107th Mississippi Valley Technology Teacher Education Conference
Nashville, Tennessee

A Quasi-Experimental Study Examining Secondary Students' Views of Physical Computing

Session I: Research in Technology Education

Tyler S. Love[1]
Rueben S. Asempapa[2]
Penn State Harrisburg
November 19, 2021

Abstract

In recent years there has been an increasing emphasis on providing access to computational thinking (CT) instruction for every K-12 student in the United States (U.S.). Concurrently, there has been an increase in the call for integrating CT concepts within STEM disciplines and standards documents. Specifically, computation, automation, artificial intelligence, and robotics has been identified as one of the eight technology and engineering context areas of the *Standards for Technological and Engineering Literacy*. However, the appearance of CT instruction in design and technology (D&T) courses varies drastically. One method that has been implemented in England and is becoming more popular in the U.S. is physical computing. This is an area of limited but growing research. This study utilized a quasi-experimental design to examine the physical computing attitudes of 170 middle school students who participated in a screen-based or physical computing unit. The results indicated that students who completed the screen-based unit reported statistically greater attitudes toward classroom applications and career/future use of physical computing. Students who participated in the physical computing unit believed that physical computing made it more difficult to learn CT concepts, but they preferred the hands-on aspect of physical computing. This study provides implications for improving physical computing instruction and the STEM contexts withing which it is taught.

Keywords: Computational Thinking, Physical Computing, Integrated STEM Education, Design and Technology, Technology and Engineering Education

Introduction

In recent years there has been an increasing emphasis on providing access to computer science (CS) education for every K-12 student in the United States (U.S.). One of the key concepts behind this movement is developing computational thinking (CT) skills in all students because of its applicability beyond computer science (Wing, 2006; Yadav et al., 2014). CT has been described as a fundamental problem-solving skill that can benefit students in solving abstract problems and understanding human behavior (Wing, 2006). The applicability of CT across science, technology, engineering and mathematics (STEM) disciplines is evident from its inclusion as one of the eight technology and engineering contexts within the *Standards for Technological and Engineering Literacy* (ITEEA, 2020) and one of the eight science and engineering practices in the *Next Generation Science Standards* (NGSS) (NGSS Lead States,

[1]Penn State Harrisburg, Assistant Professor of Elementary/Middle Grades STEM Education, TSL48@psu.edu
[2]Penn State Harrisburg, Assistant Professor of Mathematics Education, RSA26@psu.edu

2013). Furthermore, professional associations such as the International Technology and Engineering Educators Association (ITEEA) and National Science Teaching Association (NSTA) have advocated for the meaningful integration of computational thinking across STEM content areas (Asante et al., 2021; ITEEA, 2021).

While CT, technological and engineering literacy, and science and engineering practices all play a valuable role in preparing students to solve interdisciplinary STEM challenges, the best methods for integrating these skills to provide the greatest student learning benefits remains unclear. It has been proposed that physical computing can provide a more engaging and authentic integrated STEM learning experience in comparison to siloed coding instruction or solely virtual CT activities (Love & Bhatty, 2019; Love & Griess, 2020; Love & Rajyaguru, accepted; Love & Strimel, 2017; Love et al., 2016). Physical computing involves the programming of interactive physical systems or devices via software to teach students CT skills through physical tools and hands-on activity (Cápay & Klimová, 2019; Genota, 2019). It is rooted in constructivist learning theory and the work of Seymour Papert. As Romeike and Przybylla (2014) explained, “With physical computing, constructionist learning and typical processes of computer science education can be brought together in a creative and practical way” (p. 242). Anecdotal observations of physical computing in K–12 have indicated that students enjoyed the hands-on problem-solving nature of physical computing more than screen-based activities (Love & Bhatty, 2019; Love & Griess, 2020; Love et al., 2016). Despite these observations there remains a lack of empirical research examining K-12 the benefits of physical computing for students compared to traditional screen-based CT instructional methods.

Literature Review

The call for integrating computing and design and technology (D&T) within K-12 contexts is not a novel concept. England had advocated for this integration with the inclusion of physical computing concepts in their national curricula before the computer science for all initiative gained momentum in the U.S. The Royal Academy of Engineering in England has provided guiding documents to help educators apply computing within D&T to program and control physical systems (Royal Academy of Engineering, 2014). The Royal Academy of Engineering proposes that teaching students to program and control physical D&T systems they designed themselves is a highly motivating and tangible experience that reflects relevant and practical contexts which students will encounter in the world outside of school.

