Examining the Potential of Computer Science and Making for Supporting Project-Based Learning

Technical Report · July 2017

Samuel Severance
Michigan State University

Examining the Potential of Computer Science and *Making* for Supporting Project-Based Learning

Samuel Severance
Susan Codere
Michigan State University

Emily Miller
University of Wisconsin - Madison

Deborah Peek-Brown
Joseph Krajcik
Michigan State University

Author Note

This white paper was solicited by Lucas Education Research, a division of the George Lucas Educational Foundation and will appear online at lucasedresearch.org.

Correspondence concerning this article should be addressed to Samuel Severance, CREATE for STEM Institute, Michigan State University, 620 Farm Lane, Room 115, East Lansing, MI 48824. Contact: severa18@msu.edu

Abstract

In this literature review, we seek to understand in what ways aspects of computer science education and *making* and *makerspaces* may support the ambitious vision for science education put forth in *A Framework for K-12 Science* as carried forward in the Next Generation Science Standards. Specifically, we examine how computer science and making and makerspace approaches may inform a project-based learning approach for supporting three-dimensional science learning at the elementary level. We reviewed the methods and findings of both recently published articles by influential scholars in computer science education and the maker movement and more foundational highly-cited articles pertaining to each approach. Our review found (1) making and makerspace approaches offer students, particularly from historically marginalized demographics, ample agency and opportunities for ownership over their learning but pose significant challenges for implementation at the elementary level at scale within a formal learning context; (2) computer science education, when effectively mediated with tools that lower barriers to entry, can help a range of students engage in meaningful computational thinking practices and may spur their interest in computers prior to more formal computer science education opportunities, yet such an approach will require careful consideration of how to sustain what is being implemented and subsequent computer science opportunities that students see as relevant. Our examination concludes with a discussion of congruencies and incongruences of computer science and making and makerspaces with project-based learning approaches aligned to science reforms, with applications to elementary units within the Lucas Education Foundation supported project, *Multiple Literacies in Project-Based Learning*.

Introduction

Recent visions of K-12 science education reform, as put forth in *A Framework for K-12 Science Education* (*Framework*; National Research Council, 2012) and carried forward in the Next Generation Science Standards (NGSS Lead States, 2013), call for dramatic shifts in how students should learn science. The *Framework* breaks from traditional notions of learning, where teachers see learners as “empty vessels” (Engeström, 1991) or “depositories and the teacher is the depositor” (Freire, 1970, p. 58), pouring knowledge into their heads, and instead demands students have more meaningful science learning opportunities. More specifically, all students should have the opportunity to “figure out” (Reiser, 2014) phenomena and problems relevant to their lives by engaging in science and engineering practices in order to coherently build core ideas and crosscutting concepts. Supporting the instructional shifts called for in the NGSS, at scale, will require that teachers have access to coherent, developed curricula aligned to the NGSS and provide support to teachers for understanding the underlying rationale of curriculum or what Cohen and Ball (1999) refer to as its “specification” (p.19).

One instructional approach that learning scientists and designers have used for developing curricula to support students’ in integrating the three dimensions of science learning called for in the *Framework* is project-based learning, where students investigate and solve meaningful problems over long durations by using science practices in order to build deep understandings of science ideas (Krajcik, McNeill, & Reiser, 2008). From Krajcik & Shin (2014), project-based learning curricular materials should adhere to the following design principles:

1) They start with a driving question, a problem to be solved.
2) They focus on learning goals that require students to demonstrate mastery on key science standards and assessments.

3) Students explore the driving question by participating in scientific practice processes of problem solving that are central to expert performance in the discipline. As students explore the driving question, they learn and apply important ideas in the discipline.
4) Students, teachers, and community members engage in collaborative activities to find solutions to the driving question. This mirrors the complex social situation of expert problem solving.
5) While engaged in the practices of science, students are scaffolded with learning technologies that help them participate in activities normally beyond their ability.
6) Students create a set of tangible products that address the driving question. These are shared artifacts, publicly accessible external representations of the class's learning.

Project-based learning approaches have proven effective at supporting students' learning. For example, Harris and colleagues (2015) performed a randomized controlled trial of the implementation of a research-based project-based learning curriculum called Project-Based Inquiry Science (PBIS) and found that the curriculum led to better learning outcomes for students who experienced project-based learning versus those in the control group who did not engage with the PBIS curriculum.

While project-based learning provides a sound instructional approach, how learning scientists and designers take up and apply it to new contexts remains of interest to the education community. Computer science education and making and makerspace efforts may provide potentially powerful new contexts for integrating a project-based learning approach. We seek to understand how project-based learning may couple with computer science education and/or making approaches and how such coupling may provide students with opportunities for meaningful science learning that shows integrity to the NGSS, specifically at the late elementary level. As such, we explore through a review of relevant literature the following overarching questions:

1) What benefits do computer science education and making and makerspace approaches consistently offer students and under what circumstances?
2) What challenges do computer science education and making and tinkering approaches consistently pose in their traditional contexts?
3) How might the benefits and challenges of computer science education and making and makerspace approaches inform the development of formal elementary project-based learning materials aligned to tenets of the *Framework* and NGSS?

Making and Makerspace Analysis

Making, the practice that takes place within communities of makers that comprise *makerspaces*, has become the subject of increasing interest to educators and researchers. Our analysis of highly-cited articles and work from recognized authorities in making and makerspaces, found that making has a specific connotation, but vagueness still exists around what, exactly, defines *making* (Martin, 2015). Broadly, scholars have defined making as "build[ing] or adapt[ing] objects by hand, for the simple personal pleasure of figuring out how things work'' (Honey & Kanter, 2013, p. 4) or "creative production in art, science, and engineering where people of all ages blend digital and physical technologies to explore ideas, learn technical skills, and create new products'' (Sheridan et al, 2014, p. 505). Others, such as Martin (2015), argue that making involves not only a "production" or "building" aspect but must also have a community infrastructure replete with online and in-person resources, spaces and events. In addition, making involves enacting the "maker mindset," (Martin, 2015, p.35), which includes values, beliefs, and dispositions common within makerspaces, such as an emphasis on play, supporting a growth orientation to learning, seeing failure as a positive, and promoting collaboration.

