Computational Thinking as a Foundation of Project-Based Learning:

A Literature Review

Amanda Dennard

Lamar University

EDLD 5314:

Dr. Julene Reed
Abstract

This paper examines published articles and reports from research to support the implementation of project-based learning as a learning strategy combined with computational thinking skills. As the evolution of technology continues to shape the way we work and learn, it is vital for students to know how to use technology to solve-problems. Using critical thinking and problem solving without consideration to technology leaves a gap in educational opportunities. Questions research include: how is computational thinking relevant in today’s classroom setting, is project-based learning an effective strategy to student-led, inquiry based learning, and does combining the two create more authentic learning experiences?

*Keywords*: computational thinking, problem-based learning, problem solving
Computational Thinking as a Foundation of Project-Based Learning

The Monthly Labor Review calculates eighteen percent of job growth in computer-related fields (2013). Foreseeing this need, organizations such as the Computer Science Teachers Association and the International Society for Technology in Education launched efforts to increase participation in Computer Science and computational thinking across the U.S (ISTE & CSTA, 2011). The use of technology in education is ever-changing and important in the way information is absorbed and shared and remains vital in maintaining social skills and work-place readiness (Becker at el., 2018). However, simply using technology is not enough. Technology should be a tool used to solve problems, create, and collaborate. Computational thinking encourages learners to systematically think about how to solve problems (Yadav, Hong, & Stephenson, 2016) while project-based learning provides an outlet for students to engage in student-centered activities that integrate with real-world problems (Chiu, 2020). The objective of this review is to explore the benefits of including the principles of computational thinking in project-based learning.

Literature Review

Computational Thinking

The idea of bringing computation to schools was introduced in 1980 by Papert. His book argues that children are quick to learn languages and that the process of programming transforms learning to be active and self-directed. Students engaging in programming use higher-level thinking skills and grasp other curriculum concepts, particularly math at a higher rate (Papert, 1980). Limiting these skills to computer science and programming leaves out the opportunity to apply computational thinking to multiple domains (Voogt, Fissure, & Good, 2015). As computer
use continues to increase in education, Janette Wing’s influential article, “Computational Thinking,” in the *Journal of the Association for Computing Machinery* expanded and clearly defined the idea of computational thinking. Her vision supports the idea that everyone, not just computer science majors, can benefit from thinking like a computer scientist. She states:

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science… It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use (Wing, 2006, p. 33).

Computer scientists and programmers often face complex problems with multiple interrelated parts. Their ability to break down and solve problems in systematic ways is also found outside of the computing field (Beecher, 2017). In essence, computational thinking can be broken down into four parts: problem decomposition, algorithms, abstraction, and automation. Decomposition involves breaking down a large problem into smaller, more manageable parts using algorithms, or sequences of steps used to solve problems. Abstraction reduces the complexity of the solution. Finally, students must determine if and how a computer could efficiently solve the problems through automation (Yadav et al., 2016). These steps can be cross-curricular and embedded across grade levels as early as kindergarten (Wing, 2011). Through the power of logic, computational thinking can redefine the way we teach all disciplines by empowering students to use technology to explore and create while solving problems (Angeli et al., 2016).

A study of a sixth-grade classroom using Scratch to implement computational skills found that students’ understanding of math concepts increased significantly compared to the control group.
This study highlights the impact computational skills have on overall student academic achievement (Calao, Moreno-Le, Correa, & Robles, 2015). Computers have changed the way work is done in almost every field. Students must learn how to engage with technology to solve problems in a variety of disciplines (Barr, Harrison, & Conery, 2011).

**Innovation**

For decades now, education systems have been talking about 21st century skills that need to be developed in learners in order to solve the problems of tomorrow. These skills include communication, collaboration, problem-solving and critical thinking, among others, all of which are integral to the study of Computer Science. Increasingly, countries are adjusting the 21st century skills to also include computational thinking. Contrary to popular belief, CS is not just about coding. It provides an outlet for creativity and innovation. It is about problem-solving and solution-building. And, it is about collaboration and teamwork, as groups of experts in various fields work together to address the complex issues facing our societies (Passey, 2017).

