

# Creating a Relevant, Application-Based Curriculum for High Performance Computing in High School

Vincent C. Betro, Ph.D.  
Baylor School  
171 Baylor School Road  
Chattanooga, TN 37405  
+1-423-313-7884  
vbetro@baylorschool.org

Mary E. Loveless, Ph.D.  
Baylor School  
171 Baylor School Road  
Chattanooga, TN 37405  
+1-615-260-2530  
mloveless@baylorschool.org

## ABSTRACT

While strides have been made to improve science and math readiness at a college-preparatory level, some key fundamentals have been left unaddressed that can cause students to turn away from the STEM disciplines before they find their niche [10], [11], [12], [13]. Introducing collegiate level research and project-based, group-centered learning at a high school level has a multi-faceted effect; in addition to elevated learning outcomes in science and math, students exhibit improved critical thinking and communication skills, leading to improved preparedness for subsequent academic endeavors [1]. The work presented here outlines the development of a STEM ecosystem where both the science department and math department have implemented an interdisciplinary approach to introduce a spectrum of laboratory and computing research skills. This takes the form of both "in situ," micro-curricular elements and stand-alone research and computer science classes which integrate the language-independent concepts of abstraction and object-oriented programming, distributed and high-performance computing, and high and low-level language control applications. This pipeline has been an effective tool that has allowed several driven and interested students to participate in collegiate-level and joint-collegiate projects involving virtual reality, robotics and systems controls, and modeling. The willingness of the departments to cross-pollinate, hire faculty well-versed in research, and support students and faculty with the proper resources are critical factors in readying the next generation of computing leaders.

## Keywords

STEM; Education; Virtual Reality; Project-Based Learning; High School Research

## 1. INTRODUCTION

Even a cursory exploration of current educational research literature indicates that students' understanding of science, technology, engineering, and mathematics (STEM) topics is increasingly important to the well-being of our global economy

[14], [15]. As noted in a U.S. Department of Labor study, "Long-term strategies to maintain and increase living standards and promote opportunity will require coordinated efforts among public, private, and not-for-profit entities to promote innovation and to prepare an adequate supply of qualified workers for employment in STEM fields. American pre-eminence in STEM will not be secured or extended without concerted effort and investment" [6].

The demand for scientists and engineers is expected to continue to increase at a significant rate, especially in computing-related fields. However, data from international studies, such as TIMSS [16], and national studies, such as the 2007 and 2009 National Assessment of Educational Progress Report [17], [18] indicate that both mathematics and science continue to be academic stumbling blocks for many students, that students are consistently not performing well in courses in these disciplines, and that, as a consequence, too few are pursuing degrees in technical fields.

Additionally, according to Ronald Barr of the American Society for Engineering Education, "When a national sample of adults was asked what kind of career they would recommend to young women, medicine was the top choice. A scant 3 percent suggested engineering" [13]. This perception of the stature and importance of qualified engineers and computer scientists must be changed, and the way we inform under-represented groups about career preparation must include heavy doses of STEM content. Also, as posited on a study from the University of Chicago, as early as elementary school, "teachers who are anxious about their own math abilities are translating some of that to their kids," and this has been found to be particularly true in reference to both female teachers and students, likely based on outdated social norms and lack of role models [19].

The low enrollment and engagement in STEM fields has become serious enough that the State of Tennessee has incorporated engineering standards into the integrated science and math curriculum. The state's goal is to expose students to the engineering design process and computational thinking strategies early with hopes of increasing the number of students that will become interested in pursuing a degree and career in a STEM field.

However, "developing a curriculum does not guarantee that engineering education in K-12 will be successful. A critical factor is whether teachers—from elementary generalists to middle school and high school specialists—understand basic engineering concepts and are comfortable engaging in, and teaching, engineering design. For this, teachers must either have appropriate background in mathematics, science, and technology, or they must collaborate with teachers who have this background" [5]. This statement from the National Academy of Engineering's study Engineering in K-12 Education is precisely why having both interdisciplinary collaboration and expert teachers is so essential to the success of a school's STEM ecosystem.