Beyond determining how to define making, a related and important question involves how making may support learners. Similar to the rationale for students to create final products or artifacts in project-based learning (Blumenfeld et al, 1991; Krajcik & Blumenfeld, 2006; Krajcik & Czerniak, 2013), proponents of making argue that maker activities afford desired forms of learning because “what” individuals make becomes the “evolving representation of the learner’s thinking” and allows for opportunities for “understanding through interpretation” (Halverson & Sheridan, 2014, p. 507). This idea of thinking with or through an object is hardly new, a pillar of project-based learning and several learning theories. Indeed, one could convincingly argue that all of sociocultural theory rests on the notion of utilizing tools as part of cultural mediation in order to support learning and development (Vygotsky, 1978). Much of the theoretical rationale underlying the maker movement, however, stems from constructionism (Martinez & Stager, 2013), as envisioned by Seymour Papert of *Mindstorms* and LOGO programming fame (see Papert, 1980).

Constructionism, a derivation of constructivism (Papert & Harel, 1991), calls for learners to engage in activities now commonly attributed to makerspaces: create artifacts where learning is distributed within the participating community. Constructionism calls for “building relationships between old and new knowledge, in interactions with others, while creating artifacts of social relevance” (Kafai, 2006, p.35). This emphasis on the “social” aspect of learning serves to demarcate constructionism from strict individualist constructivism. Instead, constructionism calls for “learning by constructing knowledge through the act of making something shareable” (Martinez & Stager, 2013, p. 21) and suggests learning “happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sandcastle or a theory of the universe” (Papert & Harel, 1991). Notably, the Multiple Literacies in Project-Based Learning curricular materials already align strongly to constructionism, for example, when 3rd grade students work together to apply their understanding of motion to develop and build toys or when 4th grade students collaborate to engage their understanding of erosion to design and build models of solutions to prevent damage to structures in communities.

Benefits of Making and Makerspaces

During our review, we identified key benefits from making and makerspaces—in addition to the generally accepted theoretical rationale of constructionism and the benefits of enacting such a perspective—related to supporting more equitable science learning experiences, where *all* students have the opportunity to engage in learning they find meaningful and relevant. Specifically, our review found that making shows potential for (1) supporting a range of learners in having access to opportunities to pursue identities in STEM by expanding who can be a “maker”; and, relatedly, (2) empowering young learners to experience agency (i.e., the capacity to produce an effect (Kaptelinin & Nardi, 2006)) and sense of self-efficacy (i.e., the belief in one’s abilities to succeed (Bandura, 1977)) over their learning when engaging in making activities. Of note, these equity-oriented benefits mirror the desired outcomes for the Multiple Literacies in Project-Based Learning curricula.

Access to opportunities for identity development. Achieving more equitable outcomes for all students requires all students have opportunities for meaningful learning (NRC, 2012; Welner & Carter, 2013). Making has the potential to provide an opportunity for meaningful learning to support equitable aims. Rampant commercialization of the maker movement, however, and portrayals of making in the media that popularize making as an activity for white males, have served to reproduce historical inequities that marginalize youth of color and girls, situating them outside of trajectories that could increase their representativeness in STEM careers (Vossoughi, Hooper, & Escude, 2016). In response, several researchers and scholars have begun to productively problematize what it means to be a “maker” and challenge the inclusiveness of the maker movement.

Makers and researchers are beginning to productively “re-mediate” (Gutiérrez, Morales, & Martinez, 2009) past conceptions of making and the maker movement into approaches that allow for all learners to have the opportunity to identify as a “maker,” as someone who can do science and engineering as is the focus of the Multiple Literacies in Project-Based Learning. Sheridan and colleagues (2014) examined a community-based makerspace in Detroit called *Mt. Elliott Makerspace* and found that youth participants there regularly underwent “dispositional shifts” (518), that is, a change in their thinking where they began to think deeply about activities related to making they had *not* considered previously. At another site, the museum-based *Makeshop* in Pittsburgh, Sheridan and colleagues (2014) also observed how a common activity outside of STEM, sewing, could facilitate access into the making community for youth. Making tends to take an approach to building and creating objects by purposefully bringing in non-STEM disciplines to promote access for learners. Kafai and colleagues (2014), for example, draw on programmable e-textiles and put forth the notion of a "culturally hybrid construction kit" (p.538) (e.g., sewing *with* programming, circuits *with* cloth) that can attract different demographics (e.g., males and females) while still engaging them in science and technical complexities they find meaningful.

Opportunities for agency and self-efficacy. Closely related to supporting learners in having access to opportunities that allow them to develop and validate identities within makerspaces, recent work has also shown possible mechanisms that support this, chiefly, that learners in makerspaces experience meaningful agency and an increased sense of self-efficacy. Martin (2015) characterizes making as allowing for freedom and choice, a design principle in the Multiple Literacies in Project-Based Learning materials, in determining what to create but also emphasizes the need for providing infrastructure and community assets to allow for learners to grow and experience success, perhaps after a chain of failures. Barton and colleagues (2016) echo this approach within their after-school-based makerspace, where middle school girls of color felt a sense of empowerment and ownership over their learning. Barton and colleagues (2016) had students engage in ethnographic data collection to identify pressing problems in their own community from which students chose a problem to tackle. Barton and colleagues (2016) purposefully applied a light pedagogical touch "to productively engage with individual youth in ways that honor how they bring their particular interests and experiences to bear on the making enterprise" (p. 15) as students developed and owned potential solutions to their chosen problems and gained confidence in overcoming failures. Similarly, Sheridan and colleagues (2014) found that providing students with opportunities where they develop agency, where they could choose what they wanted to do but had “just-in-time” supports in place to support learners’ growth, fostered ownership and self-efficacy.

Challenges of Making and Makerspaces

Although an abundance of literature extols the virtues of making and makerspaces, challenges occur in developing and enacting maker activities. Notably, (1) the nature of primarily self-directed, Do-It-Yourself maker activities leads projects to have extended timelines and material needs; and (2) the benefits of makerspaces are reserved only for those who actually have the opportunity to experience them in typically out-of-school or limited school enactments.

