SMART castles:

integrating engineering design and computational thinking

By Scott R. Bartholomew and Nathan Pehrson

The nature of the activity necessitated conversation, discussion, and even negotiation with fellow students to produce viable and functioning castle pieces.

Introduction

A group of adults, dressed in authentic medieval costumes, organize and stage a full-scale battle at a local park; weapons (made from foam), roles, and medieval speech are all part of the event (see Figure 1). This is known as *Live Action Role Play* – more commonly referred to as "LARP." LARP itself does not have a single point of origin, rather, it has evolved over time as individuals have desired to transition from tabletop role-playing games to physical settings (Widing, 2008). In its most basic sense, LARP signifies a group of people coming together to act something out—each person playing a character in the overall experience (Snyder, 2018).

Although many settings and scenarios have been used in LARP functions (e.g., medieval, Victorian, fantasy), the most common are medieval, fantasy, or a combination of both (Konzack & Dall, 2008). LARP functions range from simple to complex and often include "gamemasters" who establish the rules, assign characters, and dictate the plot in real time (Manuel, 2015). Some LARP events last only a few hours while others may take days to resolve (Druce-McFadden, 2015). The prevalence of LARP, hundreds of years after the

Figure 1. *A scene from a LARP event in Missouri.*

time it most often portrays, is simply one expression of our society's romantic view of medieval times and everything that went with them (Parks, 2013). Interestingly, many of the concepts so popular in LARP (e.g., physical battle, unsophisticated weaponry, etc.) stand in stark contrast to many technological innovations of today.

From their perspectives as a Technology and Engineering Education (TEE) professor and an undergraduate TEE student, the authors have noticed this romantic view of medieval times in many of the K-12 students they work with, who just seem to like all things medieval! They set out to explore medieval ideas with a specific goal of creating an activity—which capitalized on this interest—that purposefully integrated medieval concepts with current ideas in engineering design and computational thinking. They eventually settled on "modernizing" one classic element of medieval times: the castle. The experience, shared herewith, was enjoyable and fulfilling—and the approach (connecting modern with antiquity) should be considered as a potential model for TEE teachers as they plan their lessons. Presented here is a brief description of related topics, followed by findings from the authors' own implementation of this activity into a local school.

Figure 2. *Motte and bailey castle (Shaw, 2001).*

Figure 3. *Stone castle with moat (McCallum, 2008).*

Medieval Castles

Castles are generally defined as a type of fortified private residence of a lord or other noble. This distinction separates castles from palaces, which were private residences, but not fortified, and fortresses, which would not have served as private residences. While the Norman Conquest in 1066 marks the beginning of the medieval castle-building age (Barrow, 2013), the picturesque stone castles most people imagine today did not come into favor until the 13th century (Johnson). The first Norman castles (see Figure 2) are referred to as motte and bailey castles and consisted of a wood or stone fortress atop an artificial mound called a motte; these were surrounded by a walled courtyard called a bailey (Johnson). As motte and bailey castles fell out of favor, castles began to be built almost entirely out of stone (see Figure 3) and were often surrounded by a moat (word adapted from motte) with water (Cartwright, 2018).

Most common castle design features are iconic and easily recognizable. For example, the moat of water around the outside, the drawbridge over the moat, the large metal or wooden grid called a portcullis, the towers on the corners of outer walls (bastions), and the merlons, which are the square saw teeth along battlements (Johnson). However, there are some additional key design features of castles that are not as well known. A battlement, for example, refers to a small wall built on walkways atop the outer walls to protect soldiers from attack (Johnson). Other lesser-known design features include the keep, barbicans, and the fortified gatehouse. The keep is typically the tallest tower of a castle and is also the strongest section of the castle—with thicker walls and usually a single entrance. Keeps served as the heart of the castle and a place of retreat should the outer (or curtain) wall be breached. Barbicans are fortified structures made to protect potentially weak sections of the castle (usually a gate). They may include a short section of fortified wall, defensive towers, and may even have a dedicated outer wall surrounding it. The fortified gatehouse, although similar to a Barbican, refers specifically to the main entrance of the castle. Fortified gatehouses often consist of large twin towers with a gate tucked in between them. The gate is also protected by heavy wooden doors and at least one—and sometimes two—portcullises (Cartwright, 2018).