More recently, the U.S. has placed greater emphasis on computer science and STEM literacy for all in K-12 to address the critical shortage of graduates prepared to enter the workforce in these fields. This was reflected in the development of computer science standards in 2017 (CSTA, 2017). While designing and controlling electronic systems was encompassed within the designed world standards of the *Standards for Technological Literacy* (ITEA/ITEEA, 2000/2002/2007), the more recently released *Standards for Technological and Engineering Literacy* (STEL) (ITEEA, 2020) specifically identified computation, automation, artificial intelligence, and robotics as one of the eight technology and engineering context areas. Various applications of computing within D&T contexts (e.g., physical computing) are highlighted within the aforementioned STEL context area. Similar to the Royal Academy of Engineering (2014) document, the STEL provides examples of authentic, design-based computational learning experiences to encourage the integration of core D&T concepts with this context area.

Examples and Benefits of Physical Computing in K-12

Aside from the examples in the STEL, a number of other resources have provided quality exemplars of K-12 physical computing lessons that integrate an array of STEM concepts. Some educators have focused on integrating science through physical computing to teach concepts related to the function of a four-chambered heart (TI, 2017a), irrigation systems (TI, 2017b), automated farming (Simpson, 2017), and smart greenhouses (Asante et al., 2021). From a D&T lens, a number of educators have demonstrated how physical computing can align with technology and engineering context area topics such as smart home devices (Love et al., 2016), autonomous vehicles (Love & Bhatty, 2019), micro electric vehicles (Bartholomew et al., 2020), e-textiles (Strimel et al., 2019), and the engineering design process (Love & Griess, 2020). These are just a few examples highlighted within science, technology, and engineering education contexts. For a variety of reasons, computational thinking concepts have often been taught via screen-based programs or digital apps. Code.org (2021) recently integrated a physical computing unit into its Computer Science Discoveries curriculum. These various examples signify that educators may believe there are benefits to implementing physical computing to teach computational thinking and engineering design skills. What the impact is compared to solely screen-based computing instructional strategies has not been explored in great detail.

Prior Research on Physical Computing in K-12

Przybylla et al. (2017) found programmable microcontrollers provide an attractive and promising approach to teaching physical computing. Teachers in their study expressed they were most interested in physical computing from a pedagogical standpoint to promote motivation, direct feedback, and tangibility of computational thinking concepts. The teachers also expressed the main issues that prevented them from implementing physical computing activities were factors such as limited time, technical difficulties with the hardware, and not enough training or instructional resources. Interestingly, the teachers did not report collaborating with other educators often to teach physical computing. The most frequent collaboration they reported was with art and physics teachers to outsource the fabrication of physical computing projects.

In regard to students, Hodges et al.'s (2020) research found that physical computing could provide more positive experiences than screen-based experiences due to the support for open ideation. They also discovered that students appreciated building real, tangible devices and that physical computing platforms stimulated students' creativity. Additionally, they found female students to be more engaged through physical computing activities than screen-based experiences. In Sentence and Schwiderski-Grosche's (2012) study, secondary students reported that they enjoyed the challenge and freedom that physical computing provided along with the tactile coding experience beyond a computer screen; however, they also indicated that physical computing was difficult. The researchers noted that physical computing appeared to increase female students' confidence and interest in learning to code.

Attitudes Toward Physical Computing

A number of attitudinal scales have been found to provide valid and reliable insight regarding the impact of computer science related educational interventions (Boulden et al., 2021; Hoegh & Moskal, 2009; Yadav et al., 2014). Love and Rajyaguru's (accepted) research explored changes in secondary students' attitudes toward physical computing after participating in a four-week physical computing unit that utilized the Crumble (a microcontroller that can be programmed using drag and drop block-based coding to control external electronic sensors).

Students' reported statistically significant increases in five computational thinking attitude constructs (definition, comfort, interest, classroom applications, and career/future use). More specifically, they found that students' attitudes toward the coding and engineering items both had significantly increased as a result of participating in the physical computing unit. Love and Rajyaguru concluded that participation in a physical computing unit has the potential to increase students' attitude toward physical computing.

Purpose of the Study

Although the literature indicated that physical computing can have positive benefits, there were also concerns raised about classroom implementation and the level of difficulty expressed by students. Furthermore, there was a limited amount of research investigating the impact that physical computing can have in comparison to traditional screen-based methods when controlling for specific factors (e.g., prior engineering and coding experiences, gender). This study sought to address that gap and was guided by the following research questions:

Research Question 1 (RQ1): Is there a statistically significant difference *between the control (screen-based) and experimental (physical computing) groups* regarding students' attitude toward physical computing?

Research Question 2 (RQ2): Is there a statistically significant difference between the control and experimental groups' attitudes toward *engineering and coding items*?

Research Question 3 (RQ3): Is there a statistically significant difference between the control and experimental groups regarding students' attitude toward physical computing *after controlling for prior engineering course experiences*?