This rapidly changing future is a challenge for educational institutions. Educating students for the unknown seems impossible, however, we can equip students with an open attitude and a wide expanse of conceptual, cognitive, and social tools that encourage lifelong learning (Haddad, 2007). According to an Organization for Economic Cooperation and Development (OECD) report, “the pressure to increase equity and improve educational outcomes for students is growing around the world” (Vieluf et al., 2012, p. 3). The US in particular undergoes pressure from political, technological, and economic forces from inside and outside of the nation (Vieluf et al., 2012).
An important role of education is to serve the needs of society. In order to thrive, institutions must meet the changing needs of our fast-changing global world. A culture of innovation not only provides a positive change in education, but should be regarded as a necessary instrument (Serdyukov, 2017). Innovation in education raises productivity and efficiency and improves the quality of learning (Levitt, 2002).

**Project-Based Learning**

In today’s world educators are expected to enhance learning using technology-based tools while engaging students in active learning practices (Becker et al., 2019). Strategies like project-based learning proved activities that are student-centered and applicable to real-life practices. Teachers transform from givers of knowledge to facilitators of knowledge (Chui, 2020). High-quality project-based learning is defined by incorporating intellectual challenge, authenticity, collaboration, project management, reflection, and public product (Evans, 2019).

**Intellectual challenge and authenticity.** The active and experimental nature of authentic projects carry over to the expectations of real-world tasks (Becker et al., 2019). When selecting a project, it is important to consider interdisciplinary connections along with the relations to society in order to maximize student motivation and project authenticity (N, D. & G, S., 2018). The topic should be open enough to encourage exploration, but defined enough to guide research (Shafer, 2020). A case study for PBLWorks found that students are more engaged and inspired to work harder when their product had authentic, real-world connections beyond school (Evans, 2019). Incorporating real-world projects enhances education in a unique way (Chui, 2020).

**Collaboration and project management.** Teamwork is a critical component of building successful 21st-century learning communities. One of the biggest challenges faced by students
during projects lies in managing participation and equal distribution of effort (Evans, 2019). By learning how to communicate effectively through project management and team structures, students develop true collaboration skills (Chui, 2020).

**Public product.** A final product is the culmination of the research and effort put in by student teams. Through student-centered investigations, projects have the potential to have a significant impact on communities and society (Yadav et al., 2016). Having a final product to showcase gives students something concrete to work towards (Shafer, 2020). Students are motivated by having an audience outside of school observe their final products (Evans, 2019).

**Using Computational Thinking in PBL**

Combining computational thinking with project-based learning creates an effective learning environment with positive student outcomes (Wahono, Lin, & Chang, 2020). The combination gives students the opportunity to be active creators instead of simple consumers of technology (Yadav et al., 2016). Because problem-solving is a creative process, using systematic approaches while incorporating strategies and heuristics elevates a student's problem-solving skills (Beecher, 2017).

**Agile Scrum framework.** The Scrum framework is used in the Agile development process as a way to manage project development and is a way to incorporate computational thinking to projects. Teams collaborate to find efficient ways to solve complex problems by breaking them down into manageable chunks, or sprints. The team works together to prioritize and manage the workflow in ways that please the product’s users. (Schwarber & Sutherland). Combining Agile methodologies with project-based learning mirrors workplace problem-solving and collaboration (Santos, Sales, Fernandes, & Nichols, 2015). This type of system challenges students to fully
understand the start and endpoint while constantly reevaluating their goals and reinforcing their why (Beecher, 2017). These stages allow for continuous improvement and cause the students to deepen their understanding of the solution (Santos et al., 2015).

Using coding and other forms of technology allows students to express themselves in innovative ways. By adding culturally relevant pedagogy and multiple disciplines, schools become a place that welcomes contributions from all students and encourages computational competencies (Aghasaleh et al., 2018). A project-based learning approach that uses multiple layers and current technologies provides effective and engaging learning (Kite & Park, 2018). A study in Asia found that project-based learning was the most effective strategy to implement STEM learning. The combination gives an enriching experience especially if it involves local culture (Wohono et al., 2020). Additionally, a study of 9th grade students found that students engaged in STEM-based project-based learning as opposed to non-STEM project-based learning, showed larger strides in mathematics achievement and engagement (Lee, Capraro, & Bicer, 2019).