Extended time periods for maker activities. Activities observed within a makerspace can range in their length of time, from under a day (see Sheridan et al, 2014) or up to several weeks or even months (see Barton et al, 2016). The reasons for the extended timeline of maker projects are not explicitly addressed in the literature we reviewed, however, we surmise that the "maker mindset" (Martin, 2015) engenders an approach that requires long periods of time. Specifically, Martin (2015) notes certain characteristics of the maker mindset that likely contribute to extended timelines for projects, namely emphasizing playfulness, encouraging asset and growth-oriented mindsets, and seeing failure as a positive. To maximize engagement and motivation, makerspaces typically encourage learners to have few constraints on what materials to use, how to use them, or how quickly to use them, and instead encourage extended periods of play and exploration. A growth-mindset, echoing work from Dweck (2006), and seeing failure as a positive also lend themselves to longer timescales of learning, as learners make and must learn from mistakes over time. The Multiple Literacies in Project-Based Learning project also fosters a growth mindset by engaging learners in iteratively building artifacts and arguing about findings, while attempting to focus choice without limiting engagement (difficult to do but necessary to scale and when focusing on learning goals).

Limited school enactment. Overwhelmingly, makerspaces and associated maker activities occur in informal settings such as after-school clubs (e.g., El Pueblo Mágico (DiGiacomo & Gutiérrez, 2016) or community and museum-based centers (e.g., Exploratorium's Tinkering Social Club). While numerous individuals have and will continue to have access to these makerspaces, the question of *who still may not have access to these spaces* remains. Certain makerspaces have provided opportunities for historically underserved youth to engage in making (see Sheridan et al, 2014), yet these "pockets of success around the nation[…]are the exception and not the norm (Barton, Tan, & Greenburg, 2016, p.30). Students deemed high-risk for behavior problems, attendance issues, and drug use often show higher attrition rates for after school activities, with relocation and access to transportation cited as significant factors (Weissman & Gottfredson, 2001). For community and museum-based makerspaces, relocation and transportation barriers also likely influence participation, including limiting the sustained attendance necessary for more involved making projects. Limited enactment of making in the science classroom has occurred via teachers using traveling "Maker Carts" and dedicated maker rooms, but usually only as an add-on to curricular activities rather than replacing core elements of a curriculum (Bevan, 2017). For comparison, project-based learning has traditionally stressed the making of artifacts in a formal science classroom to address important learning goals (Spitulnik, Stratford, Krajcik, & Soloway, 1997; Spitulnik, Zembal, & Krajcik, 1998; Wisnudel-Spitulnik, Krajcik, & Soloway, 2000)

Computer Science Education Analysis

Reviewing the whole of computer science education poses a daunting challenge, particularly when many notions of computer science abound (Barr & Stephenson, 2011). Much of the research in computer science education focuses specifically on higher education students learning to program using computer code (e.g., McCracken et al, 2001). We limited our focus to primarily the K-12 level while showing preference to articles at the K-5 level, and sought highly cited articles or articles by recognized authorities in computer science education. At the same time, we sought to adhere to the interest of the larger computer science education community in how people learn to program or code. We demarcate our interest in computer science education to a broad learning-to-program perspective where students have opportunities to place themselves on trajectories leading to deeper participation in computer science later in their learning.

An important element of computer science education, and to a certain extent in making as well (see Kafai et al, 2014), is the notion of computational thinking. Like computer science education, multiple notions of computational thinking also exist (Barr & Stephenson, 2011). Computational thinking has a prominent place as part of one of the essential science and engineering practices in the NGSS: *using mathematics and computational thinking* (NGSS Lead States, 2013). In this frame, computational thinking refers primarily to the capacity to use computational tools—including computers—and computational skills, such as constructing simulations, statistically analyzing data, applying quantitative relationships, and mathematically testing design solutions (NRC, 2012), to develop deeper understandings of core ideas in science and engineering. Computational thinking from an NGSS perspective focuses on examining the properties and relationships within systems and discerning patterns, often through engaging with—or creating—computer models of systems (Wilkerson & Fenwick, 2017). Within the literature of computer science education, computational thinking features prominently as a goal for formal computer science educators (Guzdial, 2008). Wing (2006), in particular, receives credit for successfully arguing for the necessity of computational thinking in computer science education, which she defines as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33). Significant components of computational thinking within computer science education include, among others, abstracting, systematic information processing, modularizing, conditional logic, and debugging (Grover & Pea, 2013).

Similar to the maker movement, the design of computer science education interventions at the K-12 level includes drawing on constructionism for a theoretical frame and design principles

(Kafai, 2006). As mentioned previously, constructionism became known through the seminal work *Mindstorms* by Papert (1980), which includes a description of his work with young students and how he sought to organize their engagement with the LOGO programming language:

> In most contemporary educational situations where children come into contact with computers, the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, to provide feedback, and to dispense information. The computer programming the child. In the LOGO environment the relationship is reversed: The child, even at preschool ages, is in control: The child programs the computer. (19)

Though receiving significant pushback for allegedly overstating the promise of programming for supporting learning more broadly (Pea & Kurland, 1984), constructionism's approach of providing students with opportunities to construct their own tangible programs and to construct "new relationships with knowledge in the process" (Kafai, 2006, p.38), has certainly endured in subsequent years, particularly within computer science education.

Benefits of Computer Science Education

Our review found that students at the K-12 level experienced consistent benefits when they had the opportunity to engage in quality computer programming activities. Benefits for students included (1) engaging in meaningful aspects of computational thinking that can support learning in other domains; and (2) when provided with computer programming tools that lower barriers to entry, students can develop an interest in computer science and coding more generally.

Computational thinking supports learning across domains. When K-12 students had the opportunity to engage in computer coding through programs they found accessible, they engaged in several aspects of the practice of computational thinking, particularly debugging and modularization, that supported learning in other domains. Dalton and Goodrum (1991) examined the effects of elementary and junior high students' engagement with a LOGO and BASIC computer programming course with supplemental instruction on problem-solving. While the junior high students saw insignificant gains, the elementary students significantly improved their problem-solving skills, including higher performance on a mathematics assessment (Dalton & Goodrum, 1991). Similarly, Noss (1986) found that engaging elementary students in LOGO programming supported algebraic thinking. More recently, Kafai (2006) reported a series of studies where elementary students programmed their own video games. Kafai (2006) found persistent gender differences in students' designs for math games but no significant differences their design of science games and no gender differences overall in the proficiency of making games in math and science.