Research Question 4 (RQ4): Is there a statistically significant difference between the control and experimental groups regarding students' attitude toward physical computing *after controlling for prior computer science course experiences*?

Research Question 5 (RQ5): Is there a statistically significant difference between students' attitude toward physical computing when examining the linear combination of *control and experimental groups and participants' gender*?

Research Question 6 (RQ6): Is there an identifiable difference between the control and experimental groups regarding students' *learning preference for future coding lessons*?

Methodology

A quasi-experimental design was employed to administer the survey to 170 seventh grade (12-13 years old) students in a suburban secondary school within the U.S. The students participated in either the control group (completed a Scratch game design screen-based unit) or the experimental group (completed a physical computing unit using the Crumble microcontroller). The Scratch game design unit was taught by an educator who had been teaching computer applications and coding courses at the school for seven years and they did not receive any training on physical computing. Conversely, the Crumble unit was taught by a design and technology (D&T) educator who was in their sixth year of teaching D&T at the school and who had completed undergraduate coursework in electronics and design-based pedagogy. One of the researchers who had experience leading numerous physical computing workshops at state and national conferences provided one full day of professional development (PD) for the D&T teacher only. The PD covered methods for integrating the Crumble microcontroller within integrated STEM design challenges. Based on success in prior studies, the Crumble was deemed the best device for implementing physical computing design challenges at this age level due to its low cost, easy usability, durability, and readily available instructional resources (Love & Bhatty, 2019; Love & Griess, 2020; Plaza et al., 2018).

After the one day PD session, the researcher assisted the D&T teacher in developing an authentic physical computing design challenge. The challenge required students to utilize the Crumble to control the physical prototype they created through the engineering design process. Students were tasked with solving the collision avoidance design challenge (Love & Bhatty, 2019). They had to design a mini vehicle out of cardboard, hardboard, or 3D printed material to custom fit the Crumble and various sensors. Their vehicle had to include LED lights which changed color when driving versus stopped, forward/reverse wheel motors, and a distance sensor that stopped the car to prevent collisions. Some students integrated the line follower sensor and created an autonomous vehicle course. Other students added wire tethered push buttons to control the steering and direction of the vehicle. The design challenge allowed freedom for students to customize their vehicle as needed and add additional safety features they believed were important.

The collision avoidance design challenge was implemented during a four-week physical computing unit taught to three middle school D&T class sections (n = 72). First, students were introduced to the Crumble and tasked with completing a series of Crumble design challenges to practice creating electronic circuits using various sensors and programing them using the Crumble software. Examples of these design challenges included programming LEDs to operate like a traffic signal, a burglar alarm that used the distance sensor and a buzzer, and others developed by Redfern Electronics (2021) who produces the Crumble. In the second phase of the unit, students were tasked with designing the physical prototype of their vehicle under the constraints that it must be able to go forward and backward, and securely house the Crumble plus any sensors. Students researched autonomous vehicle features and leading causes of accidents in the U.S. to design what safety features and sensors they wanted to integrate into their vehicle design (Love & Bhatty, 2019). Students then created their vehicle prototypes using various materials and tools, tested them out, and made any necessary adjustments to optimize the function of their vehicle. During the third and final phase of the unit, students were required to draw schematics of their Crumble and sensors, as well as designs for how the sensor system would be integrated into their vehicle. The instructor then helped students troubleshoot their design as they assembled their circuit and coded their Crumble.

The control group consisted of four sections of students (n = 98) at the same grade level and in the same school as the experimental group. They were led through three weeks of a screen-based Scratch game design unit. Neither instructor had honors courses and students were randomly placed in each class by the school counselors. When this study was conducted students were in their first rotation of courses for the year. Students in the D&T course did not have the computer science course that academic year, and vice versa. This allowed the researchers to examine if there were differences among the two groups of students. To examine potential differences, both the experimental and control groups completed a Likert-scale survey after their respective three-week physical computing or Scratch screen-based unit. Responses from both groups were collected online via survey software and analyzed in the SPSS 27 statistical software package.

Instrumentation

The instrument utilized in this study was based off of the work of Hoegh and Moskal (2009). They convened a panel of assessment and computer science education experts to develop a questionnaire that could measure students' attitude toward computing based on five key constructs from the literature: definition, comfort, interest, classroom applications, and career/future use. The instrument was found to be reliable and valid through Cronbach's alpha tests and a factor analysis (Hoegh & Moskal, 2009). Yadav et al. (2014) later built upon Hoegh and Moskal's work to develop the Computing Attitude Questionnaire (CAQ). The CAQ utilized the same constructs identified by the expert panel in Hoegh and Moskal's study and consisted of 21 Likert scale items. Yadav et al. tested the instrument among a sample of 377 preservice teachers and established acceptable reliability through Cronbach's alpha tests. Leonard et al. (2018) later used this instrument to examine attitudes toward and the understanding of computational thinking among teachers who participated in robotics and game design treatment groups.

