Jl. of Computers in Mathematics and Science Teaching (2020) **39**(2), 109-123

Implementing Socially Relevant Problem-Based Computer Science Curriculum at the Elementary Level: Students' Computer Science Knowledge and Teachers' Implementation Needs

THOMAS BRUSH, ANNE OTTENBREIT-LEFTWICH,
KYUNGBIN KWON, AND MICHAEL KARLIN
Indiana University, USA
tbrush@indiana.edu
aleftwic@indiana.edu
kwonkyu@indiana.edu
mike.karlin@gmail.com

The focus of this research project was to examine how problem-based learning (PBL) impacts students' interest and knowledge in computer science (CS) at the elementary level. By focusing on a problem that emphasizes social activism, we hypothesized that PBL CS could increase interest for students. We employed an iterative design-based research approach to examine how the CS PBL curriculum impacted 6th grade students' understanding of and interest in CS, as well as the supports teachers need to implement the curriculum. Results suggest that students' understanding and interest in CS increased. In addition, the teacher reported needing more content PD support, revisions to curriculum to improve comprehension, and other resources.

Keywords: computer science, problem-based learning, elementary

INTRODUCTION

According to the U.S. Bureau of Labor Statistics (BLS) Employment Projections for 2010–2020, more than half of the anticipated STEM jobs that will be created in this country will be in computing (US BLS, 2018). Other projections have indicated that computing job will be the fastest type of STEM occupation with half a million new jobs between 2014 and 2024 (Fayer, Lacey, & Watson, 2017). This trend is evident regionally, as well as nationally. It is clear, however, that not all these of jobs will be computer scientist positions. In fact, Kaczmarczyk, Dopplick, and the Education Policy Committee (2014) found that 63% of current computing jobs were in industries outside of computer science (CS), ranging from agriculture to automobile manufacturing. In addition, scholars have indicated that jobs outside of CS will still require basic CS skills (Delyser, Mascio, & Finkel, 2016; SREB, 2016). Because the future of jobs in the U.S. will require workers with CS and computational thinking, many states are adding computer science to the K-12 curriculum (Stanton et al., 2017). For example, educational leaders in Indiana were among the first to develop K-8 standards (Stanton et al.). In 2017, Indiana passed a state bill that required all schools to address K-8 CS standards and offer one high school CS class each year by 2021 (Indiana General Assembly, 2017).

While legislators and education administrators have instituted these important changes at the statehouse, it is the responsibility of districts and teachers to must implement the policy in their classrooms. Many teachers, however, struggle with where to begin, since most teachers -- particularly elementary and middle schools teachers (Ottenbreit-Leftwich & Biggers, 2017) -- describe having a lack of confidence and knowledge about computer science (Margolis, Rhoo, & Goode, 2017; Yadav, Berges, Good, & Sands, 2016). Thus, there is a need to collaborate with teachers to develop and implement CS curriculum that is engaging and effective with their students, and that teachers feel comfortable implementing in their classrooms.

There are many existing curriculum choices for elementary computer science (CS) (e.g., Code.org, Tynker, Everyday Computing, etc.) that integrate games, robots, and unplugged activities. However, there has been limited research on examining the potential of this curriculum at the elementary level (e.g., Ozturk, Dooley, & Welch, 2018). In addition, there are also CS curriculum choices that utilize an inquiry-based approach at the high school level (e.g., Exploring Computer Science, Mobile CSP). Research studies suggest that CS in an inquiry-based approach does improve student engagement (Hoffman, Rosato, & Morelli, 2019), student achievement (Chen &

Yang, 2019; Hoffman et al., 2019), and student attitudes towards CS (Hoffman et al., 2019; Pollock et al., 2017).

Although there is at least one curriculum at the elementary level that uses an inquiry-based approach for computer science (e.g., Project Lead the Way's Launch program), there has not been any research on whether this approach is successful at the elementary level. We need more information on how inquiry-based strategies such as problem-based learning (PBL) can be used to learn CS at the elementary level. CS naturally presents student-centered PBL opportunities that can also contribute to social engagement (Goldweber et al., 2011). For example, 5th grade Latino/a students who were enrolled in the CSteach curriculum, which explicitly focused on social justice and social good, increased their desire to further pursue CS as well as the value they saw in CS (Denner, Martinez, & Lyon, 2015). Because CS requires problem-solving skills for broad issues, computational problem- solving is a core competency (Liu et al., 2011) and it is important for students to learn computing in a concrete and personal way (Cooper & Cunningham, 2010; Goode, Chapman, & Margolis, 2012).

