Educational Psychologist

ISSN: 0046-1520 (Print) 1532-6985 (Online) Journal homepage: http://www.tandfonline.com/loi/hedp20

Constructionist Gaming: Understanding the Benefits of Making Games for Learning

Yasmin B. Kafai & Quinn Burke

To cite this article: Yasmin B. Kafai & Quinn Burke (2015) Constructionist Gaming: Understanding the Benefits of Making Games for Learning, Educational Psychologist, 50:4, 313-334, DOI: 10.1080/00461520.2015.1124022

To link to this article: http://dx.doi.org/10.1080/00461520.2015.1124022

Published online: 06 Feb 2016.

EDUCATIONAL PSYCHOLOGIST, *50*(4), 313–334, 2015
Published with license by Taylor & Francis
ISSN: 0046-1520 print / 1532-6985 online
DOI: 10.1080/00461520.2015.1124022

Constructionist Gaming: Understanding the Benefits of Making Games for Learning

Yasmin B. Kafai
Graduate School of Education
University of Pennsylvania

Quinn Burke
School of Education, Health, and Human Performance
College of Charleston

There has been considerable interest in examining the educational potential of playing video games. One crucial element, however, has traditionally been left out of these discussions—namely, children's learning through making their own games. In this article, we review and synthesize 55 studies from the last decade on making games and learning. We found that the majority of studies focused on teaching coding and academic content through game making, and that few studies explicitly examined the roles of collaboration and identity in the game making process. We argue that future discussions of serious gaming ought to be more inclusive of constructionist approaches to realize the full potential of serious gaming. Making games, we contend, not only more genuinely introduces children to a range of technical skills but also better connects them to each other, addressing the persistent issues of access and diversity present in traditional digital gaming cultures.

The launch of the serious gaming movement over a decade ago focused on games that are designed to teach academic content and skills to students playing them (Mayer, 2014). This development followed Gee's (2003) seminal examination of video games in terms of learning and literacy, in which he argued that many good educational principles—36 in total—could be found in the design and play of video games. Hundreds of educational games and simulations have been designed and evaluated to support learning across various domains (Shaffer, 2007; Squire, 2011). Following a report by the National Research Council (2011) on the educational potential of video games, a flurry of reviews quickly followed, each examining learning benefits of serious games. The verdict reached by these meta-analyses is decidedly mixed: Whereas one meta-analysis found significant impact (Wouters, van Nimwegen, von Oostendorp, & van der Spek, 2013), others were more hesitant in their assessment of impact (Girard, Ecalle, & Magant, 2012; Vogel et al., 2013); still others were downright dismissive of the motivation and cognitive benefits claimed by serious gaming (e.g., Young et al., 2012).

So why is there such discrepancy here in terms of learning outcomes? According to a report entitled "Moving Learning Games Forward" from MIT's Education Arcade (Klopfer, Osterweil, & Salen, 2009), the educational effectiveness of digital game play sits squarely on how well the game itself engages the learner. The writers of the report argued that "advocates for game-based learning tend to adopt one of two very different approaches to designing games for formal education" (p. 1). The first group promotes commercial gaming such as *World of Warcraft* and *Civilization* as the ideal and argues that the interactivity and immersiveness of these video games far exceeds schools' capacity to consistently engage young learners in the digital media that increasingly characterizes 21st-century life. The second group, however, generally eschews commercial games and rather focuses on those educational games such as *Word Island* and *Math*

Correspondence should be addressed to Quinn Burke, School of Education, Health, and Human Performance, College of Charleston, 86 Wentworth Street, Charleston, SC 29403. E-mail: burkeqq@cofc.edu

Blaster that serve to reinforce traditional academic content areas, particularly within the areas of mathematics and reading. The report pointed out that while “the first group embraces games and abandons school, this second group often embraces school to the detriment of anything that looks like real gaming” (Klopfer et al., 2009, p. 2).

Video games or school games? Clearly such a divide exists. Ask any child between the ages of 8 and 18 to point to the difference between a commercial and an educational video game, and she or he typically will be able to spot the difference within a few minutes of actual game play. Is the game ultimately about the game, or is it merely a veil for academic content? Is there a narrative, a storyline, that goes beyond the retention of vocabulary words and math equations? These are key questions for educators and game designers alike. They beg another question too: Is there a middle ground between popular gaming and educational gaming? This has been the leading question on which educational game designers and educators alike have focused when it comes to more effectively and widely integrating games into learning environments. But whereas the serious gaming movement has consistently posited that the educational potential of games sits somewhere between commercial products and skill-and-drill exercises and searched for this middle ground, this article argues that the real solution does not sit somewhere between the commercial and the educational but rather might be situated between the practice of *playing* and *making* games, thus combining constructionist and instructionist efforts in serious gaming.