For this study, we made slight modifications to the items from Yadav et al.'s (2014) instrument while maintaining the same number of items with similar wording and in the original five key constructs. The slight modifications made the items more readable for 12-13 year old students and allowed the researchers to not only examine students' attitude toward coding concepts, but also their attitude toward engineering design concepts. As described in the literature review, the unique blend of coding and engineering design is a key characteristic of physical computing. These slight modifications included replacing the terms "computational thinking", "computing", and "computer science" with the term "coding" as it was more familiar to students at this age. Those 21 items were then duplicated and the term coding was replaced with "engineering." An example of two items from the comfort construct are: "I use coding skills in my daily life" and "I use engineering skills in my daily life." This resulted in the "Physical Computing Attitude Survey (PCAS)", an instrument that allowed for measuring students' attitudes towards coding and engineering.

To better understand what the instrument was measuring, the five constructs must first be examined. The definition construct examined students' understanding of the definition of coding or engineering. Comfort measured participants' comfort level with coding or engineering, while the interest items gauged their level of interest in coding or engineering. The classroom applications items gauged students' attitude about the integration and learning of coding or engineering concepts in their courses. Lastly, the career/future use construct measured students' attitude regarding the influence that coding or engineering will have on their future academic and career choices. Due to the slight modifications to the instrument items, Cronbach's alpha tests were conducted to determine the reliability of the items. The PCAS (α =.908) items revealed high internal reliability. Additional Cronbach's alpha tests revealed the coding (α =.794) and engineering (α =.822) items demonstrated strong or acceptable internal reliability measures. Based on these strong internal reliability measures the PCAS was deemed a feasible instrument to examine students' attitude toward physical computing.

Participants

There were 170 total students who participated in this study, 98 from the control group and 72 from the experimental group. The two groups had an identical percentage of male and female students. The majority of students in both groups identified as White. In comparison to the control group, the experimental group had a higher percentage of students who had taken an engineering course prior to this study. Both groups had a large percentage of students who had reported taking a coding course before participating in the survey.

Table 1

Participant Demographics

Characteristic	Control n (%)	Experimental n (%)
Gender		
Male	46 (47)	34 (47)
Female	52 (53)	38 (53)
Race		
White	73 (75)	57 (79)
Black	10 (10)	4 (6)
Hispanic	2 (2)	3 (4)
Asian	0 (0)	3 (4)
Native American	0 (0)	1 (1)
Middle Eastern	1 (1)	4 (6)
Latin American	2 (2)	0 (0)
Multiple Races	10 (10)	0 (0)
Prior Coursework Completed		
Engineering	41 (42)	47 (65)
Coding	83 (85)	64 (89)

Note. Control group n = 98; experimental group n = 72

Data Analyses

The survey responses were organized using the SPSS version 27 statistical software and analyses were performed in different steps. First, descriptive statistics and tests for assumptions were conducted to analyze the overall item response and identify any possible trends or anomalies in the responses. Next, various categories from the survey items were analyzed using multivariate analysis of variance (MANOVA), multivariate analysis of covariance (MANCOVA), and multinomial logistic regression with the intervention condition (control vs. experimental) and other variables in the data set. Specifically, the participants' responses to survey items within each category were aggregated and then averaged based upon the number of items in each category. Reverse coding of the negatively worded items was implemented as specified by Yadav et al., (2014). The average scores from all the MANOVA, MANCOVA, and multinomial logistic regression analyses were utilized. We used the Wilk's Lambda value in the MANOVA and MANCOVA analyzes because the aggregate or the overall (multivariate tests) was comprehensive and suitable for a different number of respondents for each category involving independent variable(s) (Grice & Iwasaki, 2007). Meanwhile, the results of the data analysis were also carried out separately (tests of between-subjects effects) as the researchers wanted to get more accurate results by using Bonferroni alpha value. The use of Bonferroni can control the problem of Type 1 error, which often occurs in a study (An et al., 2013; Grice & Iwasaki, 2007). All analyses were considered statistically significant at the $p < .05$ level.