Extensive research conducted over the past decade has demonstrated that PBL can enhance both student engagement and students' academic achievement with challenging content in K-12 settings (Brush & Saye, 2017). A number of meta-analyses focusing on the implementation of PBL in K-12 environments concluded that PBL instruction is more effective than traditional, teacher-centered instruction with regard to student achievement (e.g., Ravitz, 2009; Walker & Leary, 2009). Studies across different subject areas have shown that students in PBL instruction tend to be more engaged, have better academic performance (Wirkala & Kuhn, 2011) and impact a wide range of student abilities (Belland, Glazewski, & Richardson, 2011).

Research examining the use of PBL to teach CS has demonstrated mixed results. For example, Dwyer et al. (2013) implemented a project where 4th grade students integrated computational thinking into a physics project. However, 4th grade students struggled due to a lack of prior background knowledge (hence the need for introductory block-based programming). Some studies have shown that if CS projects are too open-ended, students may have difficulties completing them (Cilburn & Miller, 2008). Therefore, the incorporation of CS knowledge needs to be thoughtfully integrated.

The focus of this research project was to examine how PBL impacts students' CS interest and knowledge at the elementary level. By focusing on a problem that emphasizes social activism, we hypothesized that PBL CS could increase interest for students (Goldweber et al., 2013). We employed an iterative design-based research approach to examine two questions:

1. To what extent does a computer science student-centered (problem-based learning) curriculum impact 6th grade students' understanding of and interest in CS?
2. What supports do teachers need to implement a computer science student-centered (problem-based learning) curriculum?

DESCRIPTION OF THE CS PBL CURRICULUM

To develop the CS curriculum used in this study, we collaborated with 6th grade teachers at a rural elementary school in Indiana. Our collaborative curriculum design efforts led to the development an initial curriculum that teaches CS block-based coding in a shorter time frame than most other curricula and directly addresses most of the Indiana CS K-8 standards. In addition, we incorporated a PBL activity that provided students with the opportunity to utilize their newly-acquired CS skills to address a socially relevant problem.

We started with 10 hours of a Block-Based Computer Science Curriculum utilizing Scratch to directly target the 6th grade standards. Utilizing frameworks from Brennan and Resnick (2012), the CS Framework (2017), and the Indiana CS K-8 standards, our curriculum focused on CT concepts (Sequences, Loops, Event, Condition, Parallelism, Data, Operator) and CT practices (deciding topics, decomposing tasks, developing programs, demonstrating programs). The additional component of the curriculum focused on a student-centered PBL activity that incorporated a social impact focus problem. The curriculum design efforts were a collaboration between 6th grade teachers and researchers from a variety of disciplines (computer science, educational psychology, instructional design) at Indiana University. After students had learned some basic coding, we collaborated with four 6th grade teachers and their CS/STEM specialist to co-develop a PBL curriculum that would allow students to apply their new CS skills to a relevant problem. Our teacher partners decided to focus the PBL activity on the driving question: *How can we create a culture of kindness in our school?* The activity introduced concepts of kindness and heavily incorporating elements of the partner school district's recently adopted social-emotional curriculum. Students were scaffolded on how to conduct research on this topic, and provided with resources to support their research. Students used their new CS skills to develop an "app" that would create a culture of kindness in their schools (see Figure 1). Some created games, providing suggestions on ways to be kind, while others created a kindness tracker with words of encouragement.

Figure 1. Example of “kindness” app.

METHOD

Research Context and Participants

This study took place over the course of nine weeks and involved one 6th grade elementary science teacher and two of his classes in a rural STEM school in central Indiana. The school was classified as a Title I school with 56% free/reduced lunch rate. The overall student population of the school was 6.5% multiracial, 3.8% African-American, 7.1% Hispanic, and 82% white.

