the student's technical skills. It was also interesting to see who students increased time spent in the lab while working on the project, they also made an effort to help each other, if had to face technical challenges and overall very supportive of each other progress. Part of this was that there was no competition between the students but that they actually wanted to try out each other's games. Finally, many students have showed a great level of maturity and were able to resolve issues both related to collaboration with design students and technical challenges.

Interactive Design Class: The outcome of the Demo Hall Exhibit was positive among the students and attendees. Students had the chance to share their work with the public and meet new people from other departments. Three groups who maintained collaboration with IT students presented their end product with their collaborators. This was highly valuable to students as they were given the opportunity to receive feedback from people who they did not know and had never seen these projects. It also was rewarding for students to be able to invite friends, family, and faculty from other departments. This served as an opportunity for our department to show others within the University what we do and also to prove to both IT and DD students the possibilities of collaboration.

A. Lessons Learned

We learned how to work together across their classes in more impactful ways. We learned to work together and develop stronger multi disciplinary approaches by implementing individual assets done bi-weekly into coded interactions. The design students supplied computing students with the decided upon deliverables according to the contract which they all sign during the 2nd or 3rd week of class. The deliverables and timeline are confirmed. If any changes are made to the project and the deliverable agreement has changed, then the team has to re-write their contract so that it is updated and there is documentation that everyone agreed to those changes.

This experience was greatly insightful into how both disciplines work with each other and how they communicate. Communication seemed to be the biggest obstacle, between the computing and design students. During an end of the semester debriefing session, one of the student students expressed confusion and frustration with the design students because they had completely different notions of the length of time it takes to complete art assets. This misconception led to interpersonal issues between the two groups and even a threat to discontinue collaboration unless more assets were produced within the preconceived acceptable amount of time.

The design students overwhelmed the computing students with tasks because they had underestimated how much time was required to program each part of their application. This contributed to potential conflict points between design and computing students. Clarity and assertiveness was also an issue. While the design students were more forward with things they wanted and upfront with what they realistically could and could not do, the computing students were generally more timid with what they knew they could and could not do.

Study Limitations

We did not have a pre-assessment given at the beginning of the semester, which we will do for future studies. The two classes were not aligned when we designed the curriculum. This made it difficult for us to understand the quality of the collaboration mid-semester, and had to wait until the end of the semester. To cover the worst case scenario, we should have had a plan ready when collaboration did not happen.

Future Work

We have agreed to continue the collaboration between the classes for the following semester. Before the semester begins, we should meet and discuss priority deadlines, objectives and outcomes. Once this information is understood, both instructors can more easily align their course schedules so that these priorities are met on both sides. Next, a skills and interests survey should be sent to both classes before the semester, so that we can more accurately predict what kind of projects can realistically be done in their individual classes. More importantly, we can predict collaborative efforts between both classes based on what each group is lacking. For example, if the design instructor knows what the majority of the class is skilled in, it helps the computing instructor in developing projects for the computing class and vice-versa. If the computing students want to develop games with advanced computer graphic needs, but only a small amount of design students have those skill sets, then the project goals for the computing students will have to be adjusted. If the majority of the computing class specializes in a particular game engine such as Unity rather than Unreal, then the design students will need to be prepared to design assets for this software.

The first day of class should be an introduction to both career groups; computing and design. This is so that each group is informed about the value and relevance of the others' field. It is also meant to grow a mutual respect and interest in each others' fields so that there is less room for misunderstanding certain industry terms or processes. There will be a glossary, where students from both groups will help to answer the terms of certain words, according to their field. For example, the word 'storyboard' may mean different things for each respective field, which could refer to technical flow charts to computing and illustrated flow of events within a story for design. This should help as an initial ice-breaker to encourage communication through both courses. There will be a 'tips' page that talks about advice for working within a team or working with a person from another field. These tips will include: transparency, regular updates and communication, compromising and teaching (the other about their workflow). It should also make a point for honesty, so that when one group asks another for something, the other group is clear about how skilled or unskilled they are in that area.

Conclusion

Having computing and design students work in cross-class collaboration projects is a difficult and rewarding experience for undergraduate instructors. At our university, we have designed a collaboration curriculum over the last few years. We have informally observed that such approach improves students' in classroom performance year after year. However, this is a first time when we measured and documented our experience. Our work shows that it is a necessary class structure to successfully educate future developers and designers, and we wish to share our experiences with the larger STEM educational community.

Acknowledgment

We thank the administrators, faculty, and students involved with their support of this research. Any opinions, findings, conclusions, or recommendations are those of the authors and may not reflect the views of any of these parties.

References

[1] J. P. Gee, "What video games have to teach us about learning and literacy," *Computers in Entertainment (CIE)*, vol. 1, no. 1, pp. 20–20, 2003.

[2] M. J. Lee, "How can a social debugging game effectively teach computer programming concepts?" in *Proceedings of the ninth annual international ACM conference on International computing education research*. ACM, 2013, pp. 181–182.

[3] D. Bavelier, C. S. Green, A. Pouget, and P. Schrater, "Brain plasticity through the life span: learning to learn and action video games," *Annual review of neuroscience*, vol. 35, pp. 391–416, 2012.

[4] U. D. of Education, "Teaching - Office of Educational Technology," Tech. Rep., 2018.

[5] A. Gurria, "PISA 2012 Results in focus: what 15-year-olds know and what they can do with what they know." Tech. Rep., 2012.

[6] M. Moore, "Work in progress: Teaching-obstacles in higher software engineering education," in *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2017, pp. 25–28.

[7] S. Gokhale, "Leveraging avid use of technology in software engineering education," in *2018 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2018, pp. 125–132.

[8] N. Pancratz and I. Diethelm, "Including part-whole-thinking in a girls' engineering course through the use of littlebits: A practical report on including part-whole-thinking into the content of computer science education," in *2018 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2018, pp. 1134–1138.

[9] M. Zhu, N. Panorkou, E. S. Lal, Pankaj, E. Germia, P. Iranah, B. Samanthula, and B. Debasmita, "Integrating interactive computer simulations into k-12 earth and environmental science," in *2018 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2018, pp. 220–223.

[10] C. Wang and M. Frye, "Minigems 2018 summer camp evaluation: Empowering middle school girls in steam," in *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2019, pp. 149–155.

[11] D. Seo and M. Lawrence, "Workshop to increase women's enrollment in technology discipline at the community college," in *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2019, pp. 160–162.

[12] 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.

[13] H. Shibata, H. Kasiwagi, Y. Motomura, and K. Ohtsuki, "A learning model of computer science education for liberal arts students," in *2012 IEEE Frontiers in Education Conference*, vol. 2. IEEE, 2012, pp. 1–7.

[14] S. Kim and N. Stogdill, "Inspiring high school students to engage with current issues in science and medicine with an interdisciplinary course entitled science and literature of disease," in *2018 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2018, pp. 1–7.

[15] M. Frydenberg and D. Andone, "Does creating shared projects in virtual reality capture students' interest in technology? an international project in stem education," in *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2019, pp. 311–315.