Findings

Attitude Toward Physical Computing (RQ1)

In answering RQ1, a one-way (5 x 1) between-subjects multivariate analysis (MANOVA) was performed to determine whether there were any differences between students' attitude toward physical computing. The dependent variables on students' attitude toward physical computing were the instrument constructs (definition, comfort, interest, classroom applications, and career/future use). The intervention served as the independent variable on two levels (control group and experimental group). There were satisfactory results after evaluating the assumptions of normality, homogeneity of variance-covariance matrices [the Box's M of 21.22 indicates that the homogeneity of covariance matrices across groups was assumed [$F(15, 93643) = 1.37$, $p = .15$], linearity, and multicollinearity. Using Wilk's Lambda as the criterion, the multivariate effect was statistically significant by intervention levels $F(5, 164) = 3.56$, $p < .05$, partial $\eta 2 = .10$. The univariate tests showed that there were statistically significant differences across the intervention levels on the classroom applications, $F(1, 168) = 4.20$, $p < .05$, partial $\eta 2 = .02$ and career/future constructs, $F(1, 168) = 5.44$, $p < .05$, partial $\eta 2 = .03$. The Bonferroni post hoc tests suggested that students in the control group of the intervention ($M = 7.33$, $SD = 0.13$) had greater attitudinal scores toward physical computing within the classroom applications construct. Similarly, the post hoc tests using Bonferroni suggested that students in the control group ($M = 24.47$, $SD = 0.49$) had greater attitudinal scores toward physical computing within the career/future use construct.

Attitudes Toward Engineering and Coding Items (RQ2)

In order to answer RQ2, a one-way MANOVA was used to determine whether there was an identifiable difference in students' attitudes toward engineering or coding items based on their group (control or experimental). The dependent variables were the engineering item scores and the coding item scores. The intervention group served as the independent variable on two levels (control group and experimental group). A non-significant Box's M test ($p = .32$) indicated homogeneity of covariance matrices of the dependent variables across the levels of intervention group. The multivariate effect was not statistically significant by intervention group levels, $F(2, 167) = .78$, $p = .47$, partial $\eta 2 = .01$. So, we fail to reject the null hypothesis and concluded that there was not a statistically significant difference between control and experimental group participants' attitudes toward engineering or coding items.

Controlling for Prior Engineering Coursework (RQ3)

The objective of RQ3 was to determine whether there were differences between students' attitudes towards the five instrument constructs after controlling for the effect of prior engineering courses. This was achieved by performing a multivariate analysis of covariance (MANCOVA). The dependent variables were definition, comfort, interest, classroom applications, and career/future use. The independent variable was physical computing at two levels (experimental group and control group), and the covariate was students' prior engineering course completion. Results of evaluating assumptions of normality, homogeneity of variance-covariance matrices [the Box's M of 21.22 indicates that the homogeneity of covariance matrices across groups is assumed [$F(15, 93643) = 1.37$, $p = .15$], linearity, and multicollinearity were satisfactory. With the use of Wilk's Lambda criterion, the multivariate effect was statistically significant $F(5, 163) = 3.91$, $p < .05$, partial $\eta 2 = .11$. The univariate tests showed that there were statistically significant differences across the physical computing attitude for the constructs of classroom applications $F(1, 167) = 6.51$, $p < .05$, partial $\eta 2 = .04$ and career/future use $F(7, 161) = 9.65$, $p < .05$, partial $\eta 2 = .06$. When compared to the experimental group, students in the control group showed significantly higher means in their attitudes toward the physical computing classroom applications and career/future use constructs. The Bonferroni post hoc tests suggested that students in the control group ($M = 7.33$, $SD = 0.13$) had greater attitudes toward the classroom applications of physical computing after controlling for their prior engineering course experience. Similarly, the post hoc tests using Bonferroni suggested that students in the control group ($M = 24.47$, $SD = 0.49$) had greater attitudes toward their career/future use of physical computing when controlling for prior engineering course experience.

Controlling for Prior Computer Science Coursework (RQ4)

In investigating RQ4, the dependent variables were the instrument constructs of definition, comfort, interest, classroom applications, and career/future use. The independent variable was physical computing at two levels (experimental group and control group), and the covariate was students' prior coding course experiences. Evaluation of the homogeneity of variance-covariance matrices and normality assumptions underlying MANCOVA did not reveal any substantial anomalies, and the a priori level of significance was set at .05. The multivariate analysis of covariance (MANCOVA) was performed to investigate differences in students' attitudes related to the five instrument constructs after controlling for the effect of students' prior coding course experiences. The multivariate result from the MANCOVA was statistically significant $F(5, 163) = 3.53$, $p < .05$, partial $\eta 2 = .10$. according to Wilk's Lambda criterion. The univariate tests showed that there were statistically significant differences across the two groups in regard to classroom applications, $F(1, 167) = 4.40$, $p < .05$, partial $\eta 2 = .03$ and career/future use, $F(1, 167) = 5.93$, $p < .05$, partial $\eta 2 = .03$. The Bonferroni post hoc tests suggested that students in the control group ($M = 7.33$, $SD = 0.13$) had greater attitudinal scores toward the classroom applications of physical computing after controlling for their prior coding course experience. Similarly, the post hoc tests using Bonferroni suggested that students in the control group ($M = 24.50$, $SD = 0.49$) had greater attitudinal scores toward the career/future use construct of physical computing after controlling for their prior coding course experience.