Glaringly absent in the discussions about the effectiveness of serious gaming has been the inclusion of *constructionist* gaming approaches, namely, those approaches in which games are designed by students (rather than professionals) for learning benefits (Kafai, 1995, 2006). Here the divide between gaming for the sake of gaming and gaming for the sake of schooling begins to more effectively break down as children employ academic content knowledge skills such as computer science, mathematics, or arts to create viable games that are intended first and foremost for their peers rather than their teachers. The absence of constructionist gaming in the conversations surrounding serious gaming is surprising given its past successes for not only helping children learning to program but also supporting their learning of academic content and other skills (see Burke & Kafai, 2014; Earp, 2015, Hayes & Games, 2008; Hayes-Gee & Tran, 2015).

Why has there been such glaring omission of the constructionist approach? The first and most obvious reason stems back to the instructionist desire of having a finished, downloadable, teaching product—namely, the game itself—as the party responsible (rather than the instructor) for teaching the child. As Taylor (1980) aptly pointed out in his early analysis of computers in schools, positioning the technology as the “tutor” represents the unspoken default mode of technological integration. In the 1980s, computers were introduced to schools as teaching machines, and this perception of the devices as surrogate instructors persists. Related to this, a second and less inimical reason for constructionist gaming’s general unpopularity in schools may simply be due to the fact that K-12 educators have viewed the endeavor as far too technical, particularly given game making’s association with learning programming. A third and final reason may be that until recently the gaming industry really did not want players to engage in any design or modification of the games they produced for the marketplace. It was their copyrighted product, after all. Whatever the reasons for omission though—educational, technical, or cultural—the situation is now clearly changing.

We are currently witnessing a paradigmatic shift toward constructionist gaming that is propelled by several developments, including the initiative to promote computational thinking (Grover & Pea, 2013; Wing, 2006), a need to broaden participation in computing (National Research Council, 2011), and a wider emergence of a do-it-yourself (DIY) culture among today’s youth (Honey & Kanter, 2013; Knobel & Lankshear, 2010). But the central impetus for a shift might come from the industry and gaming culture at large itself. After all, some of the most popular games on the market today include level and character modding as a central feature (El Nasr & Smith, 2006; Hayes-Gee & Tran, 2015) and encourage such modding as part of game play until the next version becomes available.

This making element of constructionism is not limited to game play itself. A closer examination of gaming cultures reveals that many rich learning activities happen in the context of what Gee (2003) referred to as “metagaming,” in which play extends beyond the game and includes participating in online discussion forums (Steinkuehler & Duncan, 2008) and even accessing and designing cheat sites (Kafai & Fields, 2013) to help players more effectively navigate the game. In the community of many instructional game designers, we also observe a recent shift to include game-making platforms and activities (Klopfer & Haas, 2012). Perhaps the clearest indicator that constructionist gaming has arrived, however, is signaled by the remarkable popularity of *Minecraft* (Garrelts, 2014), a virtual sandbox that counts now more than 100 million paying subscribers playing and making their own games.

In this article, we articulate a framework for understanding the different dimensions of making games for learning based on constructionist theory. We evaluate the educational potential of making games for learning in terms of personal, social, and cultural dimensions informed by constructionist theory (Kafai, 2006; Papert, 1980). Here, personal dimensions refer to the academic and attitudinal outcomes that making games can provide to learners. How does making games affect the way a child perceives digital media, individual academic subject matter, and the wider question of what it is to learn? The social dimension focuses on collaborative arrangements ranging from small groups to larger groups in massive online communities. What does game making rather than just playing offer young leaners in terms of not only collaborative making but also gaining a wider appreciation of designing for an audience? The cultural dimensions focus on how factors such as gender and race have been part of the learning arrangements and/or impacted outcomes. Video games and the wider gaming ecology are contested spaces, and some players—often Caucasian and Asian males—claim priority in determining what qualifies as a good video game and a good player. How does children making their own games undercut such singular authority and allow for multiple perspectives as to what qualifies as good play? Through this framework of the personal, social, and cultural, we make the case for video game making becoming a more integral part of the serious gaming movement. Game making for learning offers an effective means to bring the educational potential of games into formal learning environments, whether it be school-day classrooms, after-school clubs, or summer camps. Our goal is to illustrate how serious gaming can be more inclusive and informative for children by giving young learners a greater hand in the design and production of the video games they love and learn to play.