Accessible coding programs support interest in computer science and coding. Decades ago, Papert (1980) recognized the need to develop a computer programming language that young students could easily engage with. The product of efforts at MIT, Papert's university, LOGO has proven accessible and shown success in spurring students' interest in computer science and coding. From Dalton and Goodrum (1991), while elementary students using LOGO did not show significant improvement in their attitudes towards computers, junior high students did show a significant improvement in their attitudes towards computers, particularly when they received additional instruction in problem-solving. In the years since LOGO's inception, other simple programming platforms have come to the fore for K-12 students to use. Maloney and colleagues (2008) examined the effects of elementary and secondary students' engagement with *Scratch*, a simple drag-and-drop programming interface, within an after school Computer Clubhouse setting in an urban area. Maloney and colleagues (2008) found that a range of youth found their experiences with Scratch empowering and surmised that this interest might serve as a promising pathway into programming by “going beyond mere consumption to become content creators themselves, a role often denied to urban youth” (n.p.).

Challenges of Computer Science Education

Numerous challenges pose significant barriers to effective computer science education, including the usual suspects such as lack of consistent access to computer hardware and software. Select articles, however, raised issues related to developing quality computer science education innovations and sustaining their enactment within K-12 settings pertinent to our larger curricular interests. As such we focus on the following challenges: (1) how to support the implementation of an innovation in K-12 settings and create pathways of learning opportunities for students in computer science over time; and (2) how to effectively contextualize the use or need for students to engage in programming.

Attending to implementation over time. In order for educational innovations to achieve maximal impact, researchers and designers must attend to the capacities of districts and schools to actually take up and sustain the implementation of an intervention over time (Penuel, Fishman, Haugan Cheng, & Sabelli, 2011). All too often, promising educational innovations fail to achieve the aims of designers when taken beyond the scope of preliminary studies due to systemic constraints (e.g., assessment alignment, pacing guides) within districts and schools (Elias et al, 2007; Penuel & Fishman, 2012). Computer science education scholars have begun to problematize and address the issue of implementation. Reppenning and colleagues (2010) reported on their efforts to enact—at scale—their programming platform for middle school students to code games and provide a checklist of items they claim must be met for a computational thinking innovation to become successfully and taken up and implemented in schools:

1) **has low threshold**: a student can produce a working game quickly.
2) **has high ceiling**: a student can make a real game that is playable and exhibits sophisticated behavior, e.g., complex AI.
3) **scaffolds Flow**: the curriculum provides stepping stones with managed skills and challenges to accompany the tool.
4) **enables transfer**: tool + curriculum must work for both game design and subsequent computational science applications as well as support transfer between them.
5) **supports equity**: game design activities should be accessible and motivational across gender and ethnicity boundaries.
6) **systemic and sustainable**: the combination of the tool and curriculum can be used by all teachers to teach all students (e.g., support teacher training, standards alignment etc.).

Bringing coding into the K-12 classroom, however, must occur consistently. Rather than isolated opportunities, providing K-12 students with more and consistent opportunities to learn about coding over time can lead to their pursuing more opportunities in computer science (Guzdial et al, 2014). The vast majority of opportunities for K-12 students to engage in programming activities, however, are concentrated at the high school level (e.g., with AP computer science classes), followed by sparse opportunities at the middle and elementary school levels.

Making computer science relevant in students' learning. Moving from the macro view of looking at how to successfully implement and sustain a programming intervention, we now move to the micro view of how to flexibly design curriculum to effectively maintain the interest of individual students. That is, how to support students in seeing programming as relevant to their learning. Part of this issue overlaps with the challenge of attending to the implementation of computer science educational innovations as delineated previously, particularly the aspects of Reppenning and colleagues’ (2010) checklist that demand attending to equity and accessibility. Recent efforts by Pinkard and colleagues (2017) found success in supporting students’ interest by contextualizing computational experiences within a “narrative” or story, including allowing for students to craft their own narratives. Notably, Pinkard and colleagues (2017) employed a mix of maker and computer science activities in their approach, similar to Kafai and colleagues (2014) efforts with programmable e-textiles. A purposefully hybrid approach may offer utility in terms of providing students with opportunities to draw on other expertise outside of computer science while still leading students to recognize the relevance of programming and coding in terms of maintaining their agency and ownership in their learning.

Implications for Project-Based Learning

In order to ascertain the benefits and challenges of making and computer science education (i.e., coding), this review examined highly-cited articles and articles from scholars recognized as leaders in their respective fields, whether it be in the area of making or computer science education or a combination of both areas. This review, however, also sought insights into how making and makerspaces and programming in computer science education may inform the development of elementary project-based learning curricular materials that show integrity to the vision for science education in the *Framework* and carried forward in the NGSS. Below, we highlight (1) areas of promising congruence between making and computer science education with project-based learning aligned to the NGSS, and (2) challenging areas of incongruence between making and computer science education with project-based learning aligned to the NGSS. (For a comparison of approaches along selected constructs, see Table 1).

Promising Areas of Congruence

In addition to the overlap making and computer science education share with one another, as exemplified in the work of Kafai and colleagues (2014), making and computer science education also share significant, natural overlaps with project-based learning. Indeed, current efforts in the *Multiple Literacies in Project-Based Learning* (ML-PBL) project embody affordances identified as present in both making and computer science education. Notably, project-based learning (1) shares with making and computer science education a common theoretical rationale rooted in constructionism; (2) provides rich opportunities for computational thinking, as shared primarily with computer science; and (3) seeks to provide opportunities for students' to experience agency in their learning and community, as shared primarily with making.

Common theoretical perspective of constructionism. Project-based learning, like making and computer science education strands focused on students creating programs, draws strongly on perspectives within constructionism (Grant, 2002). Specifically, as called for in constructionism as delineated by Papert (1980), project-based learning demands students create shared artifacts (Krajcik & Shin, 2014; Spitulnik, Stratford, Krajcik, & Soloway, 1997; Spitulnik, Zembal, & Krajcik, 1998; Wisnudel-Spitulnik, Krajcik, & Soloway, 2000). During the creation of artifacts in project-based learning, students think with and through artifacts to engage with core ideas in science and engineering to address a driving question. As students construct artifacts, they construct their ideas: "the doing and the learning are inseparable" (Blumenfeld et al, 1991, p.372). Constructionism's emphasis on the social and cultural influences for constructing knowledge also has strong representation in project-based learning. Project-based learning calls on students to engage in socially-oriented science and engineering practices (e.g., argumentation), which allows their learning to become a collective "cultural accomplishment"

(NRC, 2012, p.283). Students develop a shared, public artifact that addresses the driving question of the project or their own sub-question, is representative of students' emergent states of knowledge, and can facilitate students' examination of one another's understandings (Blumenfeld et al, 1991). Within the ML-PBL project, for example, each unit demands students collaboratively engage in science and engineering practices to develop a public artifact that serves as an instantiation of students' understanding of a real phenomenon or solution to a design challenge that relates to the driving question.