Engineering and Design

Engineering design is an iterative process used to identify problems and develop and improve solutions (Engineer Girl, 2021). Engineering design includes different processes, steps, and pathways towards creating a solution to a problem. While there is no agreed upon "engineering design process," most approaches, although different, include common steps such as: defining a problem, planning a solution, creating, testing, and refining (ITEEA, 2020). The engineering design process is not something used only by classically trained engineers, rather, it is a problem-solving approach being increasingly taught to all students (Fan, Yu, & Lin, 2020). For example, *Standards for Technological and Engineering Literacy* (ITEEA, 2020) emphasizes the importance of engineering

design as an approach to creative problem solving; additionally, *Next Generation Science Standards* (NGSS, 2013) incorporates engineering design in the key science and engineering practices for all students.

One important aspect of engineering design is that of constraints; it is imperative to recognize the limits and needs of any design—what the final product must do (or not do)—to effectively employ engineering design towards creating solutions (Link Engineering, 2021). Further, engineering design is *different* from other problem-solving processes (e.g., scientific inquiry) in that it emphasizes creatively solving problems while other approaches, such as scientific inquiry, often center on asking and answering questions (e.g., through observations and experimentation; (Science Buddies, 2021).

Research supports the incorporation of the engineering design process into K-12 classrooms as a problem-solving process that students can use both inside and outside the classroom (Mangold & Robinson, 2013). While both challenges and opportunities come with engineering design in K-12 classrooms (Arik & Topçu, 2020), there have generally been very positive results from its addition to the curriculum.

Computational Thinking

Computational Thinking is a problem-solving framework that focuses on approaching the problem logically and algorithmically. Computational Thinking works by trying to "present [the] solutions in a way that a computer, a human, or both, can understand" (B.B.C.). Like Engineering Design, there is no set definition for Computational Thinking, but most sources agree that there are four main cornerstones (K12CS, 2021): Decomposition, Pattern Recognition, Abstraction, and Algorithm Design. In recent years there has been a push to include more Computational Thinking in schools because it encourages the development of "abstract thinking, problem solving, pattern recognition, and logical reasoning" (Angei & Giannakos, 2020).

A common misconception about Computational Thinking is that it is "thinking like a computer" (Victoria, 2018). A more accurate statement may be that Computational Thinking is thinking like a computer scientist. While the difference may small, it is an important to understand. Thinking like a computer suggests that the student solves the problem as a computer would. Thinking like a computer scientist means students are solving the problem by breaking it down and "expressing solutions as computational steps" (K12CS, 2021). In other words, Computational Thinking can be thought of as translating a complex problem so that a computer can help solve it.

Combining Engineering Design, Computational Thinking, and Medieval Castles: What We Did and How it Went

Students were tasked to work in groups of three to design, build, and automate a medieval castle! A pilot of this work was tested with an advanced engineering middle school class (24 students in eight groups of three). Each student was given a role in their group (designer, builder, or programmer) based on their own preferences and their team member strengths. Given the time constraints, it was determined not to be feasible for every student group to build their own castle—rather, each group was assigned to build one section of a larger castle. This castle was to be placed on a three-by-three grid (see Figure 4) with each group completing one section and all pieces being joined together with wall segments at the conclusion (see Figure 4). This represented a crucial design constraint as students needed to work collaboratively to ensure that their pieces fit with the groups immediately adjacent to them (see Figure 5). To help students design within the constraints, each group was given a baseplate with precut holes for both the walls and castle they were designing, and the baseplates were all designed to fit together. In addition to designing (using Adobe Illustrator), cutting (laser-cut Baltic birch plywood – 1/8" thick), and assembling (using super glue and activator), the students were tasked with automating some portion of their castle segment. This automation took many forms and students were excited to explore creative solutions to this challenge; the final castle had a working door, a flagpole that raised a flag up and down, a spinning Christmas tree, almost 50 LEDs that blinked in time with music, and much more. The project pilot took place during eight 45-minute class periods (see Table 1 for a day-to-day list of topics covered).

Day 1	Introduce the project, assign groups, brainstorming
Day 2	Finishing brainstorming and beginning paper plans
Day 3	Finishing and approving paper plans
Day 4	Beginning illustrator files and automation instruction
Day 5	Workday; Illustrator files and automation
Day 6	Workday; Illustrator files, automation, test cuts
Day 7	Workday; Illustrator adjustments, automation, final cuts
Day 8	Final assembly and presentations

Table 1.