BACKGROUND

In the 1995 introduction to *Minds in Play*, Seymour Papert postulated,

> Every educator must have felt some envy watching children playing video games: If only that energy could be mobilized in the service of learning something that the educator values. . . . The Constructionist mind is revealed when the wish leads to imagining children making the games instead of just playing them. Rather than wanting games to instruct children they yearn to see children construct games. (p. ii)

Of course, 15 years earlier, when Papert (1980) wrote *Mindstorms*, the term "constructionism" had yet to be coined. Papert (1991) later defined it this way:

> Constructionism—the N Word as opposed to the V word—shares Constructivism's connotation to learning as building knowledge structures irrespective of the circumstances of learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity whether it's a sand castle on the beach or a theory of the universe that can be shared with others. (p. 11)

Making games is a prime example of a constructionist activity, in particular in the age of digital play. But it even harks back to Piaget's (1951) work that examined the developmental function of game play as a venue for children to develop and exercise their understanding of rules and considered games of construction to be the highest form of game play, as games require children to build representations of the world according to their understanding (Kafai, 1995, 1998). The key idea here is that knowledge about the rules, worlds, and interactions is represented in a public entity—the game—and that playing and making games highlights the personal, social, and cultural dimensions of constructionist learning.

A primary focus of constructionism examines learning from a personal perspective, very much in the Piagetian tradition. Papert saw the engagement with Logo programming as a way to facilitate the construction of knowledge structures and what he termed "appropriation" so that learners could make knowledge their own and begin to personally identify with it. Programming the Logo turtle in the context of a game very much makes the construct an "object-to-think-with" (Papert, 1980) linking together artifacts in the physical world (in this case, a turtle) with those representations (in this case, the rules and objects) in the mind. Papert argued that objects-to-think-with such as the Logo turtle are particularly effective at supporting appropriation because they facilitate the learner's personal identification with the object and help to construct, examine, and revise connections between old and new knowledge. By designing a game (or, on a more granular level, its procedures, algorithms, and data structures), the personal knowledge becomes public and can then be shared with others. Of course, thinking about game programs as personal objects that can be shared widely as public entities articulates a phenomenon entirely akin to the growth of Internet culture, which too is built upon the amassment of intimately personal items (e.g., photos, stories, and designs) introduced on an equally massive wider public sphere, which then takes on entirely new meanings upon this wider scale. And it connects nicely to the social dimension of constructionist gaming.

The social dimensions have focused on learning communities, girded by the understanding that personal construction of programs, games, and other artifacts does not happen in a vacuum but very much in a social context. Papert's (1980) firsthand experience researching Brazilian samba schools encapsulated his sense of social norms and interactions as pivotal to any form of learning:

> These are not schools as we know them; they are social clubs with memberships that may range from a few hundred to many thousands. Each club owns a building, a place for dancing and getting together. . . . During the year each samba school chooses its theme for the next carnival, the stars are selected, the lyrics are written and rewritten, and the dance is choreographed and practiced. Members of the school range in age from children to grandparents and in

> ability from novice to professional. But they dance together and as they dance everyone is learning and teaching as well as dancing. Even the stars are there to learn their difficult parts. (p. 178)

In many ways the proposal of samba schools as an example of learning culture predates the move away from traditional schools to apprenticeship learning that was ushered in years later by Lave and Wenger's (1991) seminal work and then again in the form of affinity cultures by Gee (2003) in the context of gaming communities.

Finally, cultural dimensions have focused on the social dynamics and politics that determine how one way of knowing is valued over others. Here, the serious gaming movement itself provides a prime example of how gaming has first been dismissed but now is seen as an important tool for learning within the education community. Early research saw little promise in on digital games choosing rather to highlight its perils in terms of violence (Provenzo, 1991), whereas current research focuses more on the potential of games for learning (Gee, 2003; Squire, 2010). It is not just the context but also the approaches in how we design that reflect values and provide a context for learners to connect and engage with the practices in the field. For instance, by studying and interviewing programmers, Turkle and Papert (1991) revealed that the officially promoted "top-down" planning approach is not always superior to a more improvised, bricoleur-like approach. The bricoleur style is not a stepping stone toward more advanced abstract forms of knowledge construction but rather represents a qualitatively different way of organizing one's planning and problem solving. This valuation of the concrete is quite evident today in the growing interest that youth display for making and modding their own games.

In our approach to serious gaming, we build on these foundations of constructionist theory to understand how and what students can learn in the process of designing and making games through computer programming. We have coined the term "computational participation" to capture these different dimensions in which designing and implementing systems are not solely the function of algorithmic thinking but more fundamentally representative of the practices and perspectives necessary to contribute within wider social networks and understand the cultural and social nature of a networked society (Kafai & Burke, 2014). Whereas other developments situate game making in several different fields such as new media literacies (Gee, 2010), system-based thinking (Salen, 2007), and critical engagement with media (Buckingham & Burn, 2007; Pelletier, 2008), we draw on the broader notion of participatory culture informed by Jenkins's (2006) and Ito and colleagues' (2009) work. That is, children's capacity to create and modify digital games with and for each other offers them a tremendous advantage in understanding the ever-changing nature of digital media, public domain, and what it means to problem solve and participate. Likewise, we position video game making as an unique, early channel for children to comprehend the social, economic, and civil power of "making" and "sharing" (Grimes & Fields, 2015).