Opportunities for computational thinking. Computational thinking, as primarily defined within computer science education (see Grover & Pea, 2013), has a strong presence within project-based learning in science aligned to the NGSS, including the ML-PBL project. Computational thinking, within an NGSS frame, calls for students to explore systems and their components and discern patterns, usually through the use or creation of computer models of systems (Wilkerson & Fenwick, 2017). Explaining complex phenomenon, which can be represented as a system and modeled, has a central place in project-based learning (Krajcik & Blumenfeld, 2006), as does leveraging technological tools while engaging in science practices (Krajcik & Shin, 2014), such as computational thinking. In its course guide *AP Computer Science Principles*, the College Board (2016) conceptualizes a set of Computational Thinking Practices, which include abstracting/developing models and simulations, analyzing problems and artifacts, communicating, and collaborating. These practices, as well as others such as debugging (Grover & Pea, 2013), all have a strong presence within project-based learning, including the ML-PBL project (e.g., 4th grade students developing a model of their solution to erosion). Of note, project-based learning that does not have students code an artifact, seems less able to meet other aspects of computational thinking such as creating computational artifacts (College Board,

2016) and modularizing (Grover & Pea, 2013).

Promoting student agency. Congruent aims of empowering learners common to makerspaces (see Barton et al, 2016), and to a lesser degree computer science education (see Maloney et al, 2008), project-based learning seeks to support learners in experiencing and developing agency through their learning. We take agency, broadly, to mean the capacity of individuals to produce an effect (Kaptelinin & Nardi, 2006), and give careful consideration to the volition or control an individual has in bringing about a desired effect, such as solving a problem, making sense of a phenomena or other new learning, or a change in the world. In project-based learning, students have ample choice or control over their actions and learning, such as decisions during group work, asking and exploring sub-questions, and determining how to create a final artifact (Blumenfeld et al, 1991). Moreover, to facilitate students' engagement in science and engineering practices, project-based learning typically provides students with auxiliary cognitive tools (e.g., technological tools and scaffolds for complex practices like constructing explanations) (Krajcik & Blumenfeld, 2006), which effectively enhances students' agency over their own learning. In terms of students having agency to achieve broader effects, project-based learning can engage students with problems and questions they see as relevant (Blumenfeld et al, 1991). The effect students can have in addressing these problems and questions can go beyond learning *about* the world within the four walls of the classroom to acting *in* and affecting the world (Eisenhart, Finkel, & Marion, 1996; Severance, 2016). Within the ML-PBL project, for example, we provide lessons where students tackle local problematic phenomena in their communities and seek to address them (e.g., voicing concerns over local energy policy).

Challenging Areas of Incongruence

In addition to promising areas of congruence, our review found areas where project-based learning shows significant incongruences with descriptions of making and computer science education. The incongruences we identified pertain to our underlying interest in curriculum development and implementation in formal elementary school settings as part of the ML-PBL project. These incongruences may pose significant challenges for implementing a project-based learning approach with a making or computer science education component in K-12 settings. Specifically, we see significant challenges to (1) working within the constraints of a formal school classroom to enact previous approaches for making and, to a lesser extent, computer science within a project-based learning curriculum, and (2) reconciling the primary learning goals of previous making and computer science education approaches with the aims of a project-based learning approach designed to embody visions of science education reform.

Working within the constraints of formal school settings. While attempts to bring making approaches into school settings, such as “Maker Carts” and dedicated maker rooms have occurred (Bevan, 2017), making and makerspaces, from our review, seem to predominantly become enacted in informal settings such as after-school clubs (see Barton et al, 2016), and community or museum makerspaces (see Sheridan et al, 2014). Within these informal spaces, by and large, students tend to have extended access to time and materials to develop, iterate, and complete involved projects. In comparison, project-based learning, as commonly conceptualized (Blumenfeld et al, 1991), occurs predominantly within formal school and classroom settings, where teachers and students must navigate inflexible time and materials constraints, such as bell and calendar schedules and the availability of centralized science materials. The ML-PBL project, for example, must work within the time constraints of participating districts and constraints of materials suppliers to achieve enactment at scale.

Much of the need for extended access to time and materials in a makerspace stems from the “maker mindset” (Martin, 2015), which would likely come into tension with the constraints in formal settings. Specifically, the maker mindset calls for extended play and providing students with opportunities for multiple cycles of failure and iteration (Martin, 2015). Replicating a more open-ended, iterative maker approach, seen as particularly powerful for learners (see Barton et al, 2016; Kafai et al, 2014), would add to the length of time and materials needs students would have to have in formal settings. A project-based learning approach, on the other hand, navigates constraints of time and materials needs by seeking a productive balance of open-endedness and attending to constraints. Project-based learning provides some open-endedness, such as allowing for multiple solutions or artifacts to a given problem (Blumenfeld et al, 1991), but students’ experiences are carefully circumscribed and processes are streamlined with cognitive tools so as to fit the constraints of schools. Perhaps investigating how to foreground and extend existing areas of congruence project-based learning has with making (e.g., shared emphasis on artifacts) could prove a way forward. One possibility includes investigating the merit of coupling informal learning spaces to extend work occurring in the formal classroom. In addition, project-based learning’s tendency to bring in auxiliary tools, such as technological tools (Krajcik & Shin, 2014), could prove a means for facilitating and streamlining activity, leading to timeframes and material use more acceptable in formal settings.

In terms of computer science education, researchers have had some success in integrating coding and programming approaches into formal K-12 settings to support subject-specific domain learning in science (see Wilensky & Reisman, 2006). Integrating a computer science approach within our interpretation of formal project-based learning would potentially promote students' engagement with the science and engineering practices, notably *using mathematics and computational thinking* (NGSS Lead States, 2013), and seems an apt opportunity to leverage computer science and programming activities in formal settings. Marx and colleagues (1997), for example, surmised that a computer program could serve as the final artifact within a project-based learning sequence. How to properly embed computer science within project-based learning, however, so as to create a need for coding during learning, remains a design tension, as the need depends largely on the phenomena of interest and driving question that is aligned to NGSS learning goals. Relatedly, determining which computer science tool(s) (e.g., Scratch) may prove most apt for students to use within a broader project-based learning curriculum, particularly at the elementary level, remains a challenge from a science curriculum development standpoint and, as Repenning and colleagues (2010) have noted, for scaling up innovations in formal settings.