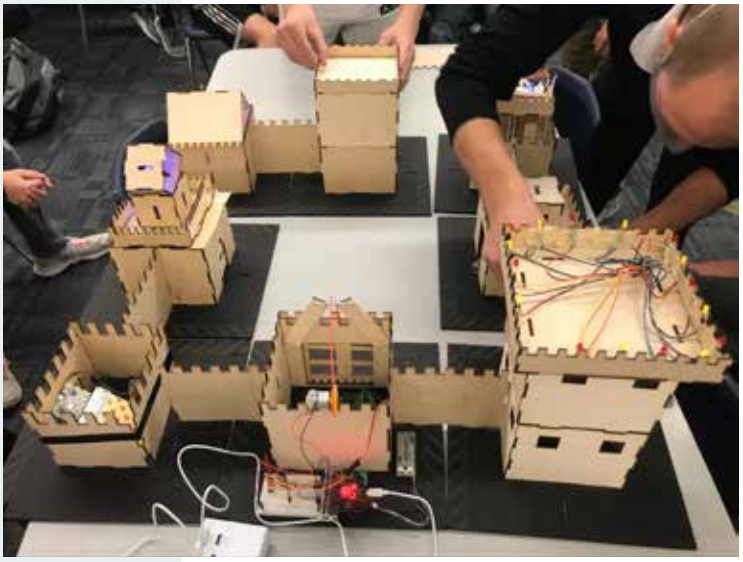
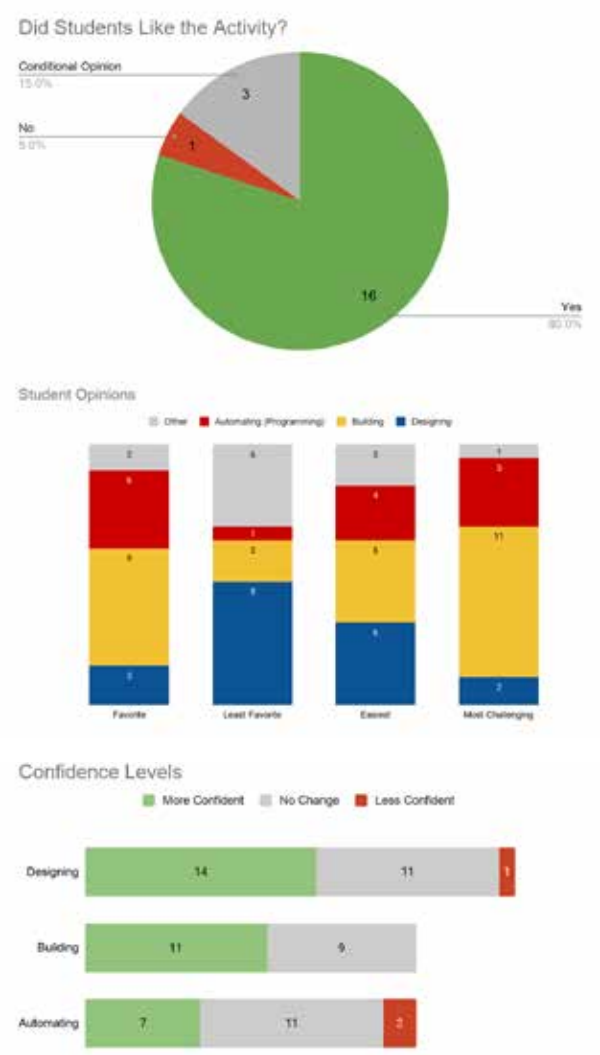


Figure 4. *Putting together each student's section in a 3x3 grid.*

At the conclusion of the project, students were surveyed to identify the strengths and weaknesses of the activity to improve the overall project and investigate the student experience. Survey results are included below; results provided insight into the students' impressions of the activity, their preferences for the various stages/roles included in the castle design and automation process, and their confidence levels following participation.