Foremost though, coding has received by far the most attention because it can include various software design practices ranging from programming, debugging, and remixing code. Taken together, these practices capture what has been described as "computational thinking," which Wing (2006) defined as designing systems for more effective problem solving with computers. Although computational thinking is not just coding, code represents one of the key avenues to engage youth in an early understanding about how effective systems are designed and maintained, a skill set that can be applied to fields as diverse as industrial mechanics, computational biology, and marketing analytics. Understanding game design is an optimal early incubator for grasping computational thinking, as would-be designers not only have to create a series of novel user interfaces but also need to ensure that these interfaces scale in complexity and even adjust to the player's capacity to accomplished digitally designed tasks. To help beginning designers to accomplish complex programming tasks, a whole array of tools and languages exist (for an overview, see Burke & Kafai, 2014).

A helpful way to operationalize the computational thinking involved in programming activities has been proposed by Brennan and Resnick (2012), who distinguished between computational concepts, practices, and perspectives. *Computational concepts* refer to elements such as sequences, loops, parallelism, events, conditionals, operators, and data structures that are present in many programming languages. To accomplish these designs, student designers had to engage in *computational practices* that are activities such as being incremental, reusing and remixing, testing and debugging, and modularizing and abstracting. Finally, *computational perspectives* such as expressing, connecting, and questioning refer to worldviews that designers develop as they engage with digital media. Coding in the context of constructionist gaming thus is not just learned for the sake of understanding and generating code, it also demands that designers be aware of perspectives other than their own and thus provides a rich context for learning other academic content, the next dimension.

In the following sections we use these personal, social, and cultural tenets of constructionist theory to frame a synthesis of learning benefits identified in various research studies. We see the first category of learning, the personal dimension, focus on knowledge appropriation and transformation that is instantiated in the process of making games. First and foremost, these benefits focus on learning programming, but they also include other academic content and skills such as various subject matters and problem-solving skills (Grover & Pea, 2013). Second, the social benefits focus on the various forms of collaboration involved in making games for learning, whether it is working with others on the design or it is about sharing and exchanging designs (Grimes & Fields, 2015). Finally, the third dimension examines the cultural benefits and boundaries that circumscribe participation in digital-game-making activities (Margolis, Estrella, Goode, Holme, & Nao, 2008). As we have noted before, the benefits of making games in terms of learning and connecting are by no means exclusive to each other, and the three dimensions that follow all overlap and mutually inform each other (Burke & Kafai, 2014). Each represents a distinct takeaway and connects to larger research efforts, but the collection as a whole represents a more comprehensive skill set that can emerge only from the making and sharing of one's games.

APPROACH

Between 2014 and 2015, we conducted several comprehensive surveys of published papers on the topic of children's learning while making games. We searched ERIC, as well as used Google scholar and backwards searches that referenced Kafai's (1995) *Minds in Play* book publication as a starting point, given its prominence as the first account of children making their own video games in a formal educational environment. We also searched journal archives such as *Computers & Education, Games & Culture*, and *International Journal of Learning and Media*, in which a number of papers had been published. In addition, we searched conference archives from *Digital Games Research Association* and the *Games, Learning and Society*, as well as the ACM Digital Library, which serves as repository for conferences such as *Foundations of Digital Games*; *Interaction Design and Children*; and *ACM Technical Symposium on Computer Science Education*, with key phrases such as "children and game making" and "children and programming games" to locate articles that focused on tool development for game making and reported on findings within an usability rather than educational evaluation. For that reason, a considerable number of articles included in our review were conference publications, especially in technical fields.

We included only distinct studies and privileged those with more recent publication dates (the past decade) that often provided more comprehensive and extensive study outcomes based on the following four selection criteria: (a) grade-level academic content: students engaged in game making on the K-12 level; (b) programming: the making of games involved computational concepts and practices involved with writing and debugging code; (c) outcomes: papers needed to provide learning and attitudinal outcomes in form of quantitative or qualitative data, or both; and (d) designated learning environments: whether an in-school class, an after-school program, or a summer workshop, the studies we selected ultimately involved children making games in a structured learning environment rather than on their own. We also included qualitative studies that focused on cases of individual students making games. We excluded articles that were essays on game making or did not include any information on study design or outcomes.