Incongruent aims and approaches and science education reform. While we acknowledge variance exists within the maker movement and within the computer science education community, our review found that the overall aim of instruction in making and computer science education can often prove incongruent for supporting subject-specific learning goals, such as those in science associated with the performance expectations of the NGSS (NGSS Lead States, 2013). In contrast, efforts in science education using a project-based learning approach, such as the ML-PBL project, demand attending to learning goals and having them inform design (Krajcik, McNeil, & Reiser, 2007). Given the ambitious nature of what constitutes meaningful science learning in the NGSS—the taking up of science and engineering practices to build core ideas and crosscutting concepts (NGSS Lead States, 2013)—instruction to this end requires instructional approaches that align to the rationale and aims of science education reform and that can best facilitate instantiating new visions of science learning in the classroom (NRC, 2012).

The goal of providing opportunities for learning through making and within makerspaces, from our review, does not seem to intentionally attend to rigorous learning goals. Instead, much of the purpose of making and makerspaces revolves around supporting the development of a "maker" identity (Martin, 2015), and, more recently, promoting more equitable outcomes for students by providing interventions that support historically underserved students' interest in STEM (Barton et al, 2016; Vossoughi et al, 2016). While we acknowledge and applaud work that rightly foregrounds how interest and identity play an important role in developing mastery and expertise, the open-ended, self-directed nature of making seems less conducive to providing students with learning experiences, as envisioned in the *Framework* and NGSS, where they can develop in a coherently organized manner over time, deep understandings of scientific core ideas underlying *phenomena* while developing the capacity to effectively take up and apply an *array* of science and engineering practices that will result in learners building agency—the capacity to produce an effect.

In defense of making, attending to top-down science education reforms and their associated standards seems almost antithetical to the bottom-up, interest driven, rationale we found pervasive in making efforts (Project-based learning, to be fair, also carves out agency and supports students' interest by organizing around driving questions and artifact creation). Makerspaces, however, have the opportunity to capitalize on reform efforts, particularly given efforts to formalize design and engineering across the country (Carr et al, 2012), and the emphasis on engineering in the NGSS (NGSS Lead States, 2013). Perhaps integrating a project-based learning curriculum with elements of making would better allow for making to become more central in science education reform. Still, substantial compromises would likely have to occur (e.g., promoting an “engineer” or “scientist” identity over a “maker/hacker”), if the learning goals of the NGSS are to remain the learning goals of a project-based learning curriculum.

While computer science education has shown some success in becoming taken up and supporting K-12 learning goals in subject specific domains like biology (see Wilensky & Reisman, 2006) and physics (see Guzdial, 1994), our review found that the learning goals for computer science education can often prove incongruent with the aims of science education reforms like the NGSS. Specifically, some strands of computer science education have developed a fixation on computational thinking, leading to interventions that place increasing students’ capacity for computational thinking as the primary learning goal. In terms of a more integrated approach to science learning, as called for in the *Framework* and NGSS, where practices (as computational thinking is defined) are not treated as learning goals in and of themselves in isolation from core ideas and crosscutting concepts, focusing on computational thinking as the sole learning goal proves problematic.

Ideally, within a project-based learning curriculum, students would use computer programming to engage in the practice of computational thinking (Blumenfeld et al, 1991), in order to support making sense of phenomena and designing solutions to problems while developing understanding of rigorous learning goals. To be fair, recent advanced courses in computer science, namely AP Computer Science Principles, have moved to contextualize learning around problems that students may find interesting, but the underlying emphasis remains on computational thinking (College Board, 2017). Rather than replicating approaches that continue to see computational thinking as a sufficient and appropriate learning goal in and of itself or seek to increase learners’ capacity for coding generally (e.g., Apple’s “Everyone Can

Code" curriculum), students engaging with a project-based learning science curriculum will have to have experiences that better reflect the rationale of the *Framework* and NGSS. From an integrative perspective, reflective of the *Framework* and NGSS, students should have the opportunity to engage in coding in order to wield computational thinking as a means for learning science.

Discussion and Conclusion

In this review we sought to explore the dual topics of making and computer science education, in order to better understand the ways in which they may inform the development of project-based learning curriculum materials in science at the elementary level as part of the *Multiple Literacies in Project-Based Learning* project. While gleaning insights from across the vast expanses of both making and computer science literature proved challenging, our focus on elementary science learning, as envisioned in the *Framework* and NGSS, provided an anchoring context. This context proved useful for deriving potential affordances and challenges in accommodating a making or computer science approach in project-based learning more broadly. In terms of pragmatic contributions, our review provides curriculum designers and researchers interested in exploring how to infuse making and/or computer science into project-based learning with new insights to consider when engaging in project-based learning curriculum development.

In this review, we identified key features and trends the making and computer science education communities have established, and showed that these features and trends present curriculum designers and researchers with potential affordances and certain challenges. We have shown that the numerous affordances of making and computer science can closely overlap with principles and values deeply held by many curriculum designers and researchers working to employ project-based learning (and could convincingly argue that they already promote), such as attending to issues of equitable access and supporting students' interest and identity trajectories for participation in STEM. While we have identified several areas of congruence from which researchers could and should begin to explore, we note that once researchers begin to embark upon this integrative work, new unforeseen "as-created tensions" (Tatar, 2007) will no doubt arise during development. These new tensions may well challenge our assertions here of the promise of congruent areas among making and computer science with project-based learning.

Perhaps most significantly, we have identified and begun the conversation around challenging tensions, or areas of incongruence, that exist between making and computer science and project-based learning, particularly in regards to the challenge of achieving materials that best reflect the vision of the NGSS and science education reform, as well as areas of congruence (i.e., engaging learners and building agency in learners). Chief among these tensions is determining the feasibility of integrating with *integrity* to their respective communities, elements of making and computer science into project-based learning at the elementary level while still showing *integrity* to established and proven tenets within project-based learning and the aims of science education reform. Tensions, beyond being potential pitfalls or barriers, can prove a productive impetus for innovative design (Engeström & Sannino, 2010). Effectively addressing the central tension identified here will require developing new ways to address the dual design constraints we identified, of working within formal settings and attending to specific learning goals derived from science education reforms.