Differences Between Male and Female Students (RQ5)

The purpose of RQ5 was to understand if there was an interaction between the two independent variables (participant groups and gender) on the two dependent variables (coding items and engineering items). Again, evaluation of the homogeneity of variance-covariance matrices and normality assumptions underlying MANOVA did not reveal any substantial anomalies, and the apriori level of significance was set at .05. Using Wilk's Lambda as a criterion, the multivariate results from the MANOVA for the combined dependent variables was statistically significant $F(2, 165) = 4.85$, $p < .05$, partial $\eta 2 = .06$ for gender. However, the MANOVA was not statistically significant for the participant groups (control and experimental) $F(2, 165) = 0.65$, $p = .53$, partial $\eta 2 = .01$. No statistically significant interaction was found, $F(2, 165) = 2.87$, $p = .06$, partial $\eta 2 = .03$. To investigate the impact of the effects on the individual dependent variables, a univariate *F*-test using an alpha level of .05 was performed. The main effect of gender was statistically significant on only engineering, $F(1, 166) = 6.94$, $p < .05$, partial $\eta 2 = .04$. The pair-wise comparison followed by a univariate *F*-test indicated that the statistically significant difference was found between male and female students in engineering.

The Bonferroni test suggested that male students (M = 40.71, SD = 0.71) had higher attitudinal scores toward the engineering items than female students (M = 38.13, SD = 0.67).

Preference for Learning Computer Science (RQ6):

The surveys asked supplemental questions about students' preferences for learning coding concepts. The control group was asked if they believed that learning coding would be easier or more difficult with the integration of hands-on engineering processes/materials and coding/electronic components to automate a product they built. They were also asked if in future lessons they would prefer to learn coding in conjunction with physical hands-on engineering design activities, or if they would prefer to learn coding in a screen-based format. The same questions were posed to the experimental group but in relation to the Crumble since they had participated in the intervention (ex. Do you think the hands-on engineering aspect of the Crumble made learning coding easier or more difficult than only using a computer?). A higher percentage of students in the control group (74%) believed physical computing would make learning coding a little or much easier in comparison to the experimental group (60%). However, a higher percentage of students in the experimental group indicated they would prefer to learn coding through physical computing as opposed to screen-based methods (Table 2).

Table 2

Supplemental survey question about preferred methods to learn coding concepts

Response	Control n (%)	Experimental n (%)
Hands-on physical computing format?		
Much easier	21 (21)	10 (14)
A little easier	52 (53)	33 (46)
No difference	20 (20)	19 (26)
More difficult	4 (4)	8 (11)
Much more difficult	1 (1)	2 (3)
Preferred learning format?		
Physical computing	70 (71)	56 (78)
Computers only	28 (29)	16 (22)

Note. Control group n = 98; experimental group n = 72

Discussion

The findings indicated students who participated in the screen-based unit had greater attitudes toward physical computing in the constructs of classroom applications and career/future use. This was true for analyses examining all instrument items, when controlling for prior coding course experience, and engineering course experience. This contradicts the researchers' hypothesis that students who participated in the physical computing intervention group would report greater attitudinal scores toward physical computing. Given the background of the instructors and the nature of the D&T course in which the physical computing unit was taught, the researchers also hypothesized students in the experimental group would have reported greater attitudinal scores toward the engineering items. When examining the open ended responses provided by students in the experimental group, they reflect similar findings to that of students in Sentence and Schwiderski-Grosche's (2012) study. In both studies, middle school students expressed the were intrigued by the open-ended and tactile characteristics of physical computing, but they found it much more challenging than screen-based learning. Their view that physical computing was more challenging or frustrating could have caused them to view it as not highly applicable in a course. In comparison, students in the screen-based group may not have experienced as many challenges that are inherent with physical computing activities (e.g., electronic components, prototyping, and software). These challenges reported by students could have also influenced their attitude toward their future use of items that they struggled to get to operate correctly. The context of the design challenge could have also had an impact on career/future use as opposed to students being able to identify their own problem to solve that may have had greater future implications in their mind.

The design challenge in the experimental group may have also not been as appealing to female students in comparison to the Scratch game design that students in the control group completed. Sentence and Schwiderski-Grosche's (2012) study indicated that because of the open-ended nature of physical computing, it appeared to increase female students' interest and confidence in coding. In this study, the higher attitude scores toward engineering by male students may be reflective of the collision avoidance design challenge. Providing a more gender neutral or open-ended design challenge, such as some of the examples cited in the literature review (e.g., e-textiles) may been more appealing and engaging to female students.