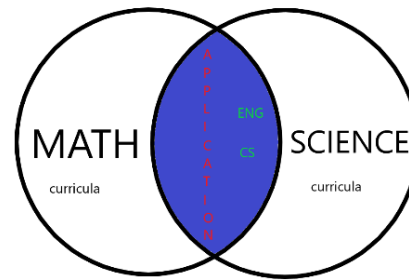
Even more specifically, high performance computing has become an indispensable technology in myriad fields of research. It has been noted by the National Science Foundation, as well as the U.S. House of Representatives, that computational simulation engineering (and thus engendering an understanding and interest amongst young people) is crucial to enhancing our national security and global competitiveness. The U.S. House of Representatives passed a resolution “recognizing modeling and simulation as a National Critical Technology important to the security and prosperity of the U.S.” [7]. Moreover, in 2006, an NSF Blue Ribbon Panel noted that “computer simulation has become indispensable to the development of all other technologies” and “promises to revolutionize the way engineering and science are conducted in the twenty-first century” [8].

High performance computing (HPC) is an excellent vehicle to steer students back into the STEM pathway. Between the applications to such a broad spectrum of real-world problems and the creativity that is implicit in computer science, it draws students to want to better their math and science backgrounds and gives them a creative outlet at the same time. Moreover, by giving students access to faculty, resources, and time to explore computing and engineering, we are giving them tools to solve problems which have not yet been solved. This is quite different from the content and approach espoused in most classes in K-12 education, wherein students are simply trying to figure out what the question is asking so they can also solve a solved, often artificial, problem. It is this aspect of research, the unknown, that allows for both creativity and the understanding of how to approach a truly unknown problem and interpret results with sophistication beyond making numbers simply match the back of the textbook. This is what most excites students because it feels, and is, authentic; it is also the skill set that students most need to take to the university to succeed.

## 2. BUILDING THE ECOSYSTEM

Baylor School has developed a multi-grade, multi-entry point ecosystem to foster interest in STEM-related fields, particularly computer science-related topics and applications; the relevant HPC academic elements are represented in Figure 1. In order to accommodate as many students, as well as levels of interest, as possible, both curricular and extracurricular components have been established, bridging computer science, mathematics and science departments. Within the curriculum, two semester courses, Engineering Design and Independent Research, were introduced. Engineering Design teaches engineering-centric skills (programming, electromechanical systems, and modeling, for examples) in a project-based learning environment that utilizes a flipped classroom design. Both Introduction to Computer Science (Intro CS) and AP Computer Science Principles (AP CSP) were introduced and curriculum was based on the University of California at Berkley curriculum, The Beauty and Joy of Computing. Streamlined coding practices (same language, Snap! and NetsBlox [22, 23], and complementary curriculum pieces) were incorporated into Engineering Design.

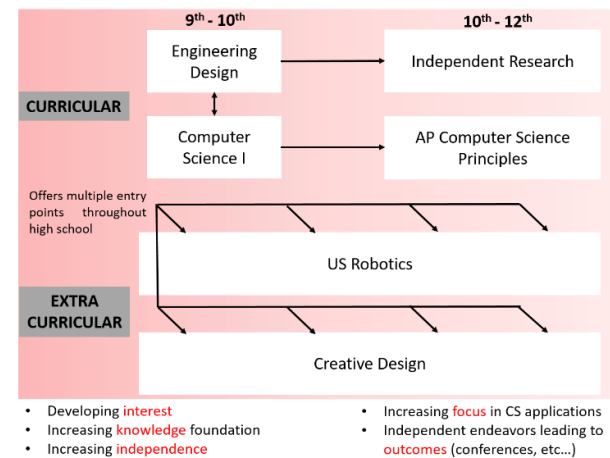
Students who complete Engineering Design (fall semester) are eligible to take Independent Research I (spring semester); students who continue to meet grade and commitment requirements can then continue on to take Advanced Research: Engineering (1 year) and subsequently Thesis: Engineering (1 year), depending on their entry point. Projects specifically offered to foster opportunity in high performance computing include embedded systems and virtual reality (VR) application development.



**Figure 1. Interaction between math and science departments is critical to fostering the interdisciplinary environment where students can work on real applications and become inspired to use HPC and other STEM tools.**

Extracurricular programming was introduced as well. Competitive afterschool Robotics teams were initiated in both the middle (MS) and upper school (US). The MS team competes in the First Lego League (FLL) while the US team competes in the First Tech Challenge (FTC). Core STEM content includes: design/design-thinking, modeling, mechanics, logic/code, physics, and prototyping/3D printing.

Another extracurricular offering is Creative Design, which allows students to participate in a range of exploratory and challenge-based STEM activities/competitions; each year, the extracurricular activity changes to meet the needs of the students’ area of interest. For example, students have participated in NASA spin-off design competitions, VR projects with collegiate partnerships, as well as motion-tracking for the Baylor Dance Club using an XBOX 360 Kinect.