We identified 55 articles suitable for a review and published in English, though a number of studies have been conducted outside of the United States, mostly in Europe. A table with all the studies considered in this article is in the appendix and provides an overview of number of students involved, demographic information (age, gender, race—when available), duration of activities, programming tool, and learning focus and outcomes (classified into computational concepts, practices, and perspectives, as well as content and learning strategies). Studies used various programming tools to have students make games (Burke & Kafai, 2014), addressed often multiple outcomes, and chose to assess and evaluate them with different methods. Studies used a wide range of tools from surveys to interviews, and tests, as well as case studies and project analyses, often combining them though experimental (e.g., Vos et al., 2011) or quasi-experimental (e.g., Kafai, 1995; Owston et al., 2009; Yang & Chang, 2013) designs were rare. The articles included in this analysis addressed a multiple number of all these aforementioned outcomes and influencing factors, such that a single study could simultaneously evaluate middle schoolers' learning to code through rudimentary game making while assessing the potential role of gender in the games the students created.

BENEFITS OF CONSTRUCTIONIST GAMING

Of the 55 studies included in our review, the largest number has and continues to focus on learning programming, followed by other academic subjects. Far fewer studies have focused on the equally important social and cultural dimensions of constructionist gaming. Because of their overall importance for learning, we report on all three dimensions but allot the most space for discussing learning outcomes related to the personal dimension of learning code and academic subject content. Of the studies, 44% focused on children developing computational strategies to problem solve, followed by 34% explicitly dealing with children learning computational concepts, whereas 27% examined shifts in learners' perspectives about the nature of programming and computer science. Furthermore, 34% examined children's own sense of learning as a particular process (e.g., "learning about learning"), whereas 16% of studies investigated children learning particular subject matter content (e.g., mathematics, science, or language arts) through the game-making process. Half of the studies took place out of schools (e.g., 23% after-school clubs, 18% summer camps, 5% community technology centers, and 4% online communities), and the other half occurred during the school day, with only a few (5%) running in combination with an after-school program.

Collectively more than 9,000 youth participated in the studies reviewed for this article; in some instances, dozens of schools were involved but not all students participated in surveys. The largest group of studies (53%) focused on middle school students, followed by 20% focused on high school students, 9% focused on elementary school students, and 18% of the studies including multiple school levels. More than half of studies (57%) included both female and male students as their participants, with three studies being girls only and one study boys only. More than one third of studies (36%) did not provide any information on gender. In terms of diversity of participants, these numbers were even more difficult to come by, with more than half (52%) of projects not providing any information regarding ethnic background. Of those that reported demographic information, studies had diverse participants, some of which targeted particular groups such as young African American males (DiSalvo, Guzdial, Bruckman & McKlin, 2014), Latina girls (Denner, Werner, & Ortiz, 2012), or Native American youth (Lansmann & Lewis, 2011). In the following sections, we review and discuss in more detail how these outcomes relate to the personal, social, and cultural dimensions of constructionist gaming.

Personal Dimensions

Learning coding. A substantial body of research has now examined learning in the context of game-making activities. Leading the way are numerous studies in which students in schools have designed games that are focused on learning programming concepts. In the original game-making project (Kafai, 1995), a class of fourth-grade students who programmed fraction games for younger students in their school learned about key computational concepts such as loops, conditionals, and even tail recursion—a procedure they used to structure the question-and-answer dialogue of their multiple-choice problems. They also improved significantly in computational practices such as writing and debugging programs when compared to students who were learning Logo programming solely in the context of smaller independent projects unrelated to gaming. Such comparative evaluations to assess and evaluate students' learning when making games are still rare but helped to tease out the effects of context (games vs. unrelated smaller programs) and time (weekly vs. daily programming) spend on programming activities. This study also confirmed the findings of Palumbo's (1990) review that revealed that the context, complexity, and time frame in which beginners encounter programming matter considerably in their capacity to learn to code.

Further support for learning programming concepts has been collected in an examination of the 221 games created by 325 middle school students using Storytelling Alice in classes and after-school clubs (Werner, Denner & Campe, 2014). The analyses revealed not only the use of simple programming constructs but also more complex constructs such as student-created abstractions, concurrent execution, and event handling—all indicative of higher order patterns. Likewise, a review of 108 games created by 59 middle school girls with Stagecast Creator showed the use of key computational concepts such as loops, variables, and conditionals but only moderate usability and low levels of code organization and documentation (Denner et al., 2012). The Globaloria online platform, through which thousands of students design video games as part of curricular activities in their schools or clubs, demonstrated learning of key programming concepts using Flash (Reynolds & Harel Caperton, 2011).