Lastly, the supplemental questions revealed some interesting results from students in regard to their preference for future coding lessons. Compared to the experimental group (60%), a higher percentage of students in the control group (74%) believed it would be easier to learn coding through the integration of hands-on engineering design activities (physical computing). However, when asked how they would prefer to learn coding concepts in future courses, the majority of students in both the control (71%) and experimental (78%) groups indicated they would prefer learning coding integrated within hands-on engineering design challenges. The supplemental open ended responses from students in the experimental group provided insight about these interesting results. Many students in the experimental group mentioned that the Crumble and sensors made the challenge more complex. Troubleshooting the electronic components, developing a physical prototype, and programming the Crumble each added additional challenges for students to solve. For example, students provided responses such as, "I liked that we could design different light patterns using things we'd learned in computer class. I didn't like how difficult some of the design challenges were" or "It was cool when you completed a challenge but it was difficult." Despite these responses about the challenges, some of the students indicated they liked having the hands-on aspect as opposed to just coding in the Crumble software, "I liked how you could actually see what the code is doing in real life." These findings are consistent with Sentence and Schwiderski-Grosche's (2012) study.

Limitations

There are a few limitations of this study that must be considered. This study may not be generalizable as it involved students from one suburban middle school. Additionally, only a posttest was conducted for each group after they completed their respective unit. Utilizing a pretest and posttest design for both groups could be beneficial in future studies to examine gains as a result of the intervention. Furthermore, the influence of the teachers is unknown. While PD was provided to the experimental group instructor, he had limited coding teaching experience in comparison to the control group instructor. Continued resources, support, and PD for inservice teachers could enhance CT instruction.

Conclusions

The results of this study indicate that although many students found the idea of hands-on physical computing intriguing, those who participated in the physical computing unit believed it was more challenging than simply screen-based coding activities. The potential for increased chance of failure and troubleshooting associated with physical computing was not appealing to students. These troubleshooting skills that integrate multiple STEM concepts to design innovative systems are important for solving authentic engineering design challenges. This learning by failure provides arguably a more valuable learning experience for students. Although not popular among the students, learning to overcome challenges and failure could provide beneficial life skills

This study provides important implications for further examining how physical computing is taught and how it can be improved to teach important skills that students may not obtain through screen-based lessons. Furthermore, the study provides insight for STEM and computer science educators considering implementing physical computing units in their courses. The findings provide substantial implications for D&T educators who may find their programs competing with computer science programs as elective courses and are seeking unique ways to integrate authentic coding and engineering experiences that appeal to all students. Physical computing can draw upon D&T educators' expertise in electronics concepts, the engineering design process, and coding to provide a unique standards-based interdisciplinary learning experience. This study calls for further research regarding physical computing instruction and student learning to enhance their views toward classroom applications and career/future use of physical computing skills.

Recommendations

For Researchers

Despite the instrument items demonstrating strong reliability measures, the items should be examined for clarity and specificity before using in future physical computing studies. In future studies it is recommended that a pretest and posttest comparison be performed to compare gains between groups. Additionally, longer training and additional support should be provided for instructors who are new to physical computing and implementing it as part of future studies.

For Practitioners

As indicated by the literature, school districts should seek to provide sufficient PD and resources to support teachers in implementing physical computing. This includes materials, instructional resources, instructional time, and collaborative planning opportunities. It is highly recommended that teachers who are interested in implementing physical computing practice conducting the lesson activities and provide a wealth of supplemental resources for students as they indicated that they find physical computing interesting but also more challenging and frustrating. Teachers should ensure the physical computing design challenges are open ended and realistic to appeal to all students.

References

An, Q., Xu, D., dan Brooks, G. P. (2013). Type I error rates and power of multiple hypothesis testing procedures in factorial ANOVA. *Multiple Linear Regression Viewpoints*, *39*(2), 1-16.

Asante, C. K., Semerjian, A., Xu, P., Jackson, D., Cheng, Y., Chasen, A., Shah, A., Brett, J., & Broadstone, M. (2021). An integrated STEM and computing curriculum for the human-technology frontier. *Connected Science Learning*, *3*(2). https://www.nsta.org/connected-science-learning/connected-science-learning-march-april-2021/integrated-stem-and

Bartholomew, S. R. & Mentzer, N. (2020). Data collection and visualization via $16 Micro:Bit. Presented at the International Technology and Engineering Educators Association STEM Showcase, Baltimore, MD. https://www.iteea.org/File.aspx?id=172215&v=89fb870a

Boulden, D. C., Rachmatullah, A., Oliver, K. M., & Wiebe, E. (2021). Measuring in-service teacher self-efficacy for teaching computational thinking: Development and validation of the T-STEM CT. *Education and Information Technologies*, *26*(4), 4663-4689. https://doi.org/10.1007/s10639-021-10487-2

Cápay, M., & Klimová, N. (2019). *Engage your students via physical computing!* Proceedings of the Annual IEEE Global Engineering Education Conference, Dubai, UAE, pp. 1216–1223. doi: 10.1109/EDUCON.2019.8725101

Code.org. (2021). Computer science discoveries (‘21-‘22) https://studio.code.org/courses/csd-2021

Computer Science Teachers Association (CSTA). (2017). K-12 computer science standards.

