Evaluating Student Spatial Skills Learning in a Virtual Reality Programming Environment

Justin Lu, Lauren Seavey, Samuel Zuk, James Dimino, and Fred Martin
Kennedy College of Science, University of Massachusetts Lowell
Lowell, Massachusetts, USA
{Justin_Lu, Lauren_Seavey, Samuel_Zuk, James_Dimino}@student.uml.edu, Fred_Martin@uml.edu

2022 IEEE Frontiers in Education Conference (FIE) | 978-1-6654-6244-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/FIE56618.2022.9962403

***Abstract*—In this research full paper, we examine students' improvement in spatial visualization skills when using MYR (short for "My Reality"), a browser-based, cloud-hosted programming environment for beginning through advanced programmers to create immersive, three-dimensional virtual reality scenes.**

The research literature suggests that there is correlation between students' spatial abilities and their success in programming. In this study, we conducted a three-week (six hour) virtual after-school program which introduced high school students with different programming backgrounds to coding in MYR. During the program, students learned the basics of MYR, introductory CS topics, and completed individual coding projects, creating an original MYR scene.

A study examined changes in students' spatial reasoning as a result of this intervention. The students' performance in spatial skills was measured using the Revised Purdue Spatial Visualization Test: Visualization of Rotations (Revised PSVT:R) [1]. Students completed this instrument using a pre/post survey design. We analyzed the impact of the intervention using a paired samples T-test. We further developed a rubric for analyzing the sophistication of students' MYR code and applied it to evaluating the programming expertise of our study participants.

The program was hosted twice with two different groups of high school students. Most showed interest in MYR programming and expressed their creativity and learned skills in their original project. With the first group of students, we found increases in their spatial visualization performance after the intervention with MYR, though statistical significance was not reached. The second group of students had higher baseline prior experience in computing and spatial visualization skills; these students did not further increase in their spatial visualization skill.

The analysis showed that the MYR has a potential to improve spatial skills and engage students' interest in computing. We recommend that MYR and related computational environments be further studied and made available to students.

***Index Terms*—Computer science education, Spatial skills, Virtual reality**

I. Introduction

MYR is a browser-based, cloud-hosted programming environment for beginner through advanced programmers to create immersive, three-dimensional virtual reality scenes [2]. The environment allows for creativity and imagination to be expressed in virtual reality scenes using simplified JavaScript based programming.

MYR provides a basic three-dimensional geometries such as cube, and sphere. To build a scene, users change the geometries' properties such as position, scale, rotation, and color, combining multiple of them to create a object they desired. After the user has completed their scene, they can use a virtual reality (VR) headset or a phone with a VR "cardboard" viewer to explore and immerse themselves in their own scene.

Within MYR, there are courses and tutorials on creating scenes. The most popular beginner course in MYR teaches how to create an ice cream cone, as shown in Figure 1. This introductory course demonstrates how to build the ice cream cone using a cone and a sphere. The course introduces MYR's use of a cursor, a hidden global object that holds state of the geometry about to be drawn. In the example, Lines 1 through 4 of the code establish the properties of the cone about to be drawn: first rotating it 180 degrees around the X axis so that the tip of the cone faces downwards, and then setting its position, scale, and color before drawing it at Line 5. Similarly, Lines 7 through 10 define the position, shape, and color of the sphere representing the scoop of ice cream, with line 9 setting it as a slightly oblong sphere, to better look like a real scoop.

The development of MYR itself was inspired by Papert's ideas of learning through "constructionism," where students learn skills building on their existing knowledge and interests, expressing their social or cultural identity as they create and share computational artifacts [3].

II. Background

A relationship between spatial skills (SS) and success in the STEM disciplines, particularly engineering, has been demonstrated over multiple studies (e.g., [4], [5]). Regarding computer science, in 2018, Parkinson and Cutts developed a model based on the six-factor analyses done by Carroll [6] to draw the connection between different visualization factors that can be developed through spatial training and how those contribute to different aspects of the programming skills [7]. The model used the five factors of SS analyzed by Carroll and connected them to two core sets of programming skills, program comprehension, and generation, with detailed aspects of each skill in between.

Parkinson and Cutt's model describes how each visualization factor contributes to a specific aspect of programming skills. One SS factor to highlight is spatial visualization, which is the ability to mentally rotate and transform a structure. Spatial visualization contributes to the construction and transformation of the mental model, which transfers its skills to identify the core plan for the problem and expand to generate the program. With the main process of building scenes in