Numerous other studies have documented students' learning of computational concepts (see Al-Bow et al., 2009; Basawapatna, Koh, Repenning, Webb, & Marshall, 2011; Holbert & Wilensky, 2014; Howland & Good, 2015; Owston, Wideman, Ronda, & Brown 2009; Pelletier, Burns & Buckingham, 2010; Repenning et al., 2015; Robertson, 2012; Robertson & Howells, 2008; Seaborn, El-Nasr, Milam, & Young, 2012). Although the learning of computational concepts can appear heavily scaffolded in these contexts by providing students with scripts that can be remixed and highly prescriptive game contexts (e.g., remix the classic *Pong* or *Ms. Pac-Man*, or make your own platform game), these scaffolds likewise need to be recognized as often necessary first steps to bring children into the game-making process. Some critics (Atwood, 2012; Cuban, 2014) have decried the recent resurgence of "coding for all" in schools approach as unrealistic and an all-too-simplified version of what "real" programming actually is. But to these critics, we see these game-making activities providing a low threshold entrance into the world of programming—and a threshold that has been utterly lacking in schools for the past 30 years despite the overwhelming economic and literacy needs to develop children with a greater appreciation of how computers actually function. We have made the case elsewhere that such forms of remixing are authentic forms of programming practices in the larger community and, in fact, offer introductory computer science instruction a valid pedagogical approach to reach children it never could reach before.

But learning computational concepts didn't just happen in school classrooms where students have access to computers and teacher support. A 2-year study in a Los Angeles Computer Clubhouse, part of a worldwide network of community technology centers (Kafai, Peppler, & Chapman, 2009) that encourage creative uses of technology, found that use of programming concepts such as loops and conditionals significantly increased from Year 1 to Year 2 as youth developed and remixed video games for themselves and each other. In examining an archive of more than 500 Scratch gaming programming projects generated and saved during this time (Kafai & Peppler, 2011), the study also revealed that some concepts such as variables and randomization did not appear frequently in Clubhouse members' projects, an important indicator that some concepts are more easily introduced to beginners whereas others require more instructional support. In fact, many game-making projects have taken place in after-school clubs or community centers because programming was not part of the standard curriculum in many schools (Adams & Webster, 2012; K. Clark & Sheridan, 2010; DeLay et al., 2013; Denner et al., 2012; Fadjo, Hallman, Harris, & Black 2009; Javidi & Sheybani, 2010; Kafai & Peppler, 2012; Mouza, Pan, Pollock, Atlas, & Harvey, 2014; Peppler & Kafai, 2007).

Of particular interest here is how making game projects compares to programming other projects, such as music videos or stories, in terms of engaging with computational concepts. A study focused on games, videos, or stories programmed by 322 youth using Alice or Scratch in summer camps found that youth making games used the most variables, loops, and if-statements in their programs (Adams & Webster, 2012). These findings of game design inviting the use of particular computational concepts has also been observed in other studies and led some researchers to design programming tools for story games that overcome this shortcoming (Howland & Good, 2015), as stories, by their very nature as established narratives, allow for less variability and conditionality in coding sequences (Burke & Kafai, in press). Conducting work in after-school settings faces additional challenges in that members' attendance is not usually mandatory and thus much more likely to fluctuate; furthermore, the use of standard assessments such as pre- and postexams is not typically an option, which itself may explain why many of the research studies resort to examine the game programs themselves as an indicator of youth learning computational concepts.

We also found compelling evidence that youth engage in various computational practices as they are making games. In the fraction game design project, case studies documented how students debugged, revised, and tested their games over and over again, especially after the periodical user evaluation sessions they conducted with younger students, over a 6-month period. Moreover, posttests revealed game design students' significantly higher performances in designing and debugging given Logo code when compared to students in control classes (Kafai, 1995). More than 10,000 students have used Agentsheets, a programming environment that students and youth from diverse backgrounds use to make simulation games (e.g., Repenning et al. 2015). Based on an analysis of 268 games made by 30 college students in a semester-long course and 73 games made by 33 middle school in an 8-week-long class, an analysis of the comprehensive skill score of computational thinking patterns revealed that over time (in this case a sequence of different game designs that students were asked to make in the course), both groups improved in their performance. As expected, the improvement was more substantial for the college undergraduates than for the middle school students because they not only came with more experience but also spent more time on their games. Numerous other studies have examined learning of computational practices out-of-school (see also Akcaoglu & Kohler, 2014; Baytak & Land, 2010; Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010; Denner et al., 2012; DiSalvo et al., 2014; Esper, Foster, & Grisowld, 2013; Fajdo, Hallman, Harris, & Black, 2009; Holbert & Wilensky, 2014; Howland & Good 2015; Pelletier, Burns, & Buckingham, 2010; Robertson, 2012; Robertson & Howells, 2008; Werner, Campe, & Denner, 2012; Werner & Denner, 2009; Werner, Denner, Campe, & Kawamato, 2012).