Survey Results



Conclusion:

The most apparent success based on collected data was that the students liked the activity. An additional success showed that, although students struggled with the design portion of the activity, there was a greater confidence increase in design than the other sections of the activity. Additionally, the teacher shared that several students had commented on how they were able to identify and implement the engineering design process in the activity because of the constraints inherent in designing one aspect of a larger building. This suggests that the activity was most successful in the engineering design section and least successful in the automation (programming) section. It is important to consider, however, that students spent the most time on the designing portion of the activity and that only about 1/3 of the students actively participated in the coding aspects of the activity.

Despite some positive takeaways from the activity, several challenges also became apparent, chiefly: the activity (especially the design/build portion) took much more time than had been initially anticipated. Programmers (one from each team) were asked to work on the automation of their castle segments simultaneously while the builders and the designers were constructing. While this was a deviation from the initial plan and introduced some further complications, it was considered to be a success overall. The survey showed little increased confidence in programming—perhaps because of the structure employed in class—or perhaps as a function of the activity.

Further, in observations of the students—and in conversations with the teacher—the authors noted that the impetus required for collaboration may have been one of the greatest lessons learned through the activity. The nature of the activity necessitated conversation, discussion, and even negotiation with fellow students to produce viable and functioning castle pieces. Ensuring that your team's piece fit—in both form and function—with both of the neighboring pieces was a challenge for all students and the required collaboration was a large part of the benefits noticed within this lesson.

Lesson Plan:

SMART Castles: Integrating Engineering Design and Computational Thinking

Grade Level: Grades 6-9

Overview: Students are divided into groups of two and each team is responsible for designing, building, and automating a section of the castle.

Big Idea: Introduce students to computer science and computational thinking by having them automate a castle they built themselves.

Enduring Understandings:

- Students will feel that programming and computational thinking are fun and approachable.
- Designing with constraints is more useful than designing without constraints.
- Ability to work within and coordinate between teams.

Purpose of Lesson

- Allow students to implement an engineering design process in self-directed project.
- Introduce students to programming and computational thinking.
- Increase student confidence in designing and building something.

Instructional Time:

- Recommend 10-14 class periods: 1 hour each.

Standards/Benchmarks

Standards for Technological and Engineering Literacy (STEL)	
2N	Illustrate how systems thinking involves considering relationships between every part, as well as how the system interacts with the environment in which it is used.
2S	Defend decisions related to a design problem.
7Q	Apply the technology and engineering design process.
8J	Use devices to control technological systems.
Next Generation Science Standards (NGSS) Benchmarks	
MS-ETS1-1.	Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.
MS-ETS 1-2.	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
Common Core Mathematics Standards (CCSS Math) Benchmarks	
6.RP.3	Use ratio and rate reasoning to solve real-world and mathematical problems, e.g., by reasoning about tables of equivalent ratios, tape diagrams, double number line diagrams, or equations.
Common Core English Language Arts Standards (CCSS-ELA) Benchmarks	
ELA-Literacy. RST.9-10.3	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.

Learning Objectives

- Students will be able to identify and define design constraints.
- Students will be able to use variables and basic logic when programming.
- Students will be able to identify how design features on castles were influenced by the historical context.

6E Lesson Plan Learning Highlights:

Engage: pique student interest and get them personally involved in the lesson, while pre-assessing prior understanding. **[1 hour]**
The Engage phase occurs during the first class period. The teacher will present an example castle, share the historic design constraints in castle building, and introduce the challenge of building their own castle to the students.
Explore: provide students with the opportunity to construct their own understanding of the topic. **[2 hours]**
Students work in teams to brainstorm what they would like to do for their castle and create drawings of the pieces they will need to construct their portion of the castle. Students also coordinate with teams over neighboring sections of the castle to ensure fit.
Explain: provide students with an opportunity to explain and refine what they have learned so far and determine what it means. **[2 hours]**
Students pass off drawings and then transition to creating vector files for the laser. Students explain what they are building, how each piece fits together, and how they will automate their castle prior to cutting on the laser.
eNGINEER: students have the opportunity to develop greater depth of understanding about the problem topic by applying concepts, practices, and attitudes. **[4 hours]**
Students work through design iterations and prototyping by cutting designs out using the laser (first with cardboard and then with 1/8" baltic birch plywood).
Enrich: provide students with an opportunity to explore in more depth what they have learned and to transfer concepts to more complex problems. **[2 hours]**
Students use a micro:bit and makecode.org to automate a portion of their castle section. Iterative cycles of design, test, and refine will be used to improve their automation.
Evaluate: review the understanding of students and assist with any misconceptions. **[1 hour]**
Student groups give final presentation and assemble the final castle. Students discuss successes and challenges and address any lingering questions.