The teacher participating in the study had 14 years of teaching experience, but this was the first year he had been asked to integrate computer science instruction into his science classes. He supervised the school’s robotics club for the past four years, but had no formal CS professional development. The teacher also had limited experience integrating problem-based learning strategies into his instruction. Fifty-three students participated in the study from the teacher’s two classes. The science class met for 50 minutes each day.

Design

This study employed educational design research, which "...is a genre of research in which iterative development of solutions to practical and complex educational problems provides the setting for scientific inquiry" (McKenney & Reeves, 2014, p. 131). Innovative educational environments and activities may be simultaneously designed, implemented, and evaluated to produce scientific knowledge and suggest interventions for practice (Wang & Hannafin, 2005).

We examined teacher and student experiences through a heuristic case study methodology to discover new understandings of the intervention with the goal of re-thinking the design or approach (Merriam, 1997). This methodology allowed us to explore our curriculum intervention in an authentic setting through detailed data collection involving multiple data sources (Stake, 2005).

Data Sources

Both qualitative and quantitative data were collected for this study, and included the following data sources:

CS knowledge pre-test and post-test. The researchers and teacher collaboratively developed a 14-item multiple-choice assessment designed to measure students' knowledge of basic computer science and coding principles. These were developed based on CS concepts (Brennan & Resnick, 2012; CSTA Framework, 2017; Indiana Department of Education, 2016). Eight of the test items focused on basic computer science concepts (e.g., sequences, loops, event, condition), and six of the items focused on CS practices (e.g., coding principles, debugging). Parallel forms of the test were administered to students three occasions: prior to the beginning of the unit (pre-test), immediately after the completion of the block-based coding portion of the curriculum activities (post-test 1), and immediately after the completion of the PBL portion of the unit (post-test 2).

Observations and debriefings. Observations were conducted in each of the classrooms participating in the project to determine how the curriculum was implemented and how it could be revised. We also conducted debriefing sessions with the teacher after each class session to garner the perceptions of the teacher regarding issues that the students may have faced as the unit was being implemented, and what supports might be needed for teachers to enhance future implementations of the unit.

Teacher interviews. We conducted a pre- and post-interview with the teacher to identify the successes and challenges faced in implementing the curriculum, as well as investigate how the curriculum could be improved which supports were beneficial and what additional supports were needed.

Procedure

The primary activities for the initial and final days of the unit were the administration of the pre- or post-test. For the other class sessions, students completed specific activities to build their CS concepts and practices. During the final week of the unit, students developed culminating presentations in which they presented their "apps" that addressed the driving question for the unit (see Table 1 for an outline of the unit implementation).

Each day of the unit was observed by the researchers. Teacher post-unit interviews were conducted after the completion of the unit.

Table 1

Pilot CS curriculum week-by-week topics, activities, and Indiana CS standards addressed

Week 1: CS Introduction & Foundations	Activities	CS Standards
Students are introduced to the basic ideas of computer science, hardware, software, and computer components.	Videos and discussions on relationship with humans and machines. Unplugged activities on binary and communication.	**6-8.CD.1** **6-8.CD.4** **6-8.IC.2**
Weeks 2-4: Extending CS Knowledge	Activities	CS Standards
Students are introduced to Scratch and the functions of different Scratch blocks.	Students create at least 5 programs in Scratch (e.g., a dance party, a maze, a quiz game, and a variables game, and functions). Final project has students create their own program incorporating these ideas.	**6-8.DI.1** **6-8.CD.2** **6-8.PA.2** **6-8.PA.3** **6-8.NC.2**
Week 5: Contextualizing the Problem	Activities	CS Standards
Students introduced to the PBL problem of "How can we create a culture of kindness in our school?"	Videos and discussions on creating a culture of kindness. Students research how acts of kindness in their school and daily lives can contribute to a culture of kindness.	N/A

Weeks 6-7: Research and Design	Activities	CS Standards
Students design and develop an “APP” using Scratch to address the PBL problem.	Students spend 8 lessons researching, planning, designing, and developing their Scratch project. Multiple check-ins and scaffolds help support development.	**6-8.DI.1** **6-8.CD.2** **6-8.PA.2 & 3** **6-8.NC.2**
Week 8: Presenting	**Activities**	**CS Standards**
Students present their final Scratch projects.	Students present and share their final projects with their peers, teacher, other students, and external visitors.	**6-8.NC.1** **6-8.IC.1**