Reconfiguring making and computer science education seems a likely course of action to address constraints we have identified that impede their implementation as part of an NGSS-aligned project-based learning approach. Overcoming these constraints within a project-based learning frame may lead to new forms of making and computer science education, ideally in ways that provide learners' with further opportunities to develop agency through choice in creating artifacts and in figuring out phenomena and solutions to design challenges. To effectively achieve such innovation, the researchers and partners involved *must* hold to an overarching goal or aim for design. The vision of science education reform, as put forth in the *Framework* and NGSS, will continue to guide our design, and we argue that elements of making and computer science should be taken up in service to the aim of providing all students with opportunities for meaningful science learning.

References

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, *84*(2), 191.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *Acm Inroads*, *2*(1), 48-54.

Barton, A. C., Tan, E., & Greenberg, D. (2016). The makerspace movement: Sites of possibilities for equitable opportunities to engage underrepresented youth in STEM. *Teachers College Record*.

Bevan, B. (2017). The promise and the promises of Making in science education. *Studies in Science Education*, *53*(1), 75-103.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational psychologist*, *26*(3-4), 369-398.

Carr, R. L., Bennett, L. D., & Strobel, J. (2012). Engineering in the K-12 STEM Standards of the 50 US States: An Analysis of Presence and Extent. *Journal of Engineering Education*, *101*(3), 539-564.

Cohen, D. K., & Ball, D. L. (1999). Instruction, capacity, and improvement. CPRE Research Report Series RR-43. Philadelphia, PA: Consortium for Policy Research in Education, University of Pennsylvania

College Board. (2017). *AP Computer science principles*. New York: College Board. https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf

Dalton, D. W., & Goodrum, D. A. (1991). The effects of computer programming on problem-solving skills and attitudes. *Journal of Educational Computing Research*, *7*(4), 483-506.

DiGiacomo, D. K., & Gutiérrez, K. D. (2016). Relational equity as a design tool within making and tinkering activities. *Mind, Culture, and Activity*, *23*(2), 141-153.

Dweck, C. S. (2006). *Mindset: The new psychology of success*. Random House Incorporated.

Eisenhart, M., Finkel, E., & Marion, S. F. (1996). Creating the Conditions for Scientific Literacy: A Re-Examination. *American Educational Research Journal*, *33*(2), 261–295. http://doi.org/10.3102/00028312033002261

Elias, M. J., Zins, J. E., Graczyk, P. A., & Weissberg, R. P. (2003). Implementation, sustainability, and scaling up of social-emotional and academic innovations in public schools. *School Psychology Review*, *32*(3), 303-319.

Engeström, Y. (1991). Non Scolae Sed Vitae Discimus: Toward Overcoming the Encapsulation of School Learning. *Learning and Lnstrucrion*, *1*(1989), 243–259.

Engeström, Y., & Sannino, A. (2010). Studies of expansive learning: Foundations, findings and future challenges. *Educational Research Review*, *5*(1), 1–24. http://doi.org/10.1016/j.edurev.2009.12.002

Freire, P. (1970). *Pedagogy of the oppressed*. New York: Herder & Herder.

Grant, M. M. (2002). Getting a grip on project-based learning: Theory, cases and recommendations. *Meridian: A middle school computer technologies journal*, *5*(1), 83.

Grant, M. M., & Branch, R. M. (2005). Project-based learning in a middle school: Tracing abilities through the artifacts of learning. *Journal of Research on Technology in Education*, *38*(1), 65-98.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, *42*(1), 38-43.

Gutiérrez, K. D., Morales, P. Z., & Martinez, D. C. (2009). Re-mediating literacy: Culture, difference, and learning for students from nondominant communities. *Review of Research in Education*, *33*(1), 212-245.

Guzdial, M. (1994). Software-realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments*, *4*(1), 001-044.

Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25-27.

Guzdial, M., Ericson, B., Mcklin, T., & Engelman, S. (2014). Georgia computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Transactions on Computing Education (TOCE)*, *14*(2), 13.

Halverson, E. R., & Sheridan, K. (2014). The maker movement in education. *Harvard Educational Review*, *84*(4), 495-504.

Harris, C. J., Penuel, W. R., Angelo, C. M. D., Debarger, A. H., Gallagher, L. P., Kennedy, C. A., … Krajcik, J. S. (2015). Impact of Project-Based Curriculum Materials on Student Learning in Science : Results of a Randomized Controlled Trial. *Journal of Research in*

Science Teaching, 1–24. http://doi.org/10.1002/tea.21263

Honey, M., & Kanter, D. E. (2013). Design, make, play: Growing the next generation of science innovators. In Honey, M., & Kanter, D. E. (Eds.), Design. Make. Play. Growing the next generation of STEM innovators (pp. 1–6). New York, NY: Routledge.

Kafai, Y. B. (2006). Constructionism. In R.K. Sawyer (Ed.), The Cambridge handbook of the learning sciences (pp. 35-46). New York: Cambridge University Press.

Kafai, Y., Fields, D., & Searle, K. (2014). Electronic textiles as disruptive designs: Supporting and challenging maker activities in schools. *Harvard Educational Review*, *84*(4), 532-556.

Kaptelinin, V., & Nardi, B. A. (2006). *Acting with technology: Activity theory and interaction design*. MIT press.

Krajcik, J. S., & Blumenfeld, P. (2006). Project-based learning. In R. K. Sawyer (Ed.), The Cambridge handbook of the learning sciences (pp. 317–334). New York: Cambridge.

Krajcik, J.S., & Czerniak, C., (2013). Teaching Science In Elementary And Middle School Classrooms: A Project-Based Approach, Fourth Edition. Routledge: London.

Krajcik, J., McNeill, K. L., & Reiser, B. J. (2008). Learning-goals-driven design model: Developing curriculum materials that align with national standards and incorporate project-based pedagogy. *Science Education*, *92*(1), 1-32.

Krajcik, J. and Shin, N. (2014) 'Project-Based Learning', in Sawyer, R.K. (ed.) *The Cambridge Handbook of the Learning Sciences*. Cambridge: Cambridge University Press, pp. 275–297. Chapter DOI: http://dx.doi.org/10.1017/CBO9781139519526.018

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: urban youth learning programming with scratch* (Vol. 40, No. 1, pp. 367-371). ACM.