Although school-based programs can provide more scaffolded introductions to computational thinking practices in outlining a sequence of game designs (such as with Agentsheets) or offering instructional support by having a teacher present, we also found evidence of these computational practices in after-school programs. For instance, the case study of 15-year-old Jorge well captures the potential for young game designers to not only employ sophisticated computational concepts and reaching the high ceiling of effective programming but also use remixing meticulously to re-create popular media through seamless imitation (Peppler & Kafai, 2007). Jorge, a regular visitor to the Clubhouse over the 8 months of the ethnographic study, created a video game entitled *Metal Slug Hell Zone X*, a play-off of the popular "run and gun" video game series *Metal Slug*. His most significant challenges were revising his code in order to make it more efficient to re-create the intuitive and fluidity of movement and feedback characteristic of the original game. Although Jorge's case was clearly the exception within the larger Clubhouse youth population that we examined over 2 years, a 4-year study of an urban informal education program in which more than 400 youth participated in designing 2D games captured related aspects by examining the agency that youth achieved by moving from student to assistant and then designer and implementer of instruction (Sheridan, Clark, & Williams, 2013).

Finally, young game designers also expand their computational perspectives. Two recent studies around game design focused on how active, productive engagement with digital media actually shifts students' attitudes toward computing and opens up the possibility of computing as a career (Repenning, 2013; Ryoo et al, 2013). Likewise, participants in the urban after-school game design program, nearly all of them male African Americans, had increased awareness of higher education and career pathways (K. Clark & Sheridan, 2010). These findings of expanding students' interests and perspectives in computing have been confirmed in other studies (DiSalvo et al., 2014; Javidi & Sheybani, 2010; Lakanen, Isomöttönen, & Lappalainen, 2014; Mouza et al., 2014; Robertson, 2013; Vattel & Riconscente, 2012; Webb, Repenning, & Koh 2012). A notable exception is the recent study by Robertson (2013), who found that female students using Adventure Author to make games did not inspire girls' interest in science, technology, engineering, and mathematics (STEM) careers—an aspect that we discuss in more detail in a later section.

Making games for learning not only increases perspectives on computing but also, equally important, changes students' attitudes toward the goals of learning, allowing children to better grasp the long-term benefits of computing and digital design in terms of a potential career pathway. These findings also reinforce the wider body of research (e.g., Moreno & Mayer, 2005; Sun & Rueda, 2012) that collectively points to the commonsense (but nonetheless routinely ignored) precept that students' sense of confidence with digital technology is inextricably tied to the actual activities in which they are engaged, with children at lower income and predominantly minority schools (see Margolis, Estrella, Goode, Holme, & Nao, 2008; Warschauer & Matuchniak, 2010) receiving considerably less access to and engagement with computing activities that focus on actual creation of digital content. It might also be one of the reasons why so few researchers have examined motivation beyond classroom learning and rather turned to examine career aspirations as a result of making games for learning.

Learning content. Like the instructionist counterpart, constructionist gaming also focuses on learning academic content such as mathematics, science, and the language arts prevalent in K-12 curriculum. One could consider the computational concepts, practices, and perceptions reviewed in the previous section to be part of computer science that is now becoming again a part of the standard curriculum. In fact, in the original conception of constructionist gaming, learning of coding and other content were seen as mutually beneficial to each other engaging in not only personal expression but also knowledge transformation. In the early 1980s, no small part of Papert's success in introducing the then foreign concept of computer programming to K-12 schools came from his use of the grounded or practical approach to explain code as a way to make mathematics more tangible and real to students (1980). Papert's metaphor of grounded math influenced the work of a number of other leading computer scientists and educators (see also Abelson & diSessa, 1980; Wilensky, 1995) at that time, who likewise employed the metaphor to explain code as mathematical proofs and "made math." A series of studies has examined content-related learning of mathematics, science, or the arts in the context of game-making activities.

Leading the way are studies where students have designed educational software games or simulations that are connected or even integrated into the curriculum in their schools (Baytak & Land, 2010; Hwang, Hung, & Chen, 2014; Kafai, 1995; Khalili, Sheridan, Williams, Clark, & Stegman, 2011; Schanzer, Fisler, Krishnamurthi, & Feilman, 2015; Vattel & Riconscente, 2012). Returning for a moment to the original game-making project, fourth-grade students not only programmed games but explicitly focused these on teaching fractions to younger students in their school—a topic that they also covered in their math class at the very same time (Kafai, 1995). Again, the posttests revealed that students became not only significantly more proficient in programming Logo when compared to students in the same school who learned Logo in a computer lab in small, disconnected programming activities but also significantly better at understanding and representing fractions measured in pre–post tests. More recent studies of making math games in Scratch confirmed these findings and found that students activated their everyday mathematical experiences and understanding (Ke, 2014). Further research has focused on integrating coding with other STEM topics such as astronomy (Kafai, 1998), which was the focus of games designed by Rosemary and her elementary classmates at Project Headlight from the chapter's introductory vignette or biology, which provided to be a significant boost to 32 seventh-grade students' content understanding and critical thinking when compared to a control class of students who did not design games (Yang & Chang, 2013).