Data Analysis

A one-way analysis of variance was conducted on student pre-test and post-test results to determine if there were differences among student scores on the three administrations of the test. For the purpose of analysis, all test scores were converted from raw scores to percentages. Effect sizes for each comparison were calculated using partial eta squared (Cohen, 1973).Qualitative data sources (e.g., interviews, observations) were analyzed by researchers for trends and patterns related to the research question (Creswell, 2002).

RESULTS AND DISCUSSION

Student Understanding and Interest

A one-way ANOVA was conducted to compare the overall scores of tests (pre, post Scratch lessons, and post PBL) (see Table 1). There was a statistically significant effect of time on the scores $F(2, 143) = 11.19$, $p < .001$, p2=.14. Post hoc comparisons using the Bonferroni test indicated that the mean score of post-tests (M = 65.9, SD = 21.55; M = 66.0, SD = 19.25) significantly improved from the pretest (M = 48.7, SD = 17.71). No significant difference was found between the post-tests.

As the tests consisted of two types of questions: conceptual understanding (8 items) and debugging skills (6 items), ANOVAs were carried out for each type. Regarding the conceptual understanding items, there was a statistically significant difference among the tests $F(2, 143) = 8.98$, $p < .001$, p2=.11. Post hoc comparisons using the Bonferroni test indicated that the mean score of post-tests (M = 75.2, SD = 23.70; M = 75.2, SD = 19.99) significantly improved from the pretest (M = 57.8, SD = 22.77). No significant difference was found between the post-tests.

Regarding the debugging skills, there was a statistically significant difference among the tests $F(2, 143) = 6.44$, $p = .002$, p2=.08. Post hoc comparisons using the Bonferroni test indicated that the mean score of post-tests (M = 53.5, SD = 24.75; M = 53.8, SD = 26.28) significantly improved from the pretest (M = 37.5, SD = 20.24). No significant difference was found between the post-tests. Comparing the scores of post-tests reveals that students achieved considerably lower scores in debugging tasks (54% correct) than conceptual questions (75% correct) (see Table 2).

Table 2
Summary of test results

	Pretest	Post Scratch lessons	Post PBL
Overall	48.7 (17.71)	65.9 (21.55)	66.0 (19.25)
Concepts	57.8 (22.77)	75.2 (23.70)	75.2 (19.99)
Debugging	37.5 (20.24)	53.5 (24.75)	53.8 (26.28)

Note. Mean (SD)

Discussion of test results. Student test results suggest that participation in the unit had a positive effect on students' knowledge of CS principles and concepts. Overall, students performed significantly better on the post-test than they did on the pre-test, and their level of performance was sustained after completion of the PBL portion of the unit. While students did not perform better on the post-test after completion of the PBL activities, they maintained their performance level which suggests that their knowledge of CS (as measured by the test) and ability to apply CS concepts towards an authentic and socially-relevant problem was maintained throughout the unit.

In addition, during the post-unit interview, the teacher discussed his perceptions that the curriculum was highly engaging and students that students even asked to explore Scratch during their free time: "I had a student ask, hey, can we work on scratch? I said, sure. I bet there was probably a third of both classes that got on scratch and began to program a new game and start to kind of manipulate some of those blocks around. And so that's the biggest engagement, when you can get kids doing what we're doing, if they're doing it on their own, you've got them. You've got something."

These results suggest that a problem-based CS curriculum can be effectively implemented at the elementary level and can engage students in the use of CS principles to support real-world problems. While other CS curricular initiatives have been implemented at the secondary level, this research provides support for engaging students in higher level CS activities that involve designing, programming, and implementing more large-scale projects at the elementary level.

Teacher Support Needs

Based on our interviews with the teacher, we identified three teacher needs to implement a computer science student-centered (problem-based learning) curriculum: (1) professional development on content, (2) revisions to curriculum and materials, and (3) additional resources.