Martin, L. (2015). The promise of the maker movement for education. *Journal of Pre-College*

Engineering Education Research (J-PEER), *5*(1), 4.

Martinez, S. L., & Stager, G. S. (2013). *Invent to learn: Making, tinkering, and engineering in the classroom.* Constructing modern knowledge press.

Marx, R. W., Blumenfeld, P. C., Krajcik, J. S., & Soloway, E. (1997). Enacting project-based science. *The elementary school journal*, *97*(4), 341-358.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B. D., ... & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, *33*(4), 125-180.

National Research Council. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. Washington DC: National Academies Press.

NGSS Lead States. (2013). *Next Generation Science Standards: For States, by States.* Washington DC: National Academies Press.

Noss, R. (1986). Constructing a Conceptual Framework for Elementary Algebra through Logo Programming. Educational Studies in Mathematics, 17, p.335-57.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1-11.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New ideas in psychology*, *2*(2), 137-168.

Penuel, W. R., & Fishman, B. (2012). Large-scale science education intervention research we can use. *Journal of Research in Science Teaching*, *49*, 281–304. http://doi.org/10.1002/tea.21001

Penuel, W. R., Fishman, B., Haugan Cheng, B., & Sabelli, N. (2011). Organizing Research and Development at the Intersection of Learning, Implementation, and Design. *Educational*

Researcher, *40*(October), 331–337. http://doi.org/10.3102/0013189X11421826

Pinkard, N., Erete, S., Martin, C. K., & McKinney de Royston, M. (2017). Digital Youth Divas: Exploring Narrative-Driven Curriculum to Spark Middle School Girls' Interest in Computational Activities. *Journal of the Learning Sciences*, (just-accepted).

Reiser, B. (2014). Designing Coherent Storylines Aligned With NGSS for the K-12: How do we bring practices into K-12 classrooms? Boston, MA: NSELA Conference.

Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265-269). ACM.

Severance, S. (2016). *Teacher and student supports for implementation of the NGSS*. University of Colorado at Boulder: ProQuest Dissertations Publishing.

Sheridan, K., Halverson, E. R., Litts, B., Brahms, L., Jacobs-Priebe, L., & Owens, T. (2014). Learning in the making: A comparative case study of three makerspaces. *Harvard Educational Review*, *84*(4), 505-531.

Spitulnik, M. W., Stratford, S., Krajcik, J., & Soloway, E. (1997). Using technology to support student's artifact construction in science. In K. Tobin (Ed.), International Handbook of Science Education. Netherlands: Kluwer Publishers.

Spitulnik, M. W., Zembal, C., & Krajcik, J. (1998). Using hypermedia to represent student understanding: Science learners and preservice teachers. In G. D. Phye (Ed.), Teaching Science for Understanding: A human constructivist view. San Diego, CA: Academic Press.

Tatar, D. (2007). The Design Tensions Framework. *Human-Computer Interaction*, *22*, 413–451. http://doi.org/10.1080/07370020701638814

Vossoughi, S., Hooper, P. K., & Escudé, M. (2016). Making through the lens of culture and power: Toward transformative visions for educational equity. *Harvard Educational Review*, *86*(2), 206-232.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard university press.

Weisman, S. A., & Gottfredson, D. C. (2001). Attrition from after school programs: Characteristics of students who drop out. *Prevention science*, *2*(3), 201-205.

Welner, K. G., & Carter, P. L. (2013). Achievement gaps arise from opportunity gaps. *Closing the opportunity gap: What America must do to give every child an even chance*, 1-10.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction*, *24*(2), 171-209.

Wilkerson, M., & Fenwick, M. (2017). *Using mathematics and computational thinking*. In C.V. Schwartz, C. Passmore, and B.J. Reiser (Eds.), *Helping students make sense of the world using next generation science and engineering practices* (pp. 181-204), Arlington, VA: National Science Teachers Association.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Wisnudel-Spitulnik, M., Krajcik, J., Soloway, E. (2000). Construction of Models to Promote Scientific Understanding. In Feurzeig, W & Roberts, N. (Eds.), Modeling and Simulations in Science and Mathematics Education, Springer, NY.

Table 1. Comparison of Approaches for Promoting Selected Constructs

	Project-Based Learning (aligned to NGSS)	Making and Makerspaces	Computer Science Education (i.e., coding)
Agency	Provides students with choice in how to construct artifacts and in asking and pursuing questions Provides technological tools and scaffolds to support productive engagement in science practices Can provide opportunities to address authentic problems in students' communities Aims to support students' self-efficacy in science and engineering	Provides students with control over how to go about constructing artifacts Provides students with choice in selection of problems relevant to them Can provide access to resources and expertise of community to support construction of artifact Aims to support students' self-efficacy in science and engineering	Can provide students with choice over the subject and workings of constructed programs or games Can provide tools to support students in coding and computational thinking
Identity	Promotes more authentic engagement in the domains of scientists and engineers Aims to place students on trajectories leading to increased participation in STEM	Promotes the "maker mindset" and becoming a "maker" Aims to place students on trajectories leading to increased participation in STEM	Aims to place students on trajectory ("pipeline") leading to increased participation in computer science
Equity	Implemented at scale across formal school systems to provide all students with meaningful science learning opportunities Promotes science learning as a "cultural accomplishment" within a community of learners Promotes accessibility to science and engineering through practices, particularly via social discourse Can utilize multiple modes of expression over time Can aim for students to address problems relevant to their lives	Variety of students can self-select into makerspaces Aims to have inclusive notion of "making" encompass broader forms of science and engineering Provides hybrid approaches (e.g., cloth with circuits) to allow broader interest points Promotes students leveraging existing interests and experiences	Promotes use of accessible tools that require little background knowledge in coding/programming Can allow students to bring in multiple forms of media seen as relevant to their lives to create digital artifacts
Learning Goals	Aims for students in formal settings to meet Performance Expectations in NGSS Promotes 3-D learning: students explain phenomena and solve design challenges through engaging with SEP to build DCI and CCC	Can provide idiosyncratic, as-needed learning for participants (e.g., "Just in time" instruction)	Can support subject-specific learning in formal settings Can narrowly focus on Computational Thinking