**Mobile applications.** Creating a mobile application to solve a problem is an example of combining project-based learning with computational thinking. Mobile development provides an innovative way to incorporate multiple disciplines (Nurbekova, Grinshkun, Aimicheva, Nurbekov, & Tuenbaeva, 2020). Applications vary in purpose from entertainment to education to communication and have changed the way we learn and communicate. Additionally, teachers found that using mobile applications increases classroom engagement and eases access to information (Chui, 2020). The main goals in developing an application would be learning computer science principles including computational thinking, understanding user interface, and
implementing the development cycle. This provides several stages for the students to research, design, debug, and test which simulates real-life scenarios (Chui, 2020). Novice programmers can gain the same benefits from visual programming environments such as Scratch (scratch.mit.edu). Visual-based programming languages are easier to understand but still allow for complex projects (Chui, 2020).

**Game Design.** Creating video games is another way to incorporate computational thinking and project based learning. The use of interdisciplinary gamification fosters students’ computational thinking and develops 21st-century competencies such as critical thinking, problem-solving, creation, and innovation while increasing engagement and motivation (Ifenthaler at el., 2017).

Game design provides a natural environment for exploring and curious interacting. This type of discovery learning provides a motivational advantage over a traditional teaching approach allowing students to directly explore and interact in an engaging way. When students are motivated to learn and create, they are more likely to use a constructivist approach to their learning, particularly when acquiring high-level skills needed to make games (Bauer, Butler, & Popović, 2017).

This enjoyable and collaborative environment provides motivation to learn complex algorithmic skills and increases development of computational thinking (Malizia at el., 2017).

**Teacher Training**

A major setback in implementing this initiative is lack of teacher experience. With nearly 1.4 million open computer science jobs open in the US and only a fraction of qualified works able to fill them, much of the nation is trying to expand Computer Science education and has placed
initiatives to encourage more teachers to become certified in Computer Science (Code.org, 2020).

In order to take on such a big task, teachers need to feel prepared and confident in incorporating computational thinking into their current curriculum. Initial exposure to ideas should be hand-on, engaging and fun. Content should be grounded in the teacher’s discipline and relevant to them (Gulamhussein, 2013). Coding and computational thinking must connect to content - coding to learn instead of learning to code so teachers directly see how they can apply the new material to their classes right away. Nothing helps students and teachers comprehend better than experiencing it first hand (Pope et al., 2011).

A study titled “Rocket to Creativity” found that dedicated Centers for Pedagogy provided great support for teachers developing innovative, high-quality learning experiences. These training labs gave teachers access to extensive resources and allowed them to develop cutting edge practices (Dole et al., 2016). The space alone is not enough to design effective and engaging programming lessons and projects, teachers must have an understanding of their individual student needs and current computational abilities (Lye & Koh, 2011).

When time to launch these new strategies in the classroom, teachers need to be fully supported. The implementation stage is critical to successful experiences and we should commit to supporting teachers during this time. Modeling and mentorship is a highly-effective practice that gives teachers support in the trenches especially when using new and experimental teaching models (Anderson & Stillman, 2013).
Conclusion

As technology continues to shape how we learn and work, it is crucial that today’s students learn not only how to use technology, but how to innovate with technology. Computational thinking skills should be used across disciplines to deepen students' problem solving abilities. Project-based learning provides a platform that encourages systematic approaches to solving problems while engaging students in quality collaborative environments. Past research indicates the strengths in applying projects to computational thinking. However, there are gaps in the way teachers are trained and prepared to implement computational thinking practices at its fullest. Research is still needed to answer what are the best pedagogical practices to deliver discovery based project learning.


Ifenthaler, D., Eseryel, D., & Ge, X. (2012). Assessment for Game-Based Learning. Assessment in Game-Based Learning, 1–8. https://doi.org/10.1007/978-1-4614-3546-4_1