Content. Based on the teacher pre-interviews, he mentioned that he needed more resources and professional development on computer science content to feel comfortable in teaching the curriculum: "My concern is just my depth of knowledge with [computer science]. I know computers, but the thing is as you get older, technology passes you up...[I need] lessons that would allow me as a learner to learn before I then turn around and present it to the class." He mentioned that although we had provided him with the curriculum, which enabled him to understand the content, he would feel more confident if he had additional background knowledge through professional development opportunities: "Professional development [is a need] for me, I know where my background knowledge wasn't as strong as I'd like it."

Revisions to curriculum and materials. Overall, the teacher felt the curriculum was successful. However, there were elements that could be improved to yield better student engagement and understanding. For example, the teacher suggested that the unplugged activities led to better student understanding, and when functions were introduced, we did not provide an unplugged activity. As a result, the teacher mentioned that students seemed more confused and off-task as a result of the lack of understanding. During the PBL activities, some of the information was too broad and the teacher needed to provide more soft scaffolding to support students. For example, in an email, the teacher described how he chunked the lessons to better guide students during their app planning sessions: "With the first class we showed them the whole example packet, then allowed them time to work. It seemed like too much instruction and too much time to work/get off task. With the second class, we broke it into smaller chunks and it worked a lot better."

In addition, the teacher suggested that the PBL research lessons where students researched acts of kindness needed more support: "I think we needed a better bridge between the culture of kindness and categories of kindness and the activity/handout. My recommendation would be to model the first act of kindness by finding an article or video as a class and filling in the boxes together, focusing on how this supports a culture of kindness and how it spreads kindness to others." In addition, the sixth graders seemed to struggle with searching and key terms. For next time, the teacher suggested "including the list of terms to search or QR codes at the top of the handout instead of the list of websites which can be hard to type in for sixth graders."

Other curriculum suggestions dealt with time management (e.g., "it was a lot of information to get through in 45 minutes," "some slides could be combined or dropped," and "[we] wished the students had a little bit more time at the end to manipulate the game.") We will revisit the curriculum to create clearer 45-minute lessons and include unplugged activities before applying them to their own Scratch designs.

Additional resources. The teacher also mentioned that additional resources were necessary to support the lesson. The technology equipment provided some challenges. Since students had iPads, we had initial problems with Scratch (not being able to save their work before Scratch 3.0 was released) and keyboards (arrows did not work with Scratch), and not being able to collaborate on Scratch (each group had to use one account to build their app, it could not be shared or copy/pasted). The teacher described that laptops could have made things easier for students: "I think that iPads are awesome, but I think laptops or computers are something that would make this process better."

The teacher also suggested that we should create student reference sheets or handouts so they could read the codes easier. In an email, the teacher described how this changed students interactions with Scratch: "I gave them the handout with the completed codes and this helped SO MUCH!! The students even commented on how helpful it was to have the reference page." Other suggestions included having more volunteers to help students with the coding process. The teacher described this in one of his interviews: "More people is always a plus in a room. We're doing good with the ones we have, but more hands to help more kids that need questions quicker is always, definitely a big thing." This was especially important as the students moved from the guided Scratch activities to building their own apps. In an email, the teacher described that "...it's been this second part with all of the students coding different projects that would be beneficial to have more hands or more experienced helpers in the room. There were times where [he] would get stuck helping one group for 10 minutes and couldn't make it around to help everyone else." To help scaffold students during that process, we created a full workbook (80 pages) with references and screenshots of how to implement specific codes they had learned.

Discussion of teacher support needs. The need for more computer science content knowledge is not new. Many teachers have described feeling apprehensive about working with computer science content, especially if it is a new area for them (Margolis et al., 2017; Ottenbreit-Leftwich & Biggers, 2017; Yadav et al., 2016). Professional development approaches like in-class coaching can be one way to help overcome these apprehensions (Margolis et al., 2017). For example, a study from Margolis and colleagues (2017) examined whether coaching could support classroom teachers with CS content and inquiry-based learning approaches. The authors found that the coaching PD improved teacher confidence with CS content, as well as their overall understanding of CS content (Margolis et al., 2017). Given these common apprehensions towards CS content knowledge, it was not surprising that our teacher mentioned a need for additional PD related to CS content.