Required Tools/Materials/Equipment:

The following is a list of supplies and equipment needed to teach this lesson.

- Laser cutter
- Foam core for castle base (alternatively, use cardboard)
- ⅛ inch cardboard sheets
- ⅛ inch plywood (e.g., baltic birch)
- CA super glue
- CA Glue activator

Lab/Classroom Safety and Conduct:

- Accountability
- Glue and activator safety
- Basic electronics safety

Student Resources:

- Student Kits (1 for each team)
  - Micro:bit starter kit
  - Graph paper
  - Group base plate
  - Example connecting wall

Technologies and Other Material Resources:

- Adobe Illustrator
- makecode.microbit.org

Vocabulary: (teachers)

- Portcullis: The metal or wood grid that can be lowered over castle entrances.
- Bastions: Towers at the end of castle walls.
- Merlons: The square saw teeth on top of castle walls.

Standards-Based Assessment:

- Formative assessment
  - 2D scale drawings
  - First Illustrator file and prototype
  - Finished cardboard Prototype and corrected 2D scale drawings
  - Automation Prototype
- Summative assessment
  - Presentation and finished castle section.

Supplies:

Item	Cost	Link
CA Glue	$10.99	www.amazon.com/Star-bond-EM-2000-Thick-PREMI-UM-Woodturning/dp/B00C32MKLK/ref=sr_1_4?crid=FQ2AM8Y1YN-QN&dchild=1&keywords=starbond+-ca+glue&qid=1615838974&sprefix-=starbond%2Caps%2C195&sr=8-4
	$10.50	www.starbond.com/products/thick-ca-glue-em-2000 (Sometimes available in larger sizes here)
CA Glue Activator	$14.40	www.amazon.com/dp/B00BU-VAZ5S/ref=sspa_dk_detail_3?p-sc=1&spLa=ZW5jcnlwdGVkUXVhb-GlmaWVyPUEyT0EyUldXTVI5RTNE-JmVuY3J5cHRlZElkPUEwMTk1MTUx-MVZETlBXRjBLODM1ViZlbmNyeX-B0ZWRBZElkPUEwMjY2NDQ3M-kNRSFhZSDk2RlBPNiZ3aWRnZXROY-W1lPXNwX2RldGFpbDImYWN0aW9uP-WNsaWNrUmVkaXJlY3QmZG9O-b3RMb2dDbGljaz10cnVl
	$14.40	www.starbond.com/products/aero-sol-accelerator
Microbit Kit		https://stemeducationworks.com/product/starter-kit/
8GB Flash Drives (10 pack)	$24.99	www.amazon.com/Aiibe-Flash-Drive-Drives-Memory/dp/B07PQF9X75/ref=sr_1_4?dchild=1&keywords=8gb+flash+drive&qid=1618234250&sr=8-4
Cardboard	$41.99	www.amazon.com/BOX-USA-BSP2418-Corrugated-Sheets/dp/B01BGFYRQU/ref=sr_1_6?dchild=1&keywords=cardboard+sheets&qid=1618234397&sr=8-6
Foam Core	$1.00	www.dollartree.com/black-foam-boards-20x30-in/25957
18" x 24" Sheets (10 pack)	$30.15	www.amazon.com/Board-Center-18x24-Backing-Boards/dp/B01HOXQDU6/ref=sr_1_4?crid=2VWFM54E3HH-J&dchild=1&keywords=18x24+foam+-core+board&qid=1618235825&sprefix-=18x24+foam%2Caps%2C193&sr=8-4
⅛ plywood	$78.99	www.amazon.com/Baltic-Birch-Plywood-Veneer-Sheets/dp/B07TN3G2WW/ref=sr_1_5?d-child=1&keywords=1%2F8+ply-wood+18x24&qid=1618235206&sr=8-5

References

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185.

Arık, M., & Topçu, M. S. (2020). Implementation of engineering design process in the K-12 science classrooms: Trends and issues. *Research in Science Education*, 1-23.

Barrow, Mandy. (2013). *Castles and the Medieval World*. Primary Homework Help. www.primaryhomeworkhelp.co.uk/Castles.html

B.B.C. (n.d.). *What is computational thinking?* www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1

Cartwright, Mark. (2018, May 17). *Medieval Castle*. World History Encyclopedia. www.worldhistory.org/Medieval_Castle/

K12CS.org (n.d.), *Computational Thinking*. Retrieved July, 31, 2021, from https://k12cs.org/computational-thinking/

Druce-McFadden, C. (2015, October 7). The Beginners Guide to LARPing. Geek and Sundry. https://geekandsundry.com/interview-mackenzie-jamieson-on-how-to-get-into-LARPing/#:~:text=There%20are%20different%20types%20of,time%20down!"%20Sunday%20morning

Engineer Girl. (n.d.). *What is Engineering Design?* www.engineergirl.org/128119/engineering-design

Fan, S. C., Yu, K. C., & Lin, K. Y. (2020). A framework for implementing an engineering-focused STEM curriculum. *International Journal of Science and Mathematics Education*, 1-19.

International Technology and Engineering Educators Association. (2020). *Standards for technological and engineering literacy: The role of technology and engineering in STEM education.* www.iteea.org/STEL.aspx

Johnson, Ben. (n.d.). *The History of Castles*. Historic UK. www.historic-uk.com/HistoryMagazine/DestinationsUK/History-of-Castles/

Konzack, L., & Dall, I. (2008). *Fantasy and Medievalism in Role–Playing Games*. Playground worlds.

Link Engineering. (n.d.) *What is Engineering Design?* From www.linkengineering.org/EngineeringDesign.aspx

Mangold, J., & Robinson, S. (2013, June). The Engineering design process as a problem solving and learning tool in K-12 classrooms. In 2013 ASEE Annual Conference & Exposition (pp. 23-1196).

Manuel, R. (2015, October 8). *3 mistakes every game master makes and how to fix them.* geek and sundry. https://geekandsundry.com/3-mistakes-every-game-master-makes-and-how-to-fix-them/

McCallum, A. (2008). Bodiam Castle East Sussex England UK [Photograph]. *Wikipedia.* https://en.wikipedia.org/wiki/File:Bodiam-castle-10My8-1197.jpg

NGSS Lead States. 2013. Next generation science standards: For states, by states. Washington, DC: The National Academies Press.

Parks, C. (2013, November 12). Why is pop culture so obsessed with the middle ages? *The New Republic*. https://newrepublic.com/article/115572/cara-parks-reviews-nicola-griffiths-hild

The University of Mississippi. (n.d.). *Castle and Siege Terminology*. http://home.olemiss.edu/~tjray/medieval/castle.htm

Science Buddies. (2021). *Comparing the Engineering Design Process and the Scientific Method.* www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-compare-scientific-method

Shaw, C. (2001). Launceston Castle [Photograph]. *Wikipedia.* https://en.wikipedia.org/wiki/File:Launceston_Castle_-_geograph.org.uk_-_22242.jpg

Snyder, C. (2018). *What is LARPING?* Insider. www.businessinsider.com/what-is-LARPing-live-action-role-playing-steampunk-aurum-LARP-2018-10

Victoria, K. (2018, December 29). *What is Computational Thinking? Why thinking like a computer builds skills for success*. Teach Your Kids Code. https://teachyourkidscode.com/what-is-computational-thinking/

Widing, G. (2008). We Lost Our World and Made New Ones: Live Role-Playing in Modern Times. Playground Worlds Creating and Evaluating Experiences of Role-Playing Games, 262.

Scott R. Bartholomew, Ph.D., is an assistant professor of Technology and Engineering Studies at Brigham Young University, Provo, UT. He can be reached at scottbartholomew@byu.edu.

Nathan Pehrson is a recent graduate of Brigham Young University's Technology and Engineering Studies Program. He graduated with an emphasis in teaching, minored in Design Thinking, and will be starting his first year of teaching middle school with Weber School District in Utah. Nathan loves making things and seeing what others can create. He seeks to provide students with experiences that build confidence in their abilities to be designers and makers. He is a game design enthusiast and is always searching for the crossover between game design and teaching theory. He can be reached at nathanhpehrson@gmail.com

This is a refereed article.