**Figure 2. Relevant HPC academic components. Both curricular and extracurricular option are available to students. Starting in 9<sup>th</sup>-10<sup>th</sup> grade, students can participate in *Engineering Design* and *Computer Science I*, which can be taken stand-alone or with the possible progression to *Independent Research* and *AP Computer Science Principles* courses. After school opportunities allow for entry at multiple points during the student’s tenure. Progressing further through the pipeline, students deepen knowledge, interest, and independence in computer science-related project, with the eventual outcome of presentations and/or publications at collegiate-level conferences.**

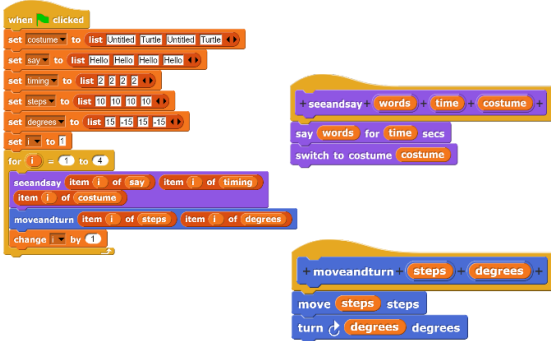
### 3. PROJECT DESCRIPTIONS AND IMPACT

#### 3.1 Within Curriculum

##### 3.1.1 Introduction to Computer Science / AP Computer Science Principles

In order to allow the most students access to computer science in their packed schedules, the decision was made to set up the introductory course and the AP course as semester offerings. This way, a student who already has some programming experience and who is willing to work through the first three units of the BJC course independently can join the AP course in the spring semester if that is the only space they have open. Otherwise, they can dive deeper into programming in the introductory course first and have more tools at their disposal when they take AP CSP. Despite the compressed schedule, our AP scores are consistently above the national and state averages.

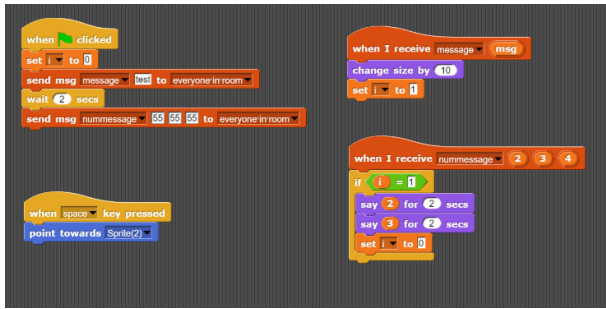
The decision was made to offer *AP CSP* instead of *AP Computer Science A* since the language agnosticism of CSP allows more students to produce projects they are interested in. Also, it leaves room for discussing HPC and applications in more depth, since it is not primarily focused on *Java* syntax.



**Figure 3.** The use of linked lists and standard lists, along with the use of “blocks” or functions, is exhibited here in Snap! Functions may be overloaded, as in C++, and if any Sprites are created as a copy of this Sprite, they will inherit all class functions.

Also, students are introduced to programming using Snap! (University of California, Berkley, CA) in order to avoid the pitfalls of compilation and syntax errors as they are honing their logic and algorithmic skills. Snap! is preferred not only because it is visual but also because unlike Scratch! (Massachusetts Institute of Technology, Cambridge, MA) it is possible to create first class objects and utilize inheritance, making function creation extremely powerful and simple.

The natural direction to take students once they have mastered the basics of programming a serial application is to look at scalability and synchronization issues by having them create a distributed application using *NetsBlox* (Vanderbilt University, Nashville, TN) [23-24]. Students initially struggle with communication because they have never had to think about that level of granularity when it comes to things like order of execution and latency. Once they grasp the concept, they are much better computational thinkers, which is the overall goal of the curriculum. After all, given the numerous programming language a student may encounter over his/her career, being able to design based on the strengths of modern computing architectures and the principles of object oriented development will serve them more appropriately.



**Figure 4.** Each “role” in a *Netsblox* project may have its own code or it may simply mirror each other “role”. Messages are passed within the “room” or to specific “roles”, and instead of addressing to a specific “role”, each role is set to only receive its type of envelope, with the contents passed in as the actual message that is sent.

Once students have been exposed to these fundamental concepts, students can begin to explore and work on projects that interest them. For example, students have used XSEDE [24] resources to compute pi to millions of digits with OpenMP and C++; they have worked to develop MPI code that runs on our 8-node Raspberry Pi cluster and beyond; and they have even developed systems for various applications on the Arduino Uno. Currently, a student is working with the Microsoft Kinect 360 and Visual Studio to create an interactive backdrop for *Verve*, Baylor’s dance troupe.

Finally, it should be noted that this approach has allowed us to grow the numbers of students in our CS program every year, especially once we introduced Intro to CS in our second year, as can be seen in Table 1.

**Table 1.** Enrollment in Intro and AP CSP over three years.

Year	Course	Enrollment
2017-18	Intro CS	--
2017-18	AP CSP	9
2018-19	Intro CS	15
2018-19	AP CSP	20
2019-20	Intro CS	18
2019-20*	AP CSP	22
*projected		

##### 3.1.2 Engineering Design Coding Challenge and Introduction to Virtual Reality

Engineering Design offers project-based curriculum components to convey engineering-centric skills. One of the most intrinsically important skills in engineering is programming. As such, a common programming language and technique were deliberately discussed and agreed upon across all three courses: *Introduction to Computer Science*, *AP Computer Science Principles* and *Engineering Design*. As mentioned previously, *Snap!* was chosen as a versatile, visual programming language that can run on any web browser (many of our students use iPads) such that students could focus understanding fundamental computational principles rather than the potential distraction of syntactic nuances of various languages such as *Java* or *C#*. To elevate curriculum elements and introduce an important yet complicated topic –

networking – students are introduced to *NetsBlox* to introduce concepts related to distributed programming.

Because *Intro CS* is not necessarily a pre-requisite for *Engineering Design*, students are offered an introduction to *Snap!* and *NetsBlox* via a video tutorial series as homework and a series of challenges increasing in complexity to be completed in class (flipped classroom paradigm). After an appropriate introduction to the environments, a challenge is presented to the students involving the gamification of a science/math concept that the students have encountered in the Baylor School curriculum. Using the Engineering Design Cycle as a guide for development, student partners work through studying previous approaches (e.g., games readily available that are educational in context), identifying design requirements, and deciding on an appropriate game design to carry on to production (digitize using *NetsBlox*). The students are then given approximately two weeks to work in a collaborative *NetsBlox* environment (meaning both students can edit the same game space); this project culminates with a paper and presentation, including testing/feedback sessions with classmates. Games that were developed through this curriculum piece included complex control structures, definition of variables, and network elements such as message passing across a network and, in some cases, remote procedure calls (RPCs). Critical thinking / computational thinking skills, creativity, and presentation skills (text and oral) were some intangible skills exercised by the students.

In order to remain relevant, the *Engineering Design* curriculum changes depending on time-sensitive opportunities. For example, one team (four students, seniors and juniors) had the opportunity to work with a VR start-up company, founded at the Massachusetts Institute of Technology. The company's focus was VR experience for the elderly, particularly investigating the use of the immersive technology to combat neurodegenerative disease. The students were asked to design and created a VR environment using *Unity 3D* development software (Unity Technologies, San Francisco, CA) and *C#* scripting for this company, which they subsequently presented at MIT during the following semester.

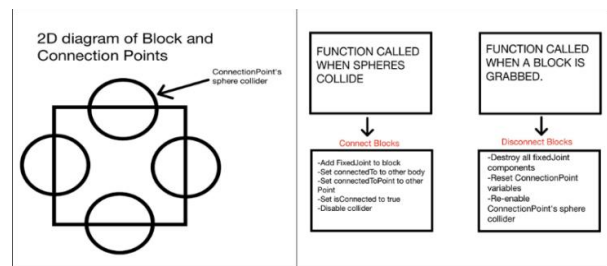
### 3.1.3 Independent Research: High Performance Computing Applications