Surprisingly, this teacher did not mention the need for more PD support for PBL. Other studies have described that teachers struggle with the new roles necessary with PBL, especially 'letting go' and enabling students to conduct their own research (e.g., Nariman & Chrispeels, 2016). To help teachers transition to this new pedagogical approach, Lee and Blanchard (2018) recommended that teachers engage in PBL professional development workshops as an interdisciplinary team, enabling them to implement these practices into their classrooms. They found that this helped motivate teachers to implement PBL (Lee & Blanchard, 2018). During the initial preparation for this implementation, we provided two days of PBL training where four teachers collaborated on our PBL CS kindness unit. Additionally, this teacher worked in a project-based STEM school, and had previously received a STEM fellowship. Perhaps due to these two reasons, the teacher already felt confident in implementing PBL as a pedagogical approach in his classroom.

Conclusion

The purpose of this research was to examine how PBL impacts students' CS interest and knowledge at the elementary level, and to determine how teachers implement a PBL CS curriculum and methods for enhancing that curriculum in future iterations. Results of this research suggest that integrating PBL and CS can lead to positive student outcomes as well as engage both the students and teachers in authentic problem-solving using CS. It is hoped that this research will lead to additional attempts to integrate inquiry-based instructional methods such as PBL with computer science instruction.

References

Belland, B. R., Glazewski, K. D., & Richardson, J. C. (2011). Problem-based learning and argumentation: Testing a scaffolding framework to support middle school students' creation of evidence-based arguments. *Instructional Science*, *39*(5), 667-694.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at the Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Brush, T. & Saye, J. (2017). *Successfully implementing problem-based learning in classrooms: Research in K-12 and teacher education*. West Lafayette, IN: Purdue University Press.

Chen, C. H., & Yang, Y. C. (2019). Revisiting the effects of project-based learning on students' academic achievement: A meta-analysis investigating moderators. *Educational Research Review*. doi: 10.1016/j.edurev.2018.11.001

Cliburn, D. C., & Miller, S. (2008). Games, stories, or something more traditional: the types of assignments college students prefer. *ACM SIGCSE Bulletin, 40*(1), 138-142.

Cohen, J. (1973). Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement, 33*(1), 107-112.

Cooper, S., & Cunningham, S. (2010). Teaching computer science in context. *ACM Inroads, 1*(1), 5-8.

Creswell, J. W. (2002). *Educational research: Planning, conducting, and evaluating quantitative* (pp. 146-166). Upper Saddle River, NJ: Prentice Hall.

Denner, J., Martinez, J., & Lyon, L. A. (2015). Computing for the social good: engaging Latino/a students in K-12. *ACM SIGCAS Computers and Society*, *45*(2), 31-32. doi: 10.1145/2809957.2809964

DeLyser, L. A., Mascio, B., & Finkel, K. (2016). Introducing student assessments with evidence of validity for NYC's CS4All. *In Proceedings of the 11th Workshop in Primary and Secondary Computing Education* (pp. 17-26). ACM.

Dwyer, H., Boe, B., Hill, C., Franklin, D., & Harlow, D. (2013). *Computational thinking for physics: Programming Models of physics phenomenon in elementary school.* In Engelhardt, Churukian, & Jones (Eds.) 2013 PERC Proceedings (pp. 133-136). College Park, MD: American Association of Physics Teachers.

Fayer, S., Lacey, A., & Watson, A. (2017). STEM occupations: Past, present, and future. Spotlight on Statistics.

Goode, J., Chapman, G., Margolis, J. (2012). Beyond Curriculum: The Exploring Computer Science Program. *ACM Inroads*. 3(2), 47-53.

Goldweber, M., Barr, J., Clear, T., Davoli, R., Mann, S., Patitsas, E., & Portnoff, S. (2013). A framework for enhancing the social good in computing education: a values approach. *ACM Inroads, 4(*1), 58-79.

Goldweber, M., J.C. Little, G. Cross, R. Davoli, C. Riedesel, B. von Konsky, and H. Walker. (2011). Enhancing the social issues components in our computing curriculum: Computing for the social good. *ACM Inroads, 2*(1).