Moving beyond traditional STEM content, game-making activities have also connected to literacy studies (Buckingham & Burns, 2007; see also Pelletier, 2008), the arts and language arts (Howland & Good 2015; Owston, Wideman, Ronda, & Brown, 2009; Robertson, 2012, 2013; Robertson & Howells 2008). In a comparative game-making study, researchers found that students in an experimental group demonstrated significantly better logical sentence construction skills in addition to showcasing better content retention, ability to compare and contrast information resources, and better integration of digital resources (Owston et al., 2009). An analysis of more than 500 Scratch projects found that the games indeed showcased the kind of idea generation and appreciation connected to the arts (Kafai & Peppler, 2012). In observing creative practices as they pertain to constructionist gaming, young designers learn about and appreciate artistic principles by making artistic choices within a single modality (e.g., visual, audio, or kinesthetic), as well as by connecting multimodal sign systems across two or more modalities (e.g., visual and sound, visual and movement or gesture, and sound and movement) to convey an artistic idea (Peppler, 2013). There are also many examples that connect game making to language arts, but the most extensive research to date conducted by Robertson (2013) has implemented game design with more than 900 students in dozens of primary and secondary school across the United Kingdom. She found that students using Adventure Author for making their games improved in their understanding of coding but that game design did not inspire girls' interest in STEM careers. This focus on storytelling has always been found a bonus with girls, and although previous studies suggested that this did not lead to as complex programming (Adams & Webster, 2012), a recent study suggests that given the right tools, even storytelling can become a fruitful context for designing more complex code (Howland & Good, 2015).

These connections of content learning to game making also illustrate the potential of curriculum integration. Rather than conceptualizing game-making activities solely as a context for learning programming and software design skills (as discussed in a previous section) and thus linking them to computer science, here learning coding is situated within a broader context of application development. After all, applications design can focus on content design, in which case designers need to learn not only about the content and skills to be included but also about coding at the same time. Some studies have observed that when student game designers are charged with this dual focus of learning content and coding, the game world and story crafting takes precedence over engagement with content (Ke, 2014). Such are not unexpected challenges that can be addressed by providing instructional scaffolds that provide better integration of content and game (Kafai, Ching, Franke & Shih, 1998). These challenges and disconnects between game mechanics and content are not new; in fact, they can be found in many instructional games that only showcase superficial or extrinsic integration of game and content (Squire, 2007).

Learning about learning. Finally, there is another important learning benefit in game making that goes beyond learning coding and content: the idea of children learning about their own thinking and learning, also called *reflection* or *metacognition*. Papert 1980 saw this a direct corollary to programming and claimed that children learn to articulate procedures, recognize repetition, and “debug” their own thinking when programs don’t run as expected: “But thinking about learning by analogy with developing a program is a powerful and accessible way to get started on becoming more articulate about one’s debugging strategies and more deliberate about improving them” (p. 23). Indeed, in one of the few experimental studies that pitched playing versus making games, education researchers Vos, van der Meijden, and Denessen (2011) found that students who engaged in making a game that the other group of student just played demonstrated significantly deeper engagement in their learning and strategy use, which involved system analysis, decision making, and troubleshooting. Of course this comparative study analyzes only students on the elementary level and is far from definitive in its examination of confidence, motivation, and content acquisition based upon the playing versus making paradigm. But it certainly points out that making games requires distinct ownership over the content and such ownership both requires and can instill a certain level of confidence in the learner.

A study comparing two summer camp groups indicated that the group involved in game making also produced measurable improvements in problem solving (Akcaoglu, 2014; Akcaoglu & Koehler, 2014). Whereas 20 students (the experimental group) learned problem-solving skills through designing and testing their own video game using Microsoft Kodu, 24 students (the control group) simply practiced their problem-solving skills by playing already-created games in Kodu. At the end of the intervention, students who designed their own video games significantly outperformed students on the validated assessment called the Program for International Student Assessment in terms of 19 questions related to the three problem types: system analysis and design, troubleshooting, and decision making. These meta-dimensions of learning in game making have also garnered the attention of other education researchers who are interested in games as learning environments. Although coding and content capture the more easily recognized knowledge and skills addressed with making games, design, problem solving, or system thinking skills, assessment-wise, also fall into a broader overarching category that we called learning about learning. Numerous other studies have used game making to examine this learning (Allsop, 2015; DeLay et al. 2013; Fristoe, Denner, Mateas, MacLaurin, & Wardrip-Fun, 2011; Games & Kane, 2011; Hwang, Hung, & Chen, 2014; Ke, 2014; