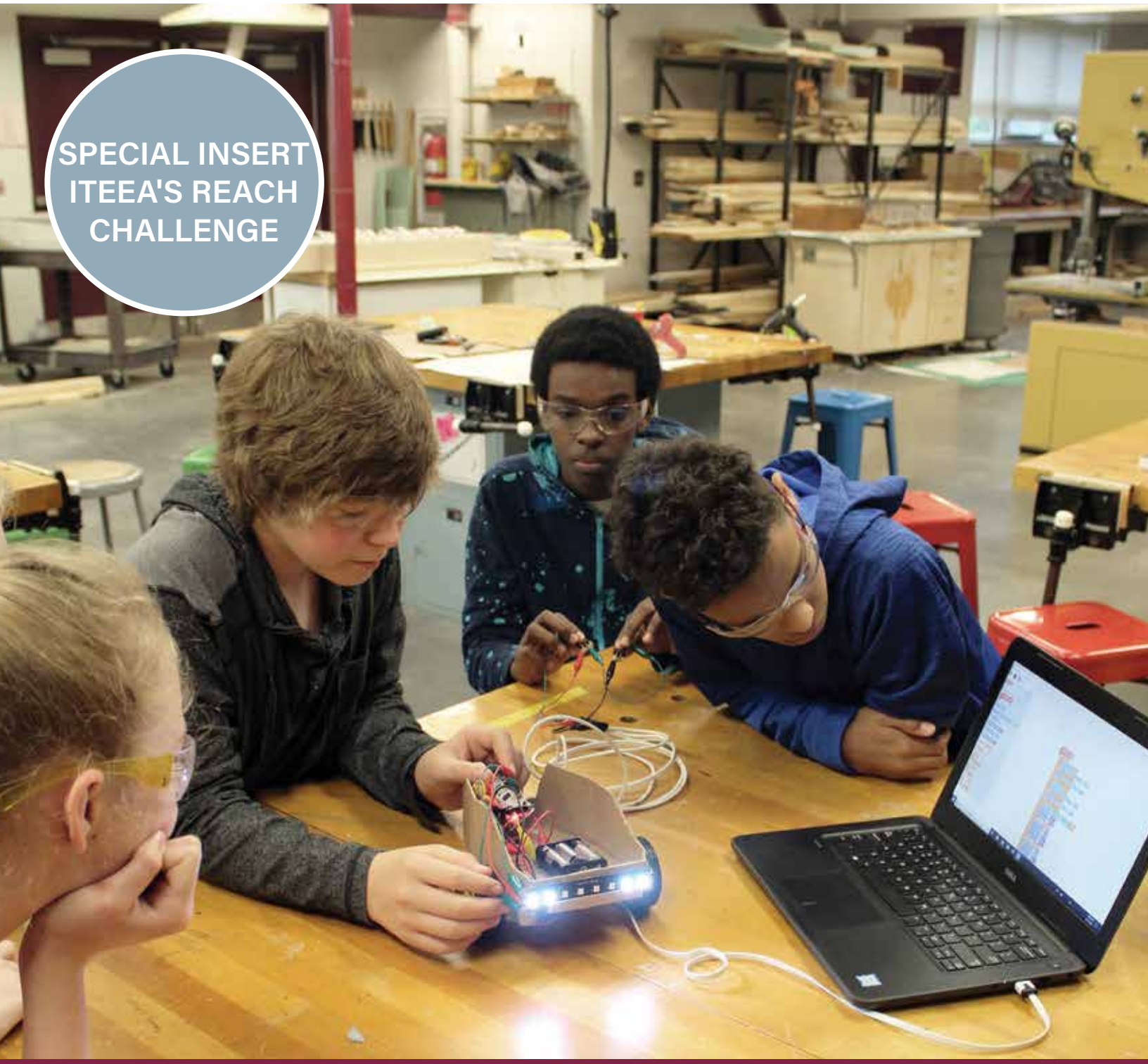
VOLUME 79 ISSUE 2

technology and engineering TEACHER

October 2019

SPECIAL INSERT
ITEEA'S REACH CHALLENGE



INTEGRATING COMPUTER SCIENCE THROUGH ENGINEERING DESIGN

www.iteea.org

feature article

the crumble:

integrating computer science through engineering design

The Crumble can provide a cost-effective way to introduce elementary and middle school students to programming, electronics, engineering design, and other designed-world concepts.

Introduction

Within the past few years there has been an increasing emphasis on making computer science (CS) more accessible for all K-12 students. In 2013 there were 12 states that allowed CS courses to count towards high school graduation, and in 2018 that number had grown to 35 states plus the District of Columbia (Code.org, 2018). Specifically, in the state of Maryland, CS courses can be used to satisfy the technology and engineering (T&E) education graduation requirement despite limited alignment with state regulations pertaining to T&E education course content (Buckler, Koperski, & Loveland, 2017; Love & Strimel, 2017; Love, Tomlinson, & Dunn, 2016). Due to the increasing demand for CS in schools and computer-based programming courses being substituted for hands-on T&E education classes, the authors developed a design challenge to demonstrate how CS concepts could be taught while maintaining a focus on engineering design and technological systems.

CS concepts such as programming can help teach important problem-solving skills and help students develop solutions to other technological problems (Moon, 2018). In 2014 the state of New Jersey revised its T&E education standards to include computational thinking and programming within the context of engineer-

by
Tyler S. Love
and Abraham
Bhatty

ing design, and in 2016 Maryland added computational thinking and CS applications to its technology education standards. Some researchers have suggested that T&E education integrates CS and computational thinking within Standard 17 (Information and Communication Technologies) of *Standards for Technological Literacy* (*STL*) (Buckler, et al., 2018; Love & Strimel, 2017; Wright, Rich, & Leatham, 2012). Love and Strimel (2017) highlighted the differences between *K-12 Computer Science Framework* and *STL*; however, they also noted specific curricular examples that integrated programming and engineering design to develop solutions to designed-world problems. From these examples it is clear that T&E education can provide a unique hands-on, open-ended design context for students to apply computational thinking and programming skills.

The Crumble Microcontroller

One challenge in incorporating computer science in T&E education is finding a viable device (often a microcontroller) that is affordable, user friendly, and has instructional resources. Love, et al. (2016) described the benefits of using the Orange Pi for open-ended design challenges; however, the coding language used to program the Pi can have a steep learning curve for many students. For that reason, we elected to use the Crumble for this design challenge. The Crumble is an inexpensive, easy to use, robust controller (Figure 1) that can be programmed using free drag-and-drop block coding software. It also connects to a wealth of external sensors via alligator clips and has been used to develop solutions to multiple open-ended design challenges. The Crumble was developed in the United Kingdom to provide an easier method for primary (elementary) level teachers to embed electronics in their design and technology (D&T) curriculum. Design and Technology teachers in the UK have been using it to teach computing and electronics concepts embedded within product/systems intelligence (Birks, 2018; Brown, 2016). The Crumble has been used at the elementary level to teach myriad STEM concepts and also at the middle and high school levels to introduce programming and sensor systems before exploring other microcontrollers that require advanced coding skills. The Crumble is now available for purchase in the U.S. through TeacherGeek®.

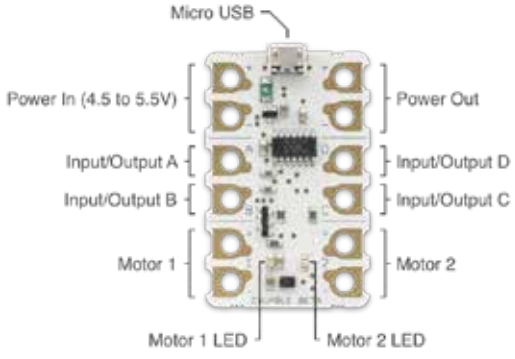
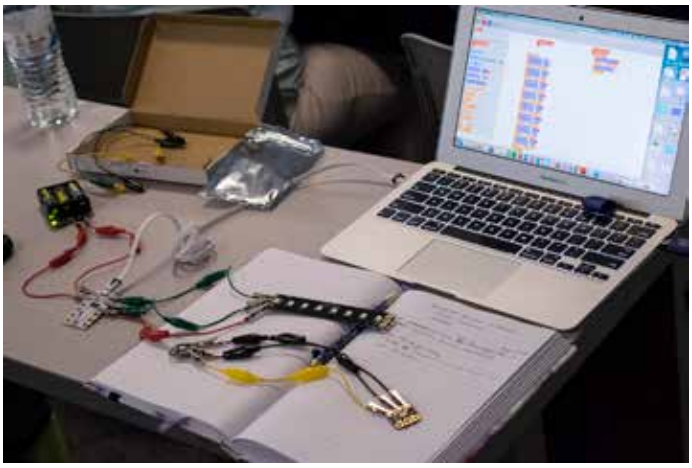


Figure 1. The Crumble Controller (Redfern Electronics, 2018).



Preparing to Use the Crumble

Before using the Crumble, the instructor must be prepared to teach some basic electronics concepts so students can successfully design their circuits. The authors found it helpful to create kits containing an array of sensors, wires with alligator clips, and a Crumble for each student or group. Computers to be used for programming the Crumble will need to have the free software installed from the Crumble website (Redfern Electronics, 2018). Moreover, the instructor should be familiar with how to program various sensors using the software. The Crumble website provides a wealth of excellent instructional resources and activities to learn how to program each sensor (Redfern Electronics, 2019).

The Collision Avoidance Design Challenge

There are many sensors and advanced electronic components in vehicles today. These components work together to create a safer technological system (automobile). According to the National Highway Traffic Safety Administration (NHTSA), 391,000 people were injured by distracted driving in 2015 (NHSTA, 2018). The Insurance Institute for Highway Safety (IIHS) reported that the rate of accidents in vehicles with a collision-avoidance sensor system was 11% less than cars without this feature. The IIHS predicted that if all passenger vehicles had been equipped with this technology in 2015, it would have prevented more than 55,000 injuries (LeBeau, 2017).

In this design challenge students were tasked with creating a small-scale, low-cost collision-avoidance system that could fit into a model car. They also had to devise a way for the Crumble and sensors to help avoid a collision. Students were allowed to safely use the Crumble, any compatible sensors, wires, wire strippers, balsa wood, cardboard, scissors, hot glue guns, small motors, batteries, plastic wheels, straws, and axles, at a minimum. The design brief can be downloaded from the Crumble Design Challenge Resources (2018) website. A benefit of using the Crumble to solve open-ended design challenges is the multitude

of content standards that can be addressed while completing the challenge. Table 1 provides an overview of some of the standards and skills addressed in the context of this design challenge.

Objectives

Objectives are an important component of any unit or lesson. For this design challenge, the authors chose to use performance objectives. These types of objectives include the condition under which learner behavior or performance is to occur, the expected behavior or performance of the learner that is to be observed, and the minimally acceptable level of performance or criterion that is to be achieved (Orlich, et al., 2013). Some of the performance objectives used for this design challenge were:

1. Using the vocabulary terms, students will be able to correctly describe the function of all Crumble sensors with 100% accuracy.
2. With electronics knowledge from the explore and explain phases, students will be able to design an accurate schematic and working circuit involving at least three sensors.
3. Using the Crumble software, students will be able to troubleshoot and program the Crumble so that the ultrasonic sensors slow and stop the vehicle when sensing an obstacle at varying distances.
4. With the materials provided, students will be able to design and safely construct a prototype of a car that uses the Crumble and appropriate sensors to avoid collisions.
5. Using data collected from testing their vehicle, students will be able to accurately calculate the deceleration of their car and apply that information to modify their Crumble program for safer stopping distances.

Table 1. Standards Addressed

Source	Standards, Subconcepts, and Skills
Standards for Technological Literacy	3D: Technological systems often interact with one another. 10F: Troubleshooting is a problem-solving method used to identify the cause of a malfunction in a technological system. 16H: Power systems are used to drive and provide propulsion to other technological products and systems. 17H: Information and communication systems allow information to be transferred from human to human, human to machine, and machine to human.
K-12 Computer Science Framework	Troubleshooting, Control, Program Development
Next Generation Science Standards	MS-PS2-2, MS-ETS1-1, MS-ETS1-2, MS-ETS1-3, MS-ETS1-4, Crosscutting Concept: Systems and System Models
21st Century Skills	Critical thinking and problem solving, Creativity and innovation, Information and communications technologies literacy, Flexibility and adaptability

For this design challenge the authors followed the 6E Learning byDeSIGN™ Model (Burke, 2014). In the 6E model, the "engineer" phase is intentionally added to the traditional 5E model. This helps emphasize design and engineering concepts that are integral to STEM instruction (Love & Deck, 2015). Within the context of this design challenge, the engineer phase helped demonstrate the importance of safely constructing a prototype (balsa vehicle), which provided an authentic designed-world application for the programming portion of the challenge.

Engage

Students were first introduced to the following vocabulary terms using the CSSR (Context, Structure, Sound, Reference) technique: collision, motor, power source, ultrasonic sensor, servomotor, and photovoltaic (PV) cell. They were instructed to read the text and use context clues, word structure, sound, and reference materials to define the vocabulary words. The instructor then corrected all misconceptions by establishing definitions of the vocabulary terms. Next the instructor showed students videos about the future of autonomous vehicles (The Verge, 2018) and how collision-avoidance systems work (SaferCarTV, 2017). These videos sparked discussions about the economical, ethical, environmental, and political impacts of emerging vehicle technologies. If using the Crumble at the elementary level, teachers could incorporate a reading-level-appropriate book related to cars.

Explore

During the "explore" phase the instructor allowed students to experiment with the circuit components, observing what each sensor does. Another method would be to allow students to experiment with the Crumble and complete some of the basic tutorials provided in the *Crumble Getting Started Guide* document and on the Projects page (Redfern Elec-

tronics, 2018, 2019). The Using Sparkles, Police Lights, and Parking Sensor tutorials are all excellent introductions to programing the sparkles (Crumble LEDs) and ultrasonic distance sensor that is used in this design challenge.

Explain

Now that students have had some exposure to the Crumble components and software, the instructor can explain some basic concepts needed to construct a successful Crumble circuit. First, basic electronics concepts (e.g., resistance, series circuit, parallel circuit, etc.) need to be explained and demonstrated for students who are not familiar with them. The instructor showed students how to develop a successful program using the Crumble and demonstrated some common troubleshooting techniques. Topics such as vehicle design characteristics, aerodynamics, friction, gear ratio, deceleration, and other content integral to this design challenge were discussed with students. All machine/tool demonstrations and safety tests were also completed at this time. Lastly, students were introduced to the design challenge scenario and criteria to set the stage for the engineering phase.

Engineer

During this phase students were first tasked with drawing a schematic of how their circuit would be assembled. The instructor checked this before students could proceed to constructing their circuit. When constructing their circuit, students were told to focus on getting the system set up to operate correctly using the Crumble. It should slow the motor and turn on the sparkles (brake lights) when an obstacle is detected within a certain distance, and fully stop the motor with the sparkles still lit when the obstacle is at a closer distance. An example of a successful Crumble program is shown in Figure 2.

Figure 2. Example of a working code and explanation of the code for the collision avoidance design challenge.

After students assembled their circuit and had it working correctly, they were tasked with designing an efficient vehicle that would house their crumble components. Their vehicle prototype was designed out of basic materials such as cardboard and balsa wood (Figure 3). The Card Buggy listed in Table 3 (page 21) is another cost-efficient and user-friendly option. Some challenges students encountered during assembly were: (1) ensuring the gears from the motor and wheels lined up correctly [unless they used the Crumble wheels, which do not require external gears], and (2) making sure the straws for all axles were straight so the vehicle would drive straight. Once students had a prototype developed, they tested it and documented all modifications to their vehicle or Crumble program.

Enrich

To enrich the lesson, students were tasked with designing a more advanced prototype. They first began by measuring the size of all essential Crumble sensors using a Vernier or digital caliper. As they documented their measurements, they were encouraged to think about design aspects that would make their vehicle more efficient. After obtaining all measurements, students were introduced to a 3-D design software from which they could 3-D print. In this example students were taught how to use the OnShape software. OnShape is a free, cloud-based CAD software

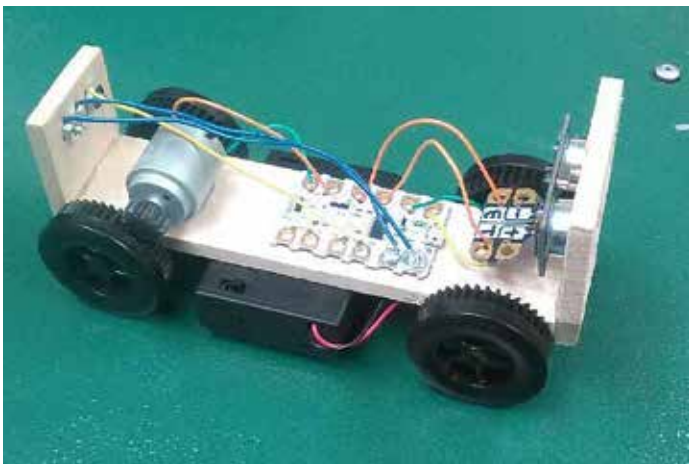


Figure 3. An example of a balsa wood prototype vehicle.

developed by the founders of SolidWorks. It allowed students to design, assemble, and animate 3-D parts from any computer or mobile device that has internet access. Multiple students could work on the same project, seeing changes in real time similar to a Google Doc. The instructor could track which students made edits and exactly when they occurred. The authors chose OnShape because of its wealth of tutorials for students, accessibility outside of the classroom, free access, and cloud-based platform that did not require any software installation by school personnel. Students saved their final designs and 3-D printed them, then added all sensors and tested their custom-built vehicle. The design shown in Figure 5 took approximately 12 hours to print; however, stricter requirements could have been instituted to limit print times. Allowing students to design and 3-D print a custom vehicle provided opportunities to consider various design aspects. If instructors do not wish to have students design their own 3-D printed vehicle, the STL file depicted in Figure 4 can be downloaded from the Crumble Design Challenge Resources (2018) website.

Figure 4. Example of a computer-generated vehicle design using OnShape.

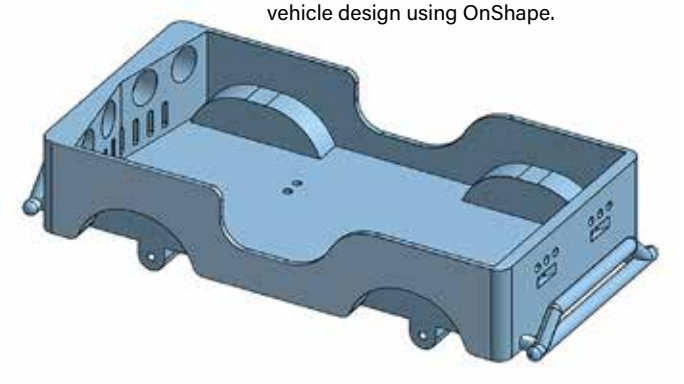


Table 2. Evaluation Rubric

	Exemplary - 5 points	Proficient - 3 points	Needs Improvement - 1 point
Sketches/3-D Designs	Produced very detailed sketches or computer-aided designs of their ideas. Included accurate measurements and detailed labels for all essential components.	Produced sketches or computer-aided designs of their ideas. Inaccurate measurements and/or was missing detailed labels for all essential components.	Produced computer-aided designs of their ideas. Did not include measurements and labels for all essential components.
Developing Alternative Solutions	Generated many well-articulated iterations and provided a clear rationale for their final design selection.	Generated a few iterations without a clear rationale for their final design selection.	Generated one iteration with or without a rationale for their final design selection.
Design Constraints	Satisfied all requirements within the constraints of the challenge.	Satisfied most requirements within the constraints of the challenge.	Satisfied a few requirements within the constraints of the challenge.
Fabrication	The entire system was neatly and sturdily constructed. It operated effectively during testing.	The system was constructed and operated during testing.	The system was constructed.
Sensors and Schematics	Used all the correct sensors and provided detailed documentation about each in the schematic.	Used all the correct sensors but provided vague or inaccurate documentation in the schematic.	Used the incorrect sensors.
Programming	Program had no errors and successfully operated the technological system within the parameters of the challenge.	Program had a few errors that prevented the system from successfully operating.	Program was incomplete.
Science and Math Applications	Correctly calculated the deceleration for multiple trials and accurately displayed results in a graph.	Minor errors in the deceleration calculations or graph.	Incomplete deceleration calculations or graph.
Final Solution	When sensing an obstacle, the vehicle turned on the brake lights and slowed down to avoid a collision.	The vehicle turned on the brake lights or slowed down when sensing an obstacle.	The vehicle operated, but the brake lights and collision-avoidance system did not work.
Presentation	Provided thorough and accurate details about the design of their vehicle, the Crumble code, their deceleration data, and applications of the engineering design process.	Provided some details about the design of their vehicle, the Crumble code, their deceleration data, and applications of the engineering design process.	Provided vague details about the design of their vehicle, the Crumble code, their deceleration data, and applications of the engineering design process.
Teamwork and Safety	Student provided significant contributions to the solution while following all safety procedures.	Student provided some contributions to the design solution or had at least one safety violation.	Student contributed very little to the design solution or had numerous safety violations.

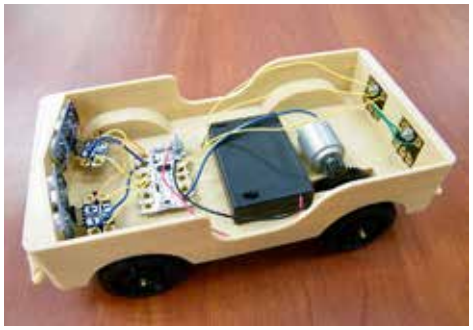


Figure 5. Example of a working 3D-printed design that stops and displays brake lights when sensing an object in its path.

This design challenge presented a great way for students to see math and science concepts in action. One method used to integrate physics practices was to ask students to calculate the deceleration of the vehicle from the time it sensed an obstacle until it came to a complete stop. To calculate this, students needed to first calculate the velocity of the vehicle at its top speed. This was done with the simple formula of velocity = distance x time. Students had to time how long it took their vehicle to travel a distance (set by the instructor) while moving at its maximum speed during that entire distance. Once students knew the velocity, they could then calculate the deceleration using the following formula: Deceleration=$(v_f-v_i)/t$. Students knew the initial velocity (v_i) from their previous calculation, and the final velocity (v_f) was 0 because the vehicle came to a complete stop. They had to time (t) how long it took the vehicle to come to a complete stop at the first sign of the brake lights turning on and the motor slowing down. Once students solved this, they could then make informed predictions to adjust the motor speed or sensor distance in their Crumble code to see how fast their vehicle could travel, yet still stop in time to avoid a collision. Students calculated the deceleration for their various trials and graphed them for comparison. The topic of deceleration provided opportunities to purposefully integrate additional physics and engineering concepts such as Newton's First and Second Laws of Motion, friction, gear ratio, momentum, and horsepower.

Evaluate

The rubric in Table 2 provides a template with criterion to evaluate students' design solutions. It can be adapted to evaluate criterion the instructor deems most important and is downloadable from the Crumble Design Challenge Resources (2018) website. One of the strengths of this design challenge is that there were various portions (e.g., circuit design, Crumble code, vehicle design) that must be completed correctly in order for the entire system to operate efficiently. This showed students how multiple technologies were needed to produce a working technological system. In addition to the summative evaluation in Table 2, an engineering notebook was used as a formal assessment.

Conclusions

From the example provided in this article it is evident that the Crumble can provide a cost-effective way to introduce elementary and middle school students to programming, electronics, engineering design, and other designed-world concepts. The user-friendly drag-and-drop format of the free Crumble software also allows students to gain a richer understanding of programming and troubleshooting. Educators at all levels in the United Kingdom have developed various applications for the Crumble that could very easily be implemented in other countries (Table 3).

Table 3. Additional Crumble Design Challenge Applications

Design Challenge	Source
Parking Sensor	https://redfernelectronics.co.uk/projects/parking-sensor/
Automated Trash Bin	https://redfernelectronics.co.uk/projects/creating-automation-project/
Card Buggy	https://redfernelectronics.co.uk/cardbuggy/
River Pollution Monitoring System	https://community.computingatschool.org.uk/resources/5229/single
Fairground Ride	https://community.computingatschool.org.uk/resources/4167/single
Robot Instruments/ Orchestra	www.manchester.ac.uk/connect/teachers/teacher-events-resources/resources/robot-orchestra-kit/

The greatest benefit of the Crumble is its ability to help teachers and students integrate programming within the context of hands-on engineering design challenges. Educators in other countries, especially in the United States where there has been a recent emphasis on computer science in every school, should consider the Crumble as a viable tool for integrating programing and engineering design in the elementary and middle grades. Furthermore, the design challenges presented in this article could serve as a foundation for similar applications utilizing other microcontrollers that require a deeper understanding of coding (ex. Raspberry Pi, Arduino) (Love, et al., 2016).

References

Birks, J. (2018). Q&A – *What is the crumble controller?* Retrieved from www.teachwire.net/products/qa-what-is-the-crumble-controller

Brown, S. (2016, July 14). *Move over raspberry pi, hello crumble!* Retrieved from https://swgfl.org.uk/magazine/move-over-raspberrypi-hello-crumble/

Buckler, C., Koperski, K., & Loveland, T. R. (2017). Is computer science compatible with technological literacy? *Technology and Engineering Teacher, 77*(4), 15-20.

Burke, B. N. (2014). The ITEEA 6E learning byDeSIGN™ model: Maximizing informed design and inquiry in the integrative STEM classroom. *Technology and Engineering Teacher, 73*(6), 14-19.

Code.org. (2018). *Promoting computer science.* Retrieved from https://code.org/promote

Crumble Design Challenge Resources. (2018). *Collision avoidance design challenge.* Retrieved from https://sites.google.com/a/vt.edu/crumble/

LeBeau, P. (2017, August 23). *New report shows how many accidents, injuries collision avoidance systems prevent.* Retrieved from www.cnbc.com/2017/08/22/new-report-shows-how-many-accidents-injuries-collision-avoidance-systems-prevent.html

Love, T. S. & Deck, A. (2015). The ocean platform engineering design challenge - Flooded with STEM content and practices. *Science Scope, 39*(3), 33-40.

Love, T. S. & Strimel, G. (2017). Computer science and technology and engineering education: A content analysis of standards and curricular resources. *The Journal of Technology Studies, 42*(2), 76-88.

Love, T. S., Tomlinson, J., & Dunn, D. (2016). The orange pi: Integrating programming through electronic technology. *Technology and Engineering Teacher, 76*(2), 24-29.

Moon, C. (2018). Learning programming through baking. *Technology and Engineering Teacher, 77*(8), 28-30.

NHTSA (National Highway Traffic Safety Administration). (2018). *Distracted driving.* Retrieved from www.nhtsa.gov/risky-driving/distracted-driving

Orlich, D. C., Harder, R. J., Callahan, R. C. Trevisan, M. S., Brown, A. H., & Miller, D. E. (2013). *Teaching strategies: A guide to effective instruction*, (10th ed). Belmont, CA: Wadsworth Cengage Learning.

Redfern Electronics. (2018). *The crumble controller.* Retrieved from https://redfernelectronics.co.uk/crumble/

Redfern Electronics. (2019). *Guide to using crumbles.* Retrieved from https://redfernelectronics.co.uk/getting-started/guide-to-using-crumbs/

SaferCarTV. (2017, March 30). *Adam Savage explains forward collision warning* [Video File]. Retrieved from https://youtu.be/GpwjcsH5aXU

The Verge. (2018, January 30). *The state of self-driving cars: 2018* [Video File]. Retrieved from https://youtu.be/_Zd1ByhigPU

Wright, G. A., Rich, P., & Leatham, K. R. (2012). How programming fits with technology education curriculum. *Technology and Engineering Teacher, 71*(7), 3-9.

All images credit: Penn State Harrisburg.

Tyler S. Love, Ph.D. *is an Assistant Professor of Elementary/Middle Grades STEM Education and Director of the Capital Area Institute for Mathematics and Science (CAIMS) at The Pennsylvania State University, Harrisburg. He can be reached at tsl48@psu.edu.*

Abraham Bhatty *is an undergraduate student in the T&E teacher preparation program at the University of Maryland Eastern Shore. Abraham can be reached via email at aabhatty@umes.edu.*

This is a refereed article.

Ad Index

Kelvin 43

Mastercam 44