Hoffman, B., Rosato, J., Morelli, R. 2019. Student engagement is key to broadening participation in CS. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), Minneapolis, MN. https://doi.org/10.1145/3287324.3287438.

Indiana DOE (2018). *Computer Science Resources - 6-8*. Retrieved from https://www.doe.in.gov/standards/computer-science-resources-6-8

Indiana General Assembly (2017). *Senate Bill 172.* Retrieved from http://iga.in.gov/legislative/2018/bills/senate/172

Lee, H. , & Blanchard, M. R. (2018). Why teach with PBL? Motivational factors underlying middle and high school teachers' use of problem-based learning. *Interdisciplinary Journal of Problem-Based Learning, 13*(1). Available at: https://doi.org/10.7771/1541-5015.1719

Liu, C.-C., Cheng, Y.-B., & Huang, C.-W. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education, 57*(3), 1907-1918. doi:10.1016/j.compedu.2011.04.002

Margolis, J., Ryoo, J., & Goode, J. (2017). Seeing myself through someone else's eyes: The value of in-classroom coaching for computer science teaching and learning. *ACM Transactions on Computing Education (TOCE), 17*(2), 6. doi: https://doi.org/10.1145/2967616

McKenney, S. & Reeves, T. (2014). Educational design research. In J.M. Spector et al. (eds.), *Handbook of Research on Educational Communications and Technology*. New York: Springer.

Nariman, N. , & Chrispeels, J. (2016). PBL in the era of reform standards: Challenges and benefits perceived by teachers in one elementary school. *Interdisciplinary Journal of Problem-Based Learning, 10*(1). Available at: https://doi.org/10.7771/1541-5015.1521

Ottenbreit-Leftwich, A.T. & Biggers, M. (2017). *Status of K-14 computer science education in Indiana: Landscape Report.* Submitted to the NSF's ECEP Alliance, the Indiana Department of Education, Governor of Indiana, Code.org, and Indiana legislators. Retrieved from https://www.csforin.org/

Ozturk, Z., Dooley, C. M., & Welch, M. (2018). Finding the hook: Computer science education in elementary contexts. *Journal of Research on Technology in Education, 50*(2), 149-163.

Pollock, L., Mouza, C., Czik, A., Little, A., Coffey, D., & Buttram, J. (2017, March). From professional development to the classroom: Findings from CS K-12 teachers. *In Proceedings of the 2017 acm sigcse technical symposium on computer science education* (pp. 477-482). ACM.

Ravitz, J. (2009). Summarizing findings and looking ahead to a new generation of PBL research. *The Interdisciplinary Journal of Problem-based Learning, 3*(1), 4-11.

SREB (2016. Bottoms, G., & Sundell, K. (2016). The Future of K-12 Computer Science Instruction. State Education Standard, 16(3), 24-31.

Stake, R. (1995). *The Art of Case Study Research.* Thousand Oaks, CA: Sage.

Stanton, J., Goldsmith, L., Adrion, W., Dunton, S., Hendrickson, K., Peterfreund, A., Yongpradit, P.,

Zarch, R., Zinth, J. (2017). State of the States Landscape Report: State-Level Policies Supporting Equitable K–12 Computer Science Education. Retrieved from https://www.edc.org/sites/default/files/uploads/State-States-Landscape-Report.pdf

U.S. Bureau of Labor Statistics. Employment Projections. Retrieved from http://www.bls.gov/emp/ep_table_102.htm

Walker, A., & Leary, H. (2009). A problem based learning meta analysis: Differences across problem types, implementation types, disciplines, and assessment levels. *The Interdisciplinary Journal of Problem-based Learning, 3*(1), 12-43.

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, *53*(4), 5-23.

Wirkala, C. & Kuhn, D. (2011). Problem-based learning in K-12 education: Is it effective and how does it achieve its effects? *American Educational Research Journal*, *48*, 1157-1186.

Yadav, A., Berges, M., Sands, P., & Good, J. (2016, October). Measuring computer science pedagogical content knowledge: An exploratory analysis of teaching vignettes to measure teacher knowledge. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education* (pp. 92-95). ACM.