Enhanced Robotics! Improving Building and Programming Learning Experiences for Students with Visual Impairments

Stephanie Ludi
Computer Science & Engineering Dept.
University of North Texas
USA
Stephanie.ludi@unt.edu

Debra Bernstein
TERC
USA
debra_bernstein@terc.edu

Karen Mutch-Jones
TERC
USA
karen_mutch-jones@terc.edu

ABSTRACT

Making technology and computer science learning experiences accessible to students with disabilities is an important step in preparing them to enter the workforce of the future—one in which many jobs will require skills to solve problems with technology. This paper presents the tool and curricular enhancements developed to make the *Exploring Computer Science* Robotics unit accessible to students with visual impairments (VI). It describes the evolution of enhancements, based on formative evaluation studies, to increase support as VI students engaged in building and programming LEGO Mindstorms robots. Results describe the ways in which enhancements were iteratively designed in response to student engagement and confidence, as well as their emerging understanding of top-down and bottom-up processes in robotics design and programming.

ACM Reference Format

Stephanie Ludi, Debra Bernstein, & Karen Mutch-Jones. 2018. Enhanced Robotics! Improving Building and Programming Learning Experiences for Students with Visual Impairments. In *SIGCSE '18: 49th ACM Technical Symposium on Computing Science Education, Feb. 21–24, 2018, Baltimore, MD, USA.* ACM, NY, NY, USA, 6 pages.
https://doi.org/10.1145/3159450.3159501

1 INTRODUCTION

Recent trends in STEM education emphasize technology, recognizing the educational benefits and career opportunities computing offers for all students [11]. Creating access and supporting students with disabilities is essential, particularly given their lower achievement levels in STEM [29] and lack of representation in STEM fields, particularly in computer science. [2, 21]

While students with visual impairments reflect their sighted peers in terms of cognitive ability [14], their participation in STEM can be limited by insufficient resources and instruction, and a lack of teacher preparation in computing and non-visual teaching methods [12, 13]. Ultimately this closes down access to the 'STEM pipeline' of courses and activities in high school, college, and (eventually) the workplace [3]. STEM participation can impact students' attitudes towards STEM fields [19], which are highly correlated with later STEM achievement [17, 28].

Participation in enhanced STEM programming can make a big difference in students' attitudes towards the STEM fields and potential careers [19]. Specifically, programs that make STEM 'active' by using hands-on materials and engaging students in design-based tasks, have been shown to increase students' interest in STEM [17, 25].

Participation in robotics activities, in particular, can: promote students' computational thinking, critical thinking, and problem solving skills [8, 27]; and increase understanding of complex systems [1]. Robotics provides opportunities to practice teamwork and collaboration [6, 22], which can influence student self-efficacy [9, 23]. Enabling teamwork is especially important for students with VI, who tend to be more socially isolated than their sighted peers [30].

Moreover, robotics is a well-used vehicle for teaching computer science concepts and skills [5, 7], and engaging students in engineering design tasks [6].

LEGO Mindstorms robots, which are used in the Robotics Unit of the Exploring Computer Science (ECS) curriculum [10] have been used to teach computer science concepts such as variables, control flow, constants, procedures and problem decomposition, expressions, and arrays [5]. Having students create their own robots encourages them to use their knowledge and understanding in creative ways [24], while engaging them in a design process that itself promotes critical thinking and problem solving [4]. Thus, designing robots introduces and allows students to iteratively practice computer science and engineering skills.

This paper presents our work in creating an accessible ECS Robotics Unit for 3 cohorts of students with visual impairments, which included designing hardware and software accommodations. Both aspects evolved over the course of the project, and this paper will show the evolution to demonstrate the challenges and rewards of the effort, both technically and in terms of student engagement.

SIGCSE '18, February 21–24, 2018, Baltimore , MD, USA

ACM ISBN 978-1-4503-5103-4/18/02...$15.00
https://doi.org/10.1145/3159450.3159501

2 EXPLORING COMPUTER SCIENCE: ROBOTICS UNIT

ECS is a high school curriculum comprised of 6 units: Robotics, the focus of this paper, as well as Human-Computer Interaction, Problem Solving, Web Development, Programming, and Data Analysis [10]. Several tools, activities, and resources used in the ECS curriculum are not accessible to students with visual impairments. As a result IECS, where the 'I' stands for Inclusive, was designed [20]. The Robotics Unit is expected to be taught in about 7 weeks. The instructional breakdown of the Robotics Unit can be found within the ECS Teacher Guide [10]. This paper will focus on building (Hardware) and programming (Software) accommodations.

3 HARDWARE & SOFTWARE ACCOMODATIONS

The ECS Robotics Unit requires students to build and program a robot, however there are many barriers that impede VI students' ability to do such work. The LEGO Mindstorms building instructions and the block-based programming software that come with the kits are not accessible to students with visual impairments due to the graphical nature of each. In order to make the ECS Robotics Unit accessible both the hardware and software aspects required significant accommodations.

3.1 Hardware

LEGO building instructions, as shown in Figure 1, are provided but are not accessible to students with visual impairments. One challenge involved identifying and differentiating LEGO Mindstorms pieces, so we created a prototype tray where parts were organized and easily accessible.

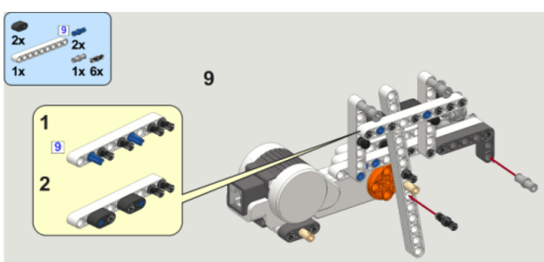

Figure 1. Sample of visual LEGO building instructions [26]

Each section had a flap with parts glued to it, so students could quickly feel the type of part within a section. Having sample pieces enabled students to visually or tactily browse for pieces without disrupting the contents of the tray. In addition, similar parts were in nearby sections (e.g., inner wheels of various sizes), so that students could evaluate their choices (e.g., a tiny, small, medium, or large wheel) and select the part needed.

Figure 2. A tray prototype with exemplar pieces on flaps

The next challenge was for students to learn how pieces fit together generally, and then how to use them to create robots. Unstructured tactile exploration is inadequate and lacks efficiency. To support the development of building skills, the team conducted an activity with the third cohort that provided a guided exploration that transitioned into explorative small building (sans the LEGO Brick itself). Figure 3 shows group work as a whole alongside a close-up of students doing some exploratory building. In the activity, students learned about the pieces by type: Structural Elements, Connectors, Wheels and Treads, Motors. Our approach required students to form highly collaborative teams that could compare and contrast individual parts for identification; and use a visual and tactile guide to map parts to their proper names, based on their degree of vision. Via self-created mini-builds, they also explored how different types of pieces impact robot design.

Figure 3. a) An overview of the group tactile activity and b) a close-up of students exploring pieces

This exploration scaffolds engineering design decisions that students will make later in the process. Through this activity, students were able to practice how to attach connectors to different supports and to appreciate their different functions. This activity was successful with a small group, but in order to scale up, we would need to restructure the process so that larger numbers of students could move through stations that are organized by type of materials (e.g., all parts related to wheels).

The third challenge was providing accessible building instructions for an initial robot to be used with the programming tutorial. The ECS curriculum includes a set of initial building instructions that are pictorial. As an alternative, one of the authors developed instructions describing the pieces, how to orient them, and how to navigate to where they must be placed on the model. This approach is based on work by a blind high school student [18]. Our text-based version of the instructions for the same robot model used in ECS was tested with the third cohort of students. The Results and Discussion section will talk about the feedback in more detail. While promising, the pilot revealed that students had trouble orienting the model, attaching sub-models to the main robot, and using LEGO terminology. Therefore, revisions to the building instructions and tray design will be conducted based on student feedback.

3.2 Software

JBrick is the software used to program LEGO Mindstorms robots. The language is NXC (Not eXactly C), thus JBrick serves as an accessible, programming tool that both sighted and visually impaired students can use when completing the ECS Robotics Unit. In contrast to many block-based tools that introduce novice programmers to robotics, JBrick requires students to program in a text-based language. Prior work has outlined the evolution of JBrick with students with visual impairments [15, 16]. Accessibility features include:

- Screen reader compatibility (Mac and PC)
- Compatibility with refreshable Braille displays
- Customization of text size, background and text colors
- Audible line numbers
- Audio feedback for compilation and program download
- Keyboard shortcuts for tasks (e.g. compile, save)
- Highlight of the current line for easier navigation
- The ability to jump the cursor to the line with the compiler error

In addition to JBrick itself, a set of tutorials provides students with the means to get started with programming, using JBrick, and working with the LEGO Mindstorms robots.

4 EVALUATION

The evaluation was designed to offer formative findings to the development team, enabling them to revise and strengthen the enhancements to better meet the learning and accessibility needs of VI students.

Data were collected via evaluator observations of students as they designed, built, and programmed robots, as well as through surveys and interviews with students, and interviews with undergraduate mentors. These instruments were designed to: 1) assess the usability of hardware and software enhancements, identifying which students were best supported by the enhancement (e.g., those with particular kinds of prior computer science experience, those who were blind vs. those with low vision); 2) gauge the influence of enhancements on student interest and engagement in computer science; and 3) identify components of the enhanced robotics unit where barriers remained.

Three cohorts of students—17 male and 2 female in total—participated in the CS Academy's robotics unit during the summers of 2013, 2015, and 2016. Participants were recruited from across the US. Eleven students (58% of participants) had limited vision, and the rest were blind. Students were entering grades 8 to 12, and were homeschooled, attending public schools, or schools for the blind. Most had little to no experience with robotics design or programming.

5 RESULTS AND DISCUSSION

Evaluation findings over three years suggest that in order to fully address accessibility needs for robotics learning, both top-down and bottom-up supports are required for VI students. For instance, enhancements can be designed to scaffold students' high-level thinking about design and problem-solving processes and to help students learn specific skills like appropriate use of programming syntax. These supports are provided by enhancing (or, when needed, re-designing) the software, hardware, and curriculum activities.

5.1 Accessibility Support for Designing and Building (Hardware)

Designing and building robots requires understanding both top-down and bottom-up processes. For example, students need to understand both how to assemble individual components of the robot and how those components can come together to create a functional piece of hardware (see table 1). Merging these two types of knowledge was essential for students with VI to successfully complete the robot design and building task.

Table 1. Top-Down and Bottom-Up Design/Build understanding that was supported by enhancements.

	Top-Down	Bottom-Up
Designing/Building	Engineering the robot's form and function – What will it do? What parts are needed, and what will those parts contribute to the robot's overall function? How will different components fit together (e.g., where is the right spot for particular components, like the robot's 3rd motor). Coordinating hardware and software	Familiarity with individual/groups of Lego pieces (form and function) Steps and details of assembling Lego structures (e.g., what pieces go together, or connect other pieces?) Using groups of Lego pieces to make larger components (e.g., assembling pieces into a support arm for a sensor)

Enhancements to support top-down understanding. In the first year of the project, evaluators noted that some students gravitated toward building robots and others toward programming with JBrick. The students were quite aware of their preferences, saying things like "I'm really not a big Lego guy" or "I'm not good at coding." Often the two activities were divorced, most observably when two boys worked together with one building a robot and the other programming, and neither checking in with the other about design until much later in the process. As a result, the physical design of the robot didn't match the movements that were programmed, creating a fair amount of frustration for this pair. Even after extensive troubleshooting which left them without a satisfying conclusion (their robot never really fulfilled the requirements of the Challenge), it was hard to tell whether they understood that it was essential to collaborate on both building and programming.

It became clear that the project needed to provide more support to overtly help students link hardware and software design processes, which necessitated engaging all students in hardware exploration. During the second project year, all students broke into groups to build a complete robot (with assistance from a mentor). The process was collaborative, making the workload more manageable for those with limited building experience. Three blind students in this cohort developed shared verbal descriptions of Lego pieces to facilitate sorting and building (e.g., 'I'm putting all the round pieces in here'), and took turns retrieving pieces and assembling the robot while the mentor read out loud from the construction manual.

Observing the robot building process, mentors noted the importance of students (particularly blind students) being able to feel the entire robot and well as the sections and individual parts, in order to create a 'schematic' of the final product in their heads. But this observed benefit came with a challenge. Keeping the big picture in mind, while at the same time remembering the shape and size of parts, the types of connectors required to create robot sections, and the ways in which these sections fit together to create the full robot, increased memory load beyond some students' capacity. As a result, additional memory supports were developed and offered to the final cohort (as described in Section 3.1).

Students in the final year of the program responded well to these changes, and were able to form a full understanding of robot design, including which components of a design could be flexible. As one student from this cohort noted, "[what] was easy was that you could change things [on the robot]. Like anything could be kind of changed to your liking. The hard part was like knowing what you could change and what you couldn't. So… you can change the body of the robot. You can change how it's formed and stuff, but you can't really change that like some wheels just won't turn, and some structures just won't move too fast or too slow."

Enhancements to support bottom-up understanding. In the early years of the program, many students (especially those who were blind) struggled with identifying Lego pieces, discerning how/where to connect them while building, and remembering the shape of the robot/robot sections they were trying to build to meet the design challenge. Thus, to enable all students to develop basic building skills, three enhancements were designed to guide bottom up learning: the trays and scaffolded exploration as described in Section 3.1, and building cloths that provided a clean surface and a tactile edge (so students could tell where to stop looking for pieces).

Data collected from students suggest that these basic enhancements were helpful. When surveyed, the majority of students (6 out of 10) agreed that it was 'easy to locate parts' and physically assemble the Lego pieces. One student commented, "it was confusing, but got easier over time. Had to get used to it at first, I'm not used to playing with those kinds of Legos."

While one student with low vision said he preferred to search for parts that were in their original bags, all of the other student felt the trays were "way better". Some recalled challenges they faced while working with Legos when younger. For instance, one blind student said, "I was in a robotics class back in elementary school. It was really bad…where everything was just thrown into a huge [bin]. Then we had to go in there and just look and look and look." These students particularly liked having the Lego piece taped to a flap on top of each section so they could slide their hands across the tray's surface and locate the section/pieces rapidly. One student suggested adding a Braille description next to the actual part (e.g., a connector with two holes). Several students thought making the flaps sturdier (a plastic piece or a stronger cardboard) would help them to lift and manipulate pieces with greater ease.

Taking time to explore parts as a group also seemed valuable. Quick activities, like passing around parts by type, engaged student and got them thinking about building robots. For instance, after feeling several key pieces, the instructor asked students to figure out how one piece could be connected with another or how two parts might work together. This led to some limited building in small groups, and the excitement during this time was visible and audible. As they reflected on this portion of the enhancement, all students said that the exploration activities were valuable. In fact, several of them wanted more. These students wanted to feel every part within a category to more fully identify differences, and to spend more time understanding sensors and motors before using them in their robots.

With a set of tested enhancements in place for our Y3 cohort, we found that addressing top-down and bottom-up needs in an integrated fashion led to improved confidence while building robots (see Figure 4).

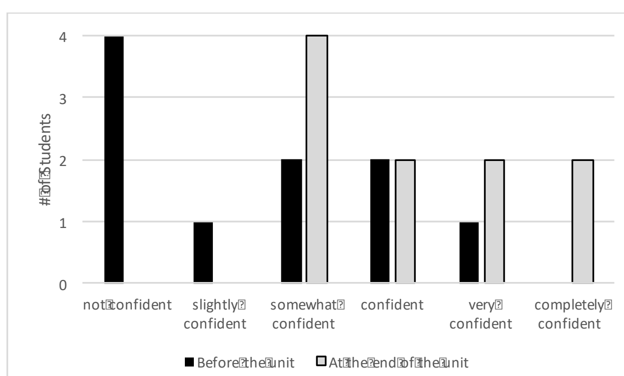

Figure 4. Students' confidence ratings about building robots; Year 3 cohort (n=10).

5.2 Accessibility Support for Programming (Software)

Programming robots requires students to engage in top-down planning (e.g., deciding what the program will accomplish, including sections in the correct order) while understanding how to use the syntax and commands of a particular language (see Table 2). IECS enhancements sought to support students in mastering top-down and bottom-up elements of programming.

Table 2. Top-Down and Bottom-Up Programming understanding that was supported by enhancements.

	Top-Down	Bottom-Up
Programming	What the whole program is designed to accomplish Coordinating hardware and software Flow of the program (i.e., order of components)	Syntax of programming What individual programming commands do and how to use them Steps of programming, debugging Programming sequences/parts of programs that need to go together (e.g., specify which motor port + speed + direction of motor)

Enhancements to support top-down understanding. Many of the IECS participants felt that learning how to program was challenging and required persistence, but in the end felt highly satisfied with their accomplishments. One student from the Y1 cohort noted that programming was, "like [learning] a language; as you practice it gets easier." Another student, from the final cohort, expressed enthusiasm about the way a program could control different components of his robot: "motors are the things that really fascinated me. And that's just going back to coding. It really fascinated me how I could write some code, plug a brick Lego in, and then download it, and then make it so the Lego would spin, rotate, move a certain amount of time, wait a certain amount of time. You know? It would just kind of fascinate me."

The relationship between the written program and the robot's actions was reinforced through the use of JBrick tutorials. The tutorials introduced students to different concepts and commands available in JBrick (e.g., moving forwards, backwards, using sensors, if statements) and provided an opportunity to for students to immediately test out short programs on their robots. For example, one section of the tutorial deals with motors and encourages students to see how different motor speeds and wait times look on their robot. Blind students would often run the programs with one hand on the robot to gauge its speed. These exercises helped prepare students to create their own programs later on by showing them how to break the programming challenge into smaller parts, and drill down further to design and program. When needed the undergraduate mentors stepped in to help students break down a challenge into a set of smaller goals, or to help with programming.

Enhancements to support bottom-up understanding. Unlike Mindstorms, a drag and drop software application that requires vision to manipulate the programming blocks, JBrick requires that students write code to program robots. The evaluators noted that while accommodations for students with disabilities often make the work easier, in this case JBrick makes the work harder, even though it increases *visual* access.

Some students found the syntax-heavy programming language to be a challenge. One student noted, "I found it hard to actually type in the code…it's cap sensitive, and you have to put a semi-colon at the end of every line." Another student noted there were "so many weird punctuation marks; I get a lot of compilation errors." However, other students found that because JBrick offered clear robotics command about each function, they were easy to remember. One student explained: "So if you're in the code, you'd see clear motors and then off motors, for example. So you'd understand it means turn off the motors [or]…if you see task main in the line of code, you know that that's where it's going to start…It's pretty easy to understand the actual code."

To help mitigate syntax challenges, the tutorials provided segments of sample code that students could paste into JBrick and then edit to create their own code. This allowed students to achieve their programming goals without first having to master all of the syntax. It also served as a memory cue for the different commands. Many students did cut and paste program segments as they were beginning to program, but then moved on to writing independent code when they wanted their robots to perform actions not described in the tutorials.

JBrick also offers other supports including auditory prompts and tools to help alleviate memory load (such as line numbering that helped students locate their place in a program). Again, with tested final enhancements in place, we asked Y3 students to rank how useful those features were in helping them program their robot. A great majority of the cohort indicated that having a *screen reader* to read their code and *audio feedback,* which indicated that their program was downloaded, was very or extremely helpful. And 75% of the group rated the clarity of menus as moderately to very helpful in finding commands.

Students with low vision were able to access the line numbering option, but they had mixed reviews about its helpfulness, in part because this sub-group still needed to hear which line they were on. Unfortunately, the NVDA screen reader did not read the line numbers, so this feature wasn't used fully. A couple of students also used a feature that allowed them to change the size and color of their code, but most students relied on the screen reader.

Overall findings from our Y3 cohort suggest that the final version of JBrick and the supportive curriculum structure led to improved student confidence about programming robots.

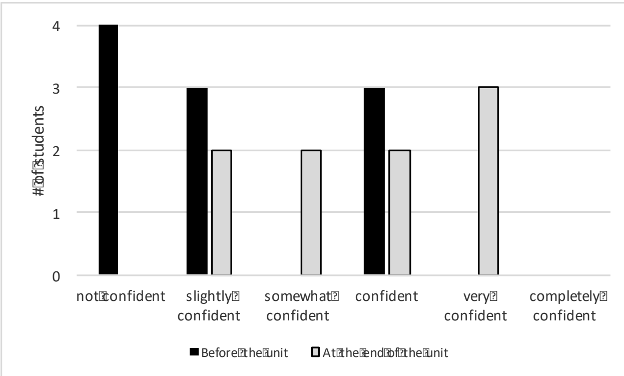

Figure 5. Students' confidence ratings about programming robots; Year 3 cohort (n=10).

5 SUMMARY & FUTURE WORK

Evaluation findings illustrate the ways in which hardware and software accommodations for the ECS Robotics Unit reduce tool and curricular barriers, and increase access for students with VI. Evaluating the tools and activities for such a complex (Robotics) unit, can help teachers provide new resources to maximize access to all students. As a result of our accessibility evaluation, we now have a complete set of tested and revised enhancements, enabling us to measure longitudinal change in student engagement and learning.

In terms of development, our future work will include written and video materials for teachers and students, to illustrate the tactile exploration activities. In addition, next steps for the development team are to include the LEGO Mindstorms EV3 platform, and simplify code reuse to help students use blocks of code more easily. A more ambitious project underway is our accessible block-based tool that also supports LEGO Mindstorms programming, but JBrick will continue to evolve.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation awards CNS-1240856 and CNS-1240809. We also thank our mentors, participants and their parents.

REFERENCES

[1] Barab, S. (2014) Design-Based Research: A methodological toolkit for engineering change, in The Cambridge Handbook of the Learning Sciences, edited by Sawyer, R.K., 151-170. New York: Cambridge University Press.

[2] Burgstahler, S., & Ladner, R.E. (2007). Increasing the participation of people with disabilities in computing fields. Computer. 40(5), 94-97.

[3] Burgstahler, S. & Chang, C. (2009). Promising interventions for promoting STEM fields to students who have disabilities. Review of Disability Studies: An International Journal, 5, 29-47.

[4] Cross, N. (1982). Designerly ways of knowing. Design Studies, 3(4), 221-227.

[5] Fagin, B. S., Merkle, L. D., & Eggers, T. W. (2001, September). Teaching computer science with robotics using Ada/Mindstorms 2.0. In ACM SIGAda Ada Letters (Vol. 21, No. 4, pp. 73-78). ACM.

[6] Gomez, K., Bernstein, D., Zywica, J., & Hamner, E. (2012). Building technical knowledge and engagement in robotics: An examination of two out-of-school programs. In B.Barker, G. Nugent, N. Grandgenett, &V.I. Adamchuk (Eds.), Robotics in K-12 Education: A New Technology for Learning (pp. 222-244).

[7] Grabowski, L. M., & Brazier, P. (2011, October). Robots, recruitment, and retention: Broadening participation through CS0. In 2011 Frontiers in Education Conference (FIE) (pp. F4H-1). IEEE.

[8] Grover, S. & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38-43.

[9] Hutchinson, M.A., Follman, D.K., Sumpter, M., & Bodner, G.M. (2006). Factors influencing the self-efficacy beliefs of first-year engineering students. Journal of Engineering Education, 95(1), 39-47.

[10] Into the Loop (2011). Retrieved at: http://www.exploringcs.org/

[11] Israel, M., Pearson, J.N., Tapia, T., Wherfel, Q.M., & Reese, G. (2015a). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. Computers & Education, 82, 263-279.

[12] Israel, M., Wherfel, Q.M., Pearson, J. Shehab, S. & Tapia, T. (2015b). Empowering K-12 student with disabilities to learn computational thinking and computer programming. Teaching Exceptional Children, 48(1), 45-53

[13] King, H. (2011). Supporting STEM teaching and learning of visually impaired students: An ISE research brief discussing in Rule et al. Impact of adaptive materials on teachers and their students with visual impairments in secondary science and mathematics classes. Retrieved from http://www.relatingresearchtopractice.org/article/250

[14] Kumar, D., Ramasamy, R., & Stefanich, G. (2001). Science for students with visual impairments. Electronic Journal of Science Education, 5. Retrieved from http://ejse.southwestern.edu/article/view/7658/5425 30 April 2013.

[15] Ludi, S., & Reichlmayr, T. (2011). The Use of Robotics to Promote Computing to Pre-College Students with Visual Impairments. ACM Transactions on Computing Education. ACM New York, NY, Volume 11 Issue 3, October 2011. doi>10.1145/2037276.2037284

[16] Ludi, S. (2014) Robotics Programming Tools for Blind Students. Journal on Technology and Persons with Disabilities. CSUN Center on Disabilities, http://scholarworks.csun.edu/handle/10211.3/125008, pp. 81-91.

[17] Maltese, A.V., & Tai, R.H. (2011). Pipeline persistence: Examining the association of educational experiences with earned degrees in STEM among U.S. students. Science Education, 95, 877-907.

[18] McDonald, C. (2016). A New System Lets Blind Kids Build Lego Sets. Popular Science. Retrieved at: http://www.popsci.com/blind-high-school-student-helps-make-legos-accessible

[19] Moore, J.L. (2006). A qualitative investigation of African American males' career trajectory in engineering: Implications for teachers, school counselors, and parents. Teachers College Record, 108(2), 246-266.

[20] Mutch-Jones, K., Bernstein, D., Ludi, S. (2016). Creating access to computer science: Enhancing engagement and learning for students with visual impairments. Visual Impairment and Deafblind Education Quarterly, 61(4), 38-51.

[21] National Science Foundation (2015), Division of Science Resources Statistics. Women, minorities, and persons with disabilities in science and engineering. Arlington, VA: NSF 15-311.

[22] Nourbakhsh, I., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., et al., (2005). The Robot Autonomy Mobile Robotics Course: Robot Design, Curriculum Design and Educational Assessment. Autonomous Robotics Journal 18(1).

[23] Nugent, G., Barker, B.S., Grandgenett, N., & Adamchuk, V.I. (2010). Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes. Teacher Education Faculty Publications. Paper 33. Available at http://digitalcommons.unomaha.edu/tedfacpub/33

[24] Resnick, M., Berg, R., & Eisenberg, M. (2000). Beyond black boxes: Bringing transparently and aesthetics back to scientific investigation. Journal of the Learning Sciences, 9(1), 7-30.

[25] Reynolds, B., Mehalik, M.M., Lovell, M.R., & Schunn, C.D. (2009). Increased student awareness of and interest in engineering as a career option through design-based learning. International Journal of Engineering Education, 25(4), 788-798.

[26] Robot Building Instructions. Retrieved from: http://robotsquare.com/2012/02/27/tutorial-manty/

[27] Sullivan, F., (2008). Robotics and science literacy: Thinking skills, science process skills and systems understanding. Journal of Research in Science Teaching, 45(3), 373-394.

[28] Tai, R.H., Liu, C.Q., Maltese, A.V., & Fan, X. (2006). Planning early for careers in science. Science, 312, 1143-1144.

[29] US Department of Education. (2011). The Nation's Report Card: Science 2011. Retrieved from http://nationsreportcard.gov/science 2011/g8_nat.aspx?subtab_id=Tab _6&tab_id=tab3#chart

[30] Wolffe, K., & Sacks, S. Z. (1997). "The Lifestyles of Blind, Low Vision, and Sighted Youths: A Quantitative Comparison", Journal of Visual Impairment and Blindness, May-June, 91, (3), pp. 245-257