Genota, L. (2019, January 23). ‘Physical computing’ connects computer science with hands-on learning. *Education Week*. https://www.edweek.org/teaching-learning/physical-computing-connects-computer-science-with-hands-on-learning/2019/01

Grice, J. W., dan Iwasaki, M. (2007). A truly multivariate approach to MANOVA. *Applied Multivariate Research*, *12*(3), 199-226.

Hodges, S., Sentance, S., Finney, J., & Ball, T. (2020). Physical computing: A key element of modern computer science education. *Computer*, *53*(4), 20-30.

Hoegh, A., & Moskal, B. M. (2009). *Examining science and engineering students' attitudes toward computer science*. Proceeding from the 39th IEEE Frontiers in Education Conference, 1-6. https://doi.org/10.1109/FIE.2009.5350836

International Technology Education Association (ITEA/ITEEA). (2000/2002/2007). *Standards for technological literacy: Content for the study of technology.* Author.

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

International Technology and Engineering Educators Association (ITEEA). (2021). Computational thinking. https://www.iteea.org/Resources1507/ComputationalThinking.aspx

Love, T. S., & Bhatty, A. (2019). The crumble: Integrating computer science through engineering design. *Technology and Engineering Teacher, 79*(2), 16-22.

Love, T. S., & Griess, C. J. (2020). Rosie revere's orangutan dilemma: Integrating computational thinking through engineering practices. Science and Children, 58(2), 70-76.

Love, T. S., & Rajyaguru, J. (accepted). *Integrating computational thinking and engineering practices to teach STEM: Examining students' attitudes about physical computing.* Paper presented at the 95th Annual Meeting of the National Association for Research and Science Teaching, Vancouver, Canada.

Love, T. S., & Strimel, G. (2017). Computer science and technology and engineering education: A content analysis of standards and curricular resources. *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.

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

Plaza, P., Carro, G., Blazquez, M., Sancristobal, E., Castro, M., García-Loro, F., & Muñoz, J. (2018). *Crumble as an educational tool to introduce robotics.* Proceeding from the XIII Technologies Applied to Electronics Teaching Conference, 1-7. https://doi.org/10.1109/TAEE.2018.8476054

Przybylla, M., Henning, F., Schreiber, C., & Romeike, R. (2017). Teachers' expectations and experience in physical computing. (pp. 49-61). Springer International Publishing. https://doi.org/10.1007/978-3-319-71483-7_5

Redfern Electronics. (2021). Guide to using crumbs. https://redfernelectronics.co.uk/getting-started/guide-to-using-crumbs/

Romeike, R., & Przybylla, M. (2014). Physical computing and its scope – Towards a constructionist computer science curriculum with physical computing. *Informatics in Education*, *13*(2), 241-254. https://doi.org/10.15388/infedu.2014.05

Royal Academy of Engineering. (2014). *Applying computing in D&T at K2 and S3: The national curriculum requirements.* https://www.data.org.uk/resource-shop/applying-computing-in-dt-at-ks2-and-ks3/

Sentance, S., & Schwiderski-Grosche, S. (2012). Challenge and creativity: using .NET Gadgeteer in schools. In M. Knobelsdorf & R. Romeike (Eds.), *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (pp. 90–100). New York: ACM.

Simpson, C. (2017). Answering food insecurity: Serving the community with food and knowledge using technology. *Purdue Journal of Service-Learning and Engagement, 4*(1), 45-49. https://doi.org/10.5703/1288284316528

Strimel, G. J., Morehouse, A., Bartholomew, S. R., Swift, C., & Woessner, J. (2019). Integrating computational thinking through wearable technologies and programmable e-textiles. *Technology and Engineering Teacher, 78*(8), 16-19.

Texas Instruments (TI). (2017a). Modeling the four-chamber heart for the TI nspire CX. https://resources.tistemprojects.com/tistemprojects-home/?resource_id=2175

Texas Instruments (TI). (2017b). Modeling the four-chamber heart for the TI nspire CX. https://resources.tistemprojects.com/tistemprojects-home/?resource_id=1802

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.

Yadav, A. , Mayfield, C. , Zhou, N. , Hambrusch, S. , & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1-16.