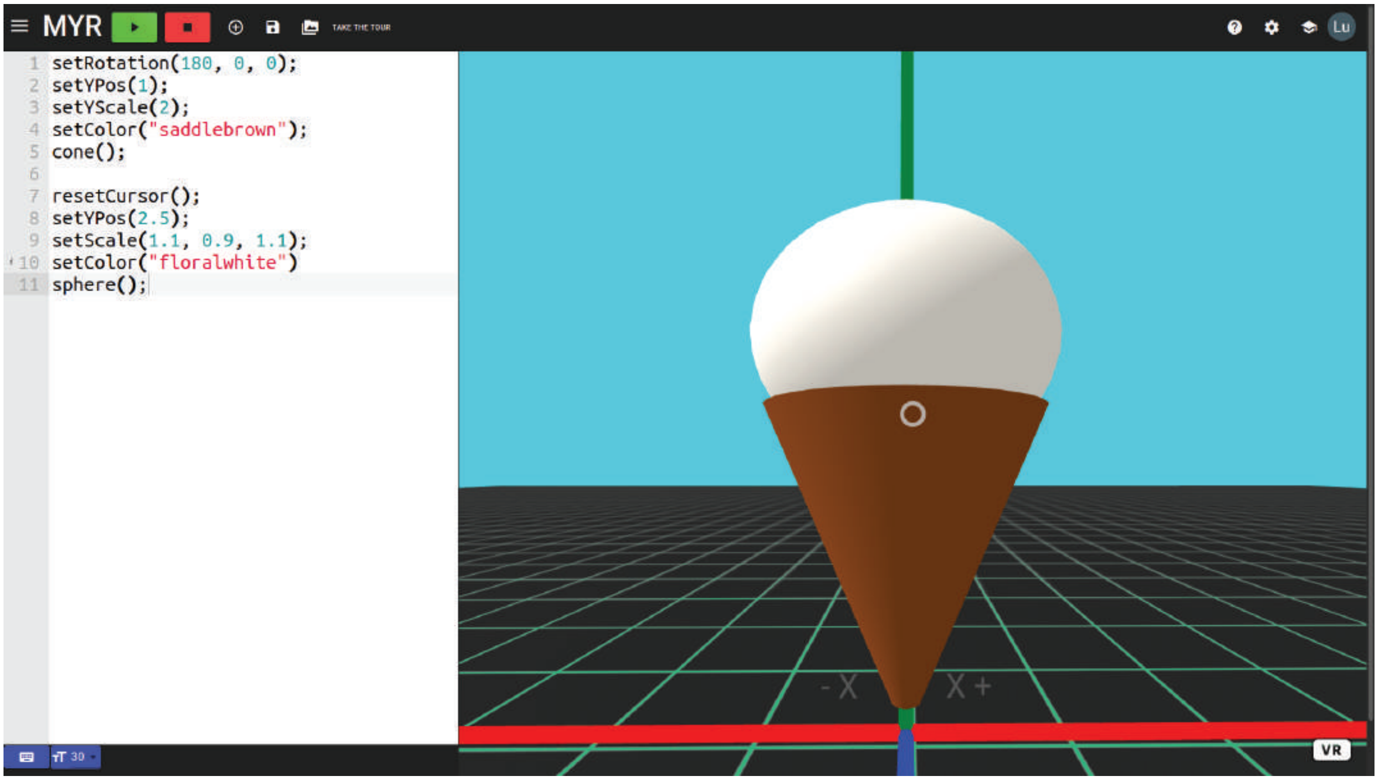

Fig. 1. An ice cream cone with associated MYR code

MYR to visualize how the objects in the scene should be transformed and placed programmatically, we highlight spatial visualization specifically to be the factor that contributes to the development out of the six.

Training of SS has been done over the decades with different methods of intervention. An early example of intervention is a study done by Sorby and Baartmans in 1993 where they ran a 10-week college-level course at Michigan Tech University (MTU). The course was offered to students who scored low on a SS exam to train their SS, primarily through sketching geometries with a different rotation, reflection, and more through traditional pen and paper with measuring tools [8]. The method was concluded to be successful and showed a statistically significant increase in students' scores in their engineering discipline [8]. As technology progressed, SS training shifted to the use of digital tools. In early 2000, MTU added CAD and computer software along with sketching to their SS training course [9].

Other studies focused on the computer science discipline. Research by Cooper *et al.* in 2015 and Bockmon *et al.* in 2021 showed an increase in students' grades, as well as their spatial test scores after training [10], [11]. A variety of spatial training tools have been developed to increase students' spatial skills, such as a video game [12] and a VR platform [13].

III. Intervention

An after-school program was designed and conducted virtually using the Zoom video-conferencing application. The program embodies well-understood principles in the design of constructionist coding environments. Per Papavalsopoulou *et al.*, it is important to pay attention to time structure, avoid cognitive overload, provide for social interaction, support children's motivations, and have involved instructors [14]. This is further discussed below.

We held six one-hour sessions over three weeks. Each of the sessions included mini-lectures and practical work in breakout rooms. Owing to the program being hosted online, we recognized the need to manage students' attention spans and therefore kept each session to just one hour [15]. The study's researchers conducted the sessions and provided assistance to study participants during the breakout room activities.

During the first four sessions, we taught the introductory concepts of computer science (e.g., variables, loops, conditionals, and functions) through building scenes in MYR and using live coding. The sessions were focused on introducing new ideas in a structured fashion, so as to avoid student cognitive overload [14]. In the fifth session, creative coding activities were introduced to students. The final session was devoted to students sharing their original work products.

In preparation for that last session, students were to create their scene using MYR, displaying their knowledge learned from previous weeks as well as expressing their creativity. Students shared the project they created along with an explanation of how the scene was coded, their intention behind their code design, and the difficulties they encountered.

Most of students' engagement with MYR was on the

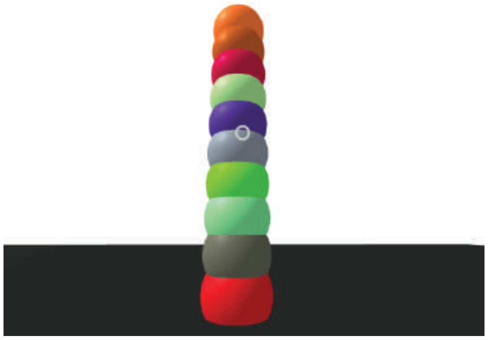

Fig. 2. A stack of ten spheres

computer screen, which rendered scenes as 3D graphics (like an immersive computer game). Some students may have also used VR equipment on their own at home. This usage itself was not part of the study.

Students were recruited by email invitations to computer science high school teachers, via the last author's network of teacher-members of the Computer Science Teachers Association. Participants invited were restricted to matriculated high school students. Based on the available network, students were located across two states in the Northeast USA and had various ranges of experiences with programming.

Student registration for the study was a two-step process beginning with parent consent for the study procedures and concluding with student assent for the same. Upon completion of this, students were provided with a link to a study-specific Discord server (an instant messaging platform), which served as the hub of the program.

Discord was a primary place for students to receive information about the program: to submit and share their work and as a general chatting place for students to ask and discuss the content of sessions. We encouraged students to indicate their previous experience with the coding by self-categorizing into three levels (beginner/Code Apprentice, intermediate/Code Chef, and advanced/Code Wizard). Thus, Discord was also the setting for "social interaction," a valuable component of learning per [14].

The subsections below present more detail about the program design.

A. Week 1

The first session introduced the basic features of MYR, how to create a shape onto the scene using an editor, and how to transform those shapes using the MYR API and a cursor system. We taught them by creating an ice cream cone (Figure 1). This scene introduced how to create simple shapes and how to apply the transformation using minimal lines of code to not overwhelm the students. After the lecture, students were tasked with changing the color of the ice cream to their favorite flavor and scaling the cone to be more visually appealing.

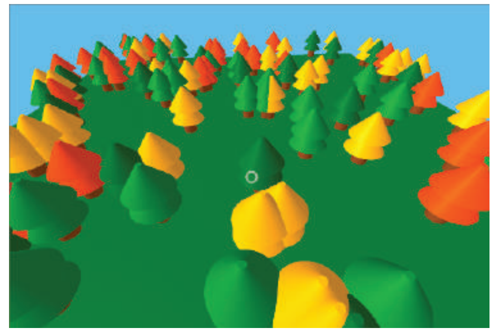

Fig. 3. A forest with random trees

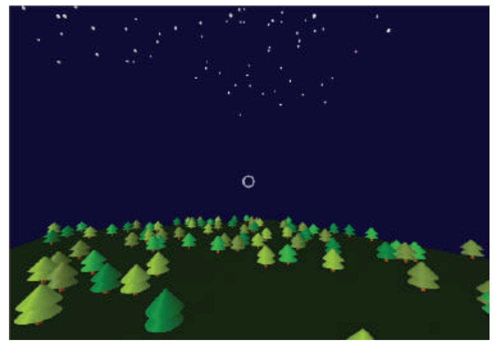

Fig. 4. A forest with twinkling stars

In the second session, the concepts of function, variable, and loop were presented. These concepts simplified the challenge of creating a group of ten spheres (more ice cream scoops) that were displaced by one unit on Y-axis with a random color (Figure 2). In an iterative fashion, we showed how to create a single sphere, move up the position by one, and repeat. Then we showed that it can be simplified by moving that single section of code into the function and calling it ten times to reduce the redundancy. The code was further optimized with the introduction of a loop and a variable to create a counter to track the number of function calls and the condition to check when the counter exceeds the number of desired spheres.

B. Week 2

For the third session, students learned how to create a forest, randomly placing each tree using function parameters (Figure 3). We introduced and explained a supplied function that takes X and Z parameters, which created and positioned the tree respect to X-Z plane. Randomness was introduced using JavaScript's built-in `Math.random()` function, which returns a pseudo-random number between 0 and 1. Students were taught how to arithmetically manipulate this value by scaling and offsetting to position trees across the MYR drawing plane.

On the fourth session, we taught two advanced MYR features: groups and animations. The group is a feature that allows grouping together a series of shapes, which then can be transformed into a single function. The forest scene from the last session was used to introduce animation by creating stars

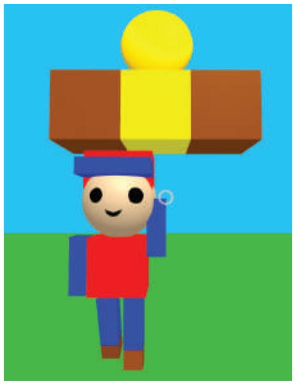

Fig. 5. Video Game Character from Spring Group

in the sky with each star having fading animation to simulate the twinkling (Figure 4).

C. Week 3

On the fifth session, a round-robin style coding activity was done. Students were assigned to breakout rooms of three to five students. They were given the option to modify a starter scene provided by us or create a scene from scratch. After five minutes, students switched their scene with another person in the group, extending work on the scene they received. The process was repeated until the scene was back to the original creator. The purpose of the activity was to stimulate students' creativity and apply the knowledge they learned from previous two weeks. After the in-class activity, students were introduced to the final project challenge and invited to continue the work they had started or begin something new.

On the final session, the entire period was dedicated to students presenting their final project work. Students presented their project by first showing what they made by moving around the scene, then students explaining the code they wrote, detailing how the general structure of their code and any struggle they faced while working on the project. We present examples of the students' work in detail in the next section.

The program was hosted for free. All participation was voluntary for students. They were given an incentive to continue joining the sessions by rewarding them with a T-shirt for all who joined at least four of six sessions, shared a final project, and completed the pre- and post-surveys to foster "children's attitudes and motivation attributes and motivation. The program was run a total of twice, once during Spring 2021 and once during Fall of 2021.

IV. Student Work

This section highlights three scenes from each of the Spring and Fall groups, illustrating a variety of creative, computational, and mathematical approaches.

A. Spring Group

Spring Group Scene 1 is shown in Figure 5: a representation of a famous video-game character. The code associated with this scene is basic; no functions, loops, or MYR groups are used. The student used a trial-and-error approach of writing code, running it, and making minor adjustments to position the shapes in their intended locations. The scene reflects the abilities learned in the sessions such as scale, position, and color adjustment. Although the scene is rudimentary in terms of its coding complexity, it is one of the more original scenes that were submitted and it was evident that the student took care in its development.

Fig. 6. Town Scene from Spring Group

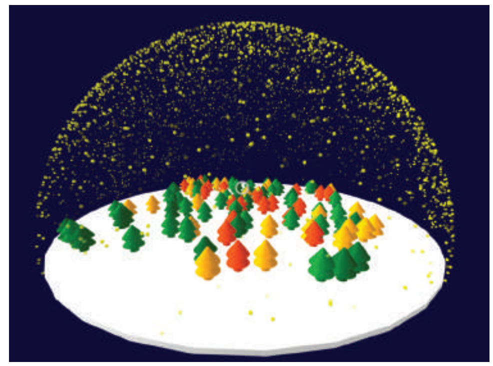

Fig. 7. Snow Globe Scene from Spring Group

Spring Group Scene 2 is a neighborhood with detailed houses, trees, and a sun that moves up and down (Figure 6). The code associated with this scene uses functions to define and place houses and trees with position arguments and uses loops to call and place multiple houses and trees in different locations. Animations were used to simulate the sunrise. During the presentation, the student expressed their frustration with difficulties they encountered, such as positional errors with respect to the roof, or the windows of the houses not having consistent color (the brown houses had reddish windows while the blue houses had white windows). The student explained how they tried to remedy the discoloration with a few different techniques but was unable to solve the problem. Despite not being able to solve the problems, the student demonstrated the ability to identify a "bug" within their code and potential causes for these bugs along with the hopes of eventually finding the solutions.

Spring Group Scene 3 is one of the most mathematically advanced scenes across both groups of students. The scene is

```
let r = 56.5685424949; // this radius will touch the far
    coners of the plot
let xPos = (Math.random() * r*2) - r; // Using the radius
    finds the point of x
let zPos = ((Math.random() * Math.sqrt(r**2 - xPos**2)) -
    Math.sqrt(r**2 - xPos**2) / 2) * 2; // Makes the
    spawning area of the stars into a circle; y will use
    √radius^2-x^2 to find its point
let yPos = Math.sqrt(r**2 - (xPos ** 2 + zPos ** 2)); //
    finds the heigh using  ythagorean theorem
```

Fig. 8. Snow Globe’s code for randomly placing stars using sphere equation

an extension of the forest scene in that it took advantage of randomness. The student used their knowledge of geometry to position the stars on the surface of a sphere, which then appeared as the stars filled it (Figure 7). The student used the circle equation to find random points on the X-Z plane with the Pythagorean theorem to find the possible Y-position out of the determined X-Z position and radius of the sphere (Figure 8). This project showed the student’s ability to apply their existing mathematical knowledge to newly learned skills to create a unique MYR scene.

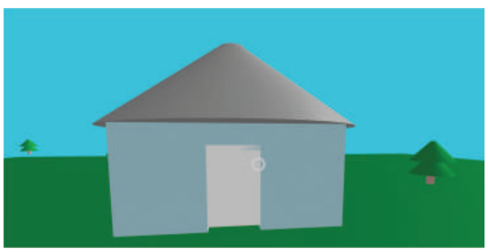

Fig. 9. Small House from the Fall Group

B. Fall Group

Fall Group Scene 1 is a basic house with green grass and a few trees (Figure 9). While simplistic visually, this scene demonstrated the student’s command of MYR groups, for/while loops, and randomness. The interior of the house was decorated with a bed and bureau.

Fall Group Scene 2 is “Three Wonders of the World,” which consists of the Eiffel tower, a pyramid, and a UFO abducting a cow (Figure 10). Aside from demonstrating the student’s

Fig. 10. Three Wonders of the World from Fall Group

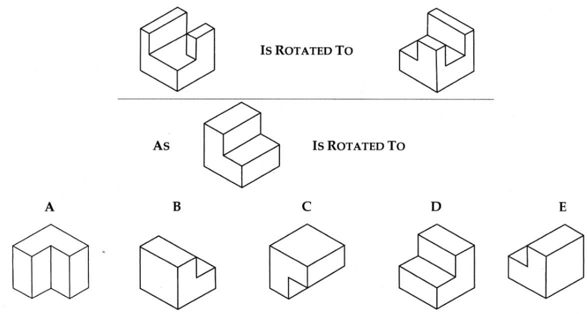

Fig. 11. Tutorial, non-scored item from PSVT:R instrument

sense of humor, this scene showed advanced programming techniques, including the use of MYR groups, animations, and custom functions with parameters. For example, the cow was made to spin and a beam from UFO moved upside down to simulate abduction. The student also used comments throughout the code.

Fall Group Scene 3 is a recreation of the Titanic (Figure 12). The ship showcases the student’s attention to detail and an advanced technical approach. The windows, decking, hull shape, tilted smokestacks, and lighting reveal the student’s visual abilities and coding fluency. The student created their own functions with parameters and used MYR groups, animations, randomness, and for/while loops.

Across both the Spring and Fall groups, these scenes show us that no matter what technical background any of the participants had, they were able to produce their own MYR scene through the material that we covered during the Zoom sessions and employing prior knowledge that they had coming into the program.

V. Data Collection

To quantify participants’ spatial skills, the Revised Purdue Spatial Visualization Test: Rotation (PSVT:R) by Yoon was used [1]. The Revised PSVT:R is a 30-question multiple-choice test with each question depicting the rotation of a 3D object. A different object is then displayed; the participant is asked to choose one of the five choices that depict the result of applying the initially depicted rotation to the new object. Figure 11 shows the tutorial, non-scored item from the instrument; the correct answer is “D.”

The PSVT:R instrument was imported to Qualtrics, and survey links were emailed to the participants at the beginning of the program (pre) and after the program was finished (post). Participants were given two to three days to complete the surveys. A separate demographic survey was included in the pre-survey email. The identical PSVT:R instrument was administered for both pre and post. Students were not allowed to review their answers. (Individual scoring results were shared with participants upon request after the study had concluded.)

To protect the privacy of the participants, participants were given a unique study code word, which they entered into each survey. The entire study was approved by the university’s Institutional Review Board.

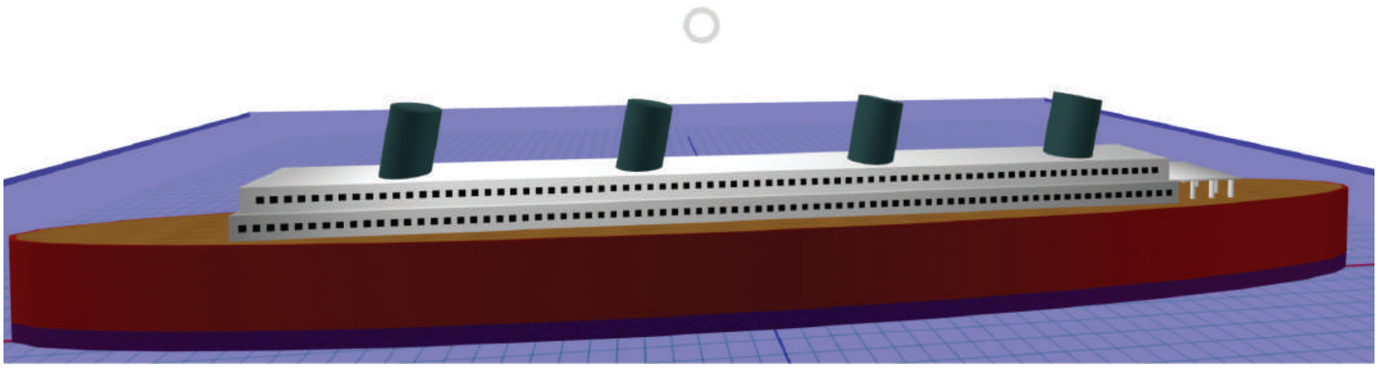

Fig. 12. The Titanic from Fall Group

TABLE I
DEMOGRAPHICS OF COMPLETED PARTICIPANTS

	Spring	Fall
N	9	13
Average Age (Years)	15.7	16.2
Gender		
Male	4	7
Female	4	6
Non-Binary	1	0
Race & Ethnicity		
White	5	6
Asian	2	4
Black/African American	0	1
Other	2	2

TABLE II
PRE-POST MATCHED PAIR CORRELATION FOR THE TWO GROUPS

	Correlation	Significance (1-tailed)
Spring (N=9)	.825	.003
Fall (N=13)	.944	$< .001$

VI. DATA RESULT AND ANALYSIS

For the Spring Group, we received nine participants who completed the program and all pre/post surveys. For the Fall Group, 13 participants completed the program and surveys.

A. Student Demographics

For the nine participants in the Spring Group, the average age was 15.7 years, with four males, five females, and one non-binary student. This group's ethnic identities included five White, two Asian, and two students identifying as Other. The 13 participants in the Fall Group had a slightly higher average age of 16.2 years, with seven males and six females, and the ethnic identities including six White, four Asian, one Black/African American, and two students identifying as Other (Table I).

TABLE III
PSVT:R RESULTS FROM SPRING GROUP (N=9)

	Mean[a]	Standard Deviation	t-value	Significance (1-tailed)
Pre	16.33	6.04	1.612	0.073
Post	18.22	4.15		

[a] The maximum score is 30

TABLE IV
PSVT:R RESULTS FROM FALL GROUP (N=13)

	Mean[a]	Standard Deviation	t-value	Significance (1-tailed)
Pre	23.85	6.20	0.940	0.183
Post	24.38	6.74		

[a] The maximum score is 30

B. PSVT:R

We analyzed the difference in pre- and post- results of PSVT:R from each group. Paired-sample T-tests were separately performed on each of the Spring and Fall groups using IBM SPSS software. We used the 1-tailed p-value as we did not hypothesize that our methodology could worsen the students' spatial skills.

Matched pair correlations for the two groups are shown in (Table II). Both groups demonstrate high correlation values (.825 and .944) with strong statistical significance, indicating that our students treated the spatial skills assessment seriously and confirming that the PSVT:R is a valid instrument.

Students in the Spring Group improved their pre/post average PSVT:R scores from 16.33 to 18.22, an increase of 1.89 on the 30-point scale of the instrument. The standard deviation of the scores closed from 6.04 to 4.15. However, the resulting p-value of 0.073 indicates that, while promising, the improved PSVT:R scores cannot be declared statistically significant (Table III).

Students in the Fall Group had a much higher baseline (pre-test) result from the PSVT:R instrument, scoring an average of 23.85 of 30 points (compared to the pre-test average of 16.33 from the Spring Group). The Fall Group showed a tiny improvement on the post-test, yielding an average of

TABLE V
COMPARING PSVT:R SCORE BETWEEN FALL AND SPRING GROUPS

	Mean Difference	t-value	Significance (1-tailed)
Pre	7.513	-2.82	0.005
Post	6.161	-2.61	0.008

TABLE VI
CRITERIA FOR FINAL PROJECT EVALUATION

1	Made Own Functions
2	Own Function has a Parameter
3	Used Groups
4	Used For/While Loop
5	Used Animation/Physics
6	Used Randomness
7	More than 30 Lines of Code
8	Comments in at Least 3 Locations
9	Used If/Then/Else Statements

24.38. This difference was not statistically significant; the T-test indicated a p-value of 0.183.

The higher average score for the Fall group in comparison with the Spring group for both pre- and post- survey indicated that the Fall group was already equipped with higher spatial skills before they joined the program. To test this, an independent samples T-test was run to examine if there were significant differences between the two groups. Table V shows the Fall group scored 7.513 in pre and 6.121 in post higher than the Spring group with a t-value of 2.82 and 2.61 respectively with p–value significantly less than 0.05. This test demonstrates that the Spring and Fall groups had differing pre- and post- spatial skills as measured by the PSVT:R. We further explore the differences between the two groups in the next section.

C. Assessment of Students' Programming Skill

To further analyze the differences between the two groups, we developed a rubric to evaluate the coding sophistication represented in participants' final projects. The rubric is based on the topics the program covered throughout the program and whether the participants applied that knowledge in their final projects. There are a total of nine criteria as shown in Table VI; each criterion was worth one point, yielding a total of nine points.

The projects were separately scored by the two of the researchers separately to ensure there are all the final projects criteria were met accordingly. We wish to note that this rubric does not reflect the participants' quality or creativity of their projects but rather the degree of programming complexity exhibited.

The independent sample T-test was run again between the Spring and Fall group and Table VII shows that Fall Group scored a higher mean than Spring Group and has tighter standard deviation. The t-value score 1.787 with one-sided significance of less than 0.05 reject the null hypothesis. The

TABLE VII
COMPARISON BETWEEN FALL AND SPRING GROUP'S EVALUATION SCORE

	Mean Score	Standard Deviation	t-value	Significance (1-tailed)
Spring (N=9)	4.89	2.93	1.787	0.045
Fall (N=12)	6.58	1.31		

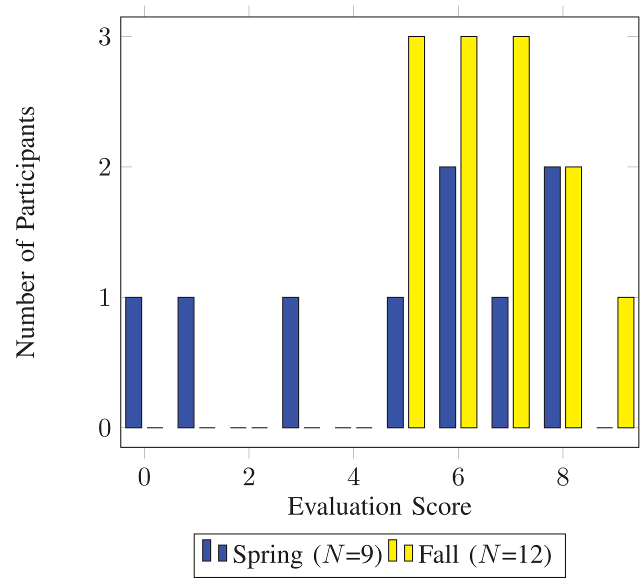

Fig. 13. Distribution of Project Evaluation Score between Spring and Fall group

significance indicates the Fall group has more coding experience before joining the program; the prior result (statistically significantly different baseline PSVT:R scores between the two groups) further indicates that the Fall Group had better SS prior to the program. (Owing to one participant not submitting a final project, the Fall Group had an N of 12, vs. 13 in the PSVT:R analysis.)

Figure 13, a histogram distribution of the final project evaluation scores, shows this result visually. All 12 projects from Fall group scored at least five points, with a greater mean than the Spring group.

VII. DISCUSSION

For PSVT:R, the close statistical significance of the Spring Group led us to believe that with more sample size, the result has a possibility to reach the significance. The Fall Group's much higher pre/post mean scores, with no change from pre to post, compared to Spring group suggests that MYR may provide more effective SS training to students with lower spatial skills and more nascent coding skills. This was further illustrated with the project evaluation analysis between two groups, showing that the Fall group had higher baseline programming skills. The total sample size of 22 does leave us to be cautious to generalize our conclusions yet, and warrants further investigations in the future to strengthen our findings.

TABLE VIII
QUESTIONS ASKED IN EVALUATION SURVEY AND MEAN OF RESPONSE

	Questions	Mean Scores[a]
Q1	Overall, how satisfied are you with the program?	4.00
Q2	Zoom mini-lectures	3.73
Q3	Zoom breakout rooms	3.09
Q4	General use of Discord	3.82
Q5	Pods in Discord	2.45
Q6	Supplemental activity sheets	2.81
Q7	Watching previously recorded sessions	3.09
Q8	Sharing my work with the group	2.91

[a] The maximum mean score is 5

VIII. PROGRAM EVALUATION

With intentions to potentially recreate this study, we found it important to get feedback on how the participants felt the program went. We distributed a post-survey evaluation of our program to only Spring group which consisted of eight Likert-scale items and several additional open-response items. A total of ten participants completed the post-survey evaluation. The Likert-scale prompts and mean of students' responses are shown in Table VIII.

Items Q2 and Q3 were directed towards the method of holding the sessions over the Zoom platform and using its breakout room feature. For the overall effectiveness of Zoom mini-lectures, the average of the responses was 3.8 and the overall effectiveness of Zoom breakout rooms was 3.2. Item Q8 asked for students' satisfaction sharing their work in those Zoom online environments, and the score of 2.91 suggests that most of the students were comfortable with presenting their work in the virtual environment.

Items Q4 and Q5 examined our use of Discord as a platform to communicate outside of our regularly scheduled sessions (Q4, Q5). Within our Discord server, we had pods that were set up to break up our group of participants into smaller sections based on skill level. The purpose of these pods was to provide better help in a more controlled environment where students could feel comfortable reaching out for help. For the effectiveness of the Discord server, the average score was 3.9. The effectiveness of the Discord pods had an average of 2.6. This lower score matches our observations, which suggest we did not receive much usage of the pods.

Items Q6 and Q7 target students' engagement with MYR outside of the live meeting times. To aid the students, the supplemental activities related to topics covered in each weeks' lesson were provided as well as recorded sessions for students who missed it live. Overall, the satisfaction scores were 2.81 and 3.09 respectively, a moderate satisfaction suggesting that students had some use outside of the session.

Item Q1 addressed overall enjoyment of the program. The average overall satisfaction of the study was 4.0, the highest score of any individual item. All but one of the participants felt that the pace of the program was just right with one participant feeling it was too fast. We tried to keep a constant flow of new material for the participants for people to stay active and continually learn new material. Due to the high volume of material covered, if the participants did not have any prior programming experience, one could become overwhelmed. We planned for this and tried to keep our live lectures short and to the point and allowed for breakout rooms with planned activities in order to reinforce what was just taught.

The last portion of the evaluation survey consisted of open feedback from the participants about the program. The first item was "Is there anything you wish we covered that we didn't?" We received mostly no, but we also saw that one participant wanted to see more animations and another participant wanted to cover more of the fundamentals of Java. Although some of the suggestions were out of the scope of the study, some lessons did cover animation. The next item asked, "Is there anything you found difficult?" We did strive to provide a balance between supporting the most beginning students in our program while also engaging the more advanced ones, which is a challenge in most computer science teaching environments.

IX. CONCLUSIONS AND FUTURE WORK

With this study, our goal was to evaluate a custom programming environment that was developed by our research group, called "My Reality" (MYR), which allows beginner and more advanced programmers to create immersive three-dimensional scenes which can be explored in virtual reality. We administered the PSVT:R test to examine students' spatial skills in a pre/post format. The evidence from our PSVT:R scores suggests that the use of MYR may be beneficial to more beginning students who are both learning computer science concepts and who also have more opportunities to develop their spatial skills.

Based on our students' creative products and their evaluation of the program, we found that students enjoyed working with MYR and were successful in using it for creative expression. We wish to highlight this important distinction between our work and other programs designed to improve students' spatial skills: our intervention shows promise for students to improve their spatial skills while also producing creative works and exercising their coding skills.

In future work, we plan to conduct the study again with students who are earlier in their development of both coding skills and spatial skills. We are encouraged by the improvement in spatial skills scores for our first group, which was demonstrably more early in these skills than our second group. We are also encouraged by both groups' engagement in the study, including the quality of the work they produced and their enjoyment in doing so.

ACKNOWLEDGMENTS

We would like to give thanks to all the students who participated in this study. Their time and effort are much appreciated. We would also like to thank Dr. So Yoon Yoon, who permitted us to use their PSVT:R instrument for this study. Finally, we want to give thanks to Jaelyn Dones from our research group for help running the program and Jason Kiesling for initial planning of the program.

REFERENCES

[1] S. Y. Yoon, *Psychometric properties of the revised purdue spatial visualization tests: visualization of rotations (The Revised PSVT: R)*. Purdue University, 2011.

[2] C. Berns, G. Chin, J. Savitz, J. Kiesling, and F. Martin, "Myr: A web-based platform for teaching coding using vr," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 77–83. [Online]. Available: https://doi.org/10.1145/3287324.3287482

[3] S. A. Papert, *Mindstorms: Children, computers, and powerful ideas*. Basic books, 1980.

[4] J. Wai, D. Lubinski, and C. P. Benbow, "Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance." *Journal of educational Psychology*, vol. 101, no. 4, p. 817, 2009.

[5] S. A. Sorby, "Educational research in developing 3-d spatial skills for engineering students," *International Journal of Science Education*, vol. 31, no. 3, pp. 459–480, 2009. [Online]. Available: https://doi.org/10.1080/09500690802595839

[6] J. B. Carroll *et al.*, *Human cognitive abilities: A survey of factor-analytic studies*. Cambridge University Press, 1993, no. 1.

[7] J. Parkinson and Q. Cutts, "Investigating the relationship between spatial skills and computer science," in *Proceedings of the 2018 ACM Conference on International Computing Education Research*, ser. ICER '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 106–114. [Online]. Available: https://doi.org/10.1145/3230977.3230990

[8] S. A. Sorby and B. J. Baartmans, "The development and assessment of a course for enhancing the 3-d spatial visualization skills of first year engineering students," *Journal of Engineering Education*, vol. 89, no. 3, pp. 301–307, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2168-9830.2000.tb00529.x

[9] S. A. Sorby, "Developing 3d spatial skills for engineering students," *Australasian Journal of Engineering Education*, vol. 13, no. 1, pp. 1–11, 2007. [Online]. Available: https://doi.org/10.1080/22054952.2007.11463998

[10] S. Cooper, K. Wang, M. Israni, and S. Sorby, "Spatial skills training in introductory computing," in *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ser. ICER '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 13–20. [Online]. Available: https://doi.org/10.1145/2787622.2787728

[11] R. Bockmon, S. Cooper, J. Zhang, M. Dorodchi, and S. Sorby, "A hybrid approach to administering a spatial skills intervention," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–9.

[12] H. C. Wauck, E. D. Mekler, and W.-T. Fu, "A player-centric approach to designing spatial skill training games," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–13. [Online]. Available: https://doi.org/10.1145/3290605.3300296

[13] L. Stewart and C. Lopez, "Developing spatial visualization skills with virtual reality and hand tracking," in *HCI International 2021 - Late Breaking Posters*, C. Stephanidis, M. Antona, and S. Ntoa, Eds. Cham: Springer International Publishing, 2021, pp. 390–398.

[14] S. Papavlasopoulou, M. N. Giannakos, and L. Jaccheri, "Exploring children's learning experience in constructionism-based coding activities through design-based research," *Computers in Human Behavior*, vol. 99, pp. 415–427, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563219300184

[15] K. Mukhtar, K. Javed, M. Arooj, and A. Sethi, "Advantages, limitations and recommendations for online learning during covid-19 pandemic era," *Pakistan journal of medical sciences*, vol. 36, 2020.