Motivated students that finish *Engineering Design* with proficiency are eligible to take *Independent Research I*. This course allows students to delve deeper into a relevant engineering-centric research project, such as modeling and virtual reality application development. One student, continuing from the previous MIT collaboration in Engineering Design, took the application of VR in a very insightful direction. The student designed and developed a VR calm room application for children with autism. One challenge from which children on the autism suffer is sensory overload episodes; many places, such as amusement parks and stores have begun to offer physical spaces with dim lights, soothing music, and lowered sensory environments to offer families more flexibility on outings. While this is a step in the right direction, this Baylor student took it a step further, removing the limitations of needing a physical space by offering a virtual solution. The student designed and created a virtual calm room, mimicking those used in private therapy offices as well as those found in progressive places of business. The goal was to create something that could be carried with the family that offered a solution anytime, anywhere. The phone application (paired with something as simple as Google Cardboard / VR Cardboard viewers (\$5-\$15)) offers a flexible, inexpensive solution for families with autistic children. Professionals in occupational therapy have lauded this development. The student was also accepted to present the device at The American Occupational Therapy Association National

Conference in Salt Lake City, UTC in 2018 [26]. While the application development was a step toward using advanced computing techniques for rendering and scaling, the goal of designing a VR experience on a mobile device also offered conversations and successful (and plenty of failed!) attempts at optimization; rendering optimization and reduced scene projection became a very advanced curriculum piece that was subsequently incorporated in off-shoot VR applications discussed below.

Following up on this work, another student expanded the application of the VR calm room to offer passive and functional VR de-escalation experiences for children with emotional troubles, such as those who may suffer from early childhood trauma. The VR application offers a way for teachers, professionals, and families to offer a safe virtual space for the child to be removed from triggering situations (passive VR calming space) as well as to learn techniques and exercises within the VR experience to help manage emotional outbursts, particularly anger and frustration (functional). This work was recently presented as a poster entitled “A Virtual Reality Approach to Pediatric Conflict De-escalation and Anger Management” at Practice and Experience in Advanced Research Computing (PEARC 19) in Chicago, IL in 2019 [21]. Currently, collaborations with the University of Tennessee at Chattanooga Occupational Therapy program are underway to test the efficacy of such experience in a real patient population.

Another relevant application of VR has been the use of immersive technology in education. One student in *Independent Research* developed the backbone for a virtual “hands-on” learning tool that works toward enabling a virtual dissection (and subsequent rebuilding) of a variety of models, ranging from components of human cell to the devices in a computer or car engine. In many classrooms, time, space, and personnel can limit the amount of hands-on learning as well as what type of hands-on learning can occur within a classroom. As educators are aware, hands-on learning has shown to be critical in the comprehension process [20]. The student developed multiple class structures, scripted in *C#* and implemented in *Unity* that established generic component and connections between game object (Figures 5 and 6); further, the student functionalized control of these objects using the *Unity Touch* system.



**Figure 5. Example of student work presented at PEARC. The student is defining generic gameObject handling for hands-on learning in a VR environment. On the left a 2D depiction of block and connection points is represented while the right box is the description on connect and disconnect functions [22].**

With further work, game objects could be interchanged for relevant models and connection definitions (i.e., this piece connects in this way to this other piece) to create virtual hands-on experiences that could be created with by educators with very little training necessary. This student's work was also presented as a poster at PEARC 19; the work was entitled “Virtual Reality Based Environment for Immersive, Hands-On Learning” [22].



**Figure 6. Example of the student work presented at PEARC. The Point class definition for VR hands-on learning and the Connect function definition are above. Code is written in C# [22].**

Four years ago, in collaboration with a digital design teacher in the art department, a STEM after school offering was begun called *Creative Design*. The multi-disciplinary atmosphere brought aboard many students who had been looking for both a creative and technical outlet by which to join the new STEM ecosystem, not simply the latter.

Another example of success from the *Creative Design* program would be the four students who participated in the NASA OPSPARC design challenge in 2017. Their design for a microshutter array fiber optic switch won the competition for best high school design in the country. Students were flown to NASA Goddard in Greenbelt, MD by the organizers in order to present their technology transfer idea to NASA engineers. These students became even more motivated to continue in computing, took more elective STEM courses than they had planned, and one even went on to create a poster for the PEARC 19 on yet another computer science topic: virtual reality [21]. Not only has he explored this space, but he is developing easy to understand tools for teachers to use to create their own VR units. Also, all four students who were on the winning OPSPARC team have gone to University of Alabama at Huntsville to major in aerospace and/or computer engineering.

## 4. LESSONS LEARNED AND RECOMMENDATIONS

Early in the development of these academic elements, it was discovered that allowing students to pursue “anything related to STEM they wanted” was ineffective, as students often do not understand scope control as they have been fooled by the seamless nature of modern technology to think that their idea will require relatively little effort to implement. While the goal was for students to work harder or be more motivated if they controlled the selection of the project or topic, this open-ended project methodology did not manifest as intended. In some cases the students were too overwhelmed to even start a project; in other cases, they projects were so many and diverse that it was logistically unfeasible to mentor effectively. This resulted in students stagnating in projects, getting frustrated, and allowing distraction to take away potentially productive time if the mentor was working with another group. Thus, it is advantageous to both students and mentors to define constraints regarding applications. An example of this would be as follows:

**RECOMMENDED** Problem Definition: Using Unity, create VR curriculum complement pieces for a science class of your choice at Baylor School. Working with a science teacher, select a science topic in which your immersive experience will aid comprehension for the students in the class.

The vagueness of the first problem definition might paralyze students from even knowing where to start. They may try to do things outside of the scope of a beginning project and get frustrated by a steep learning curve. Additionally, every student will have a completely distinct experience in mind, stretching the mentor's time and breadth of knowledge. The specificity of the RECOMMENDED problem statement allows the students

autonomy (selecting the science course and/or topic) while having boundaries that allow them to focus on distinct aspects of development that the teacher has deemed approachable.

In classes such as *Intro CS* or *Engineering Design*, partner work can be very helpful. Collaboration and communication techniques are fostered in this model and confidence in the topic is developed. Partners should be changed regularly to allow students the opportunity to take on a different role. As motivated students delve deeper (*AP CSP* and *Independent Research*), they should progress toward independent contributions. With the growth of a program of this nature, managing individual projects can be time-consuming but, given proper foundational elements and class opportunities, motivated students can be trusted to take a more active role in problem solving at a more advanced level.

Offering both curricular and extracurricular opportunities not only allow for multiple entry points for busy students, it also allows for a range of risk/reward activities. As illustrated above, commitment levels and performance levels vary throughout the offerings, leading to student selection of projects based on their learning pace and abilities in areas of strength. As with any educational program, there needs to be room to fail and develop resiliency. Offering projects for the school, collaborations with external partners, and opportunities to contribute to the large scientific community helps a student develop confidence in lower risk options while highlighting the power of perseverance with high-reward opportunities.

## 5. FUTURE DIRECTIONS

The *AP Computer Science Principles* assessment will change after the 2019-2020 school year, requiring that more foundational knowledge be tested on the written exam. This will change the structure of the course somewhat, but it will make for a deeper dive into the hardware, as the “computing innovation” research project will be scrapped allowing for more time to be spent on how the computer works. This is an area of HPC that hasn’t been explored outside of specific projects where the students worked hands on with an at times temperamental Raspberry Pi cluster.

Additionally, as we add more students to the pipeline, the projects will become more varied and there will be even more carry over allowing us to expand project scopes year after year, given that we set up a good “mentoring” program where students transfer their knowledge of the project and foundational issues so that each project does not die unfinished when a student graduates.

And finally, as Baylor School has a middle school (6<sup>th</sup>-8<sup>th</sup> grades), creating components (in addition to the already offered *Robotics* extracurricular program), that help streamline computational thinking and programming practices/applications will allow students to achieve a high knowledge base and confidence level more rapidly. This extension would continue to allow an expanded project scope as well as the discussion of more high-caliber electives in the upper school.

## 6. ACKNOWLEDGMENTS

We would like to acknowledge the support of Baylor School and the Baylor Board of Trustees for allowing us the resources and the latitude to form the STEM ecosystem we now have. We would also like to thank the many faculty to have participated in these STEM-related endeavors, specifically Heath Montgomery. We thank the Weeks family, through the Harrison and Katherine Weeks endowment, which has been critical to the growth of programs presented here. Also, this work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

## 7. REFERENCES

- [1] Angela Eeds, Chris Vanags, Jonathan Creamer, Mary Loveless, Amanda Dixon, Harvey Sperling, Glenn McCombs, Doug Robinson, and Virginia L. Shepherd (2014). The School for Science and Math at Vanderbilt: An Innovative Research-Based Program for High School Students. *CBE—Life Sciences Education* 2014 13:2, 297-310
- [2] Bevan, Bronwyn & Gutwill, Joshua & Petrich, Mike & Wilkinson, Karen. (2015). Learning Through STEM-Rich Tinkering: Findings From a Jointly Negotiated Research Project Taken Up in Practice. *Science Education*. 99. 10.1002/sce.21151.
- [3] Blikstein, Paulo & Krannich, Dennis. (2013). The makers' movement and FabLabs in education: experiences, technologies, and research. *ACM International Conference Proceeding Series*. 613-616. 10.1145/2485760.2485884.
- [4] National Research Council. 2003. Evaluating and Improving Undergraduate Teaching in Science, Technology, Engineering, and Mathematics. Washington, DC: The National Academies Press. <https://doi.org/10.17226/10024>.
- [5] Katehi, Linda, Greg Pearson, and Michael Feder (2009). *Engineering in K-12 Education: Understanding the Status and Improving the Prospects*. The National Academies Press, Washington, D.C. 2009.
- [6] Jobs for the Future (2007). *The STEM Workforce Challenge: the Role of the Public Workforce System in a National Solution for a Competitive Science, Technology, Engineering, and Mathematics (STEM) Workforce*. U.S. Department of Labor, Employment and Training Administration. April 2007.
- [7] House of Representatives (110th Congress, 2007). Recognizing the contribution of modeling and simulation technology to the security and prosperity of the United States, and recognizing modeling and simulation as a National Critical Technology. House Resolution 487. Passed July 16, 2007.
- [8] National Science Foundation (2006). *Simulation-Based Engineering Science – Revolutionizing Engineering Science through Simulation*. Blue Ribbon Panel on Simulation-Based Engineering Science. 2006.
- [9] Information Technology Association of America (2005). *Innovation and a Competitive U.S. Economy: The Case for Doubling the Number of STEM Graduates*. Washington: ITAA. 2005.
- [10] National Science Foundation (2002). *Characteristics of Scientists and Engineers in the United States: 1999*. Division of Science Resources Statistics. Arlington, VA (SRS 03-407). November 2002. <http://www.nsf.gov/statistics/us-workforce/1999/tables/TableB2.pdf> . February 8, 2010.
- [11] Higher Education Research Institute (2007). *Survey of the American Freshman, special tabulations*. University of California at Los Angeles. Los Angeles, CA, 2007. <http://www.nsf.gov/statistics/wmpd/pdf/tabb-8.pdf> . February 8, 2010.
- [12] American Association of State Colleges and Universities (2005). *Strengthening the Science and Mathematics Pipeline for a Better America*. Policy Matters. Volume 2, Number 11. November/December 2005.

- [13] Barr, Ronald (2005). U.S. Needs More Engineering Students. Miami Herald. August 11, 2005.
- [14] Committee for Prospering in the Global Economy (2007). April 8, 2011.
- [15] National Science Board (2006). Science and Engineering Indicators 2006, Volume 1. Washington, D.C.: National Academies Press. 2006.
- [16] Michigan State University (2001). Third International Mathematics and Science Study. National Center for Education Statistics. April 4, 2001.
- [17] National Center for Education Statistics (2007). National Assessment of Educational Progress Report. 2007.
- [18] National Center for Education Statistics (2009). National Assessment of Educational Progress Report. 2009.
- [19] Kaplan, Karen (2010). Female teachers may pass on math anxiety to girls, study finds. Los Angeles Times. Jan 26.
- [20] Samantha Cleaver. 2018. Hands-On Is Minds-On. Retrieved April 9, 2019 from <http://www.scholastic.com/browse/article.jsp?id=3751901>.
- [21] A. Mook and M. Loveless. Virtual Reality Based Environment for Immersive, Hands-On Learning. Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) (2019). ACM, New York, NY, USA.
- [22] J. Liu and M. Loveless. A Virtual Reality Approach to Pediatric Conflict De-escalation and Anger Management. Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) (2019). ACM, New York, NY, USA.
- [23] B Broll, A Lédeczi, P Volgyesi, J Sallai, M Maroti, A Carrillo. A visual programming environment for learning distributed programming (2017). Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science.
- [24] B Broll, Á Lédeczi, H Zare, DN Do, J Sallai, P Völgyesi, M Maróti, L Brown. A visual programming environment for introducing distributed computing to secondary education. Journal of Parallel and Distributed Computing 118, 189-200.
- [25] Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, Nancy Wilkins-Diehr. XSEDE: Accelerating Scientific Discovery. Computing in Science & Engineering, vol.16, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80.
- [26] Harwood, H and M. Loveless. Virtual Reality-Based Calm Room for Individuals With Autism Spectrum Disorder (2018). Poster presented at The American Occupational Therapy Association National Conference. April 20, 2018 (Salt Lake City, UT).

**Columns on Last Page Should Be Made As Close As Possible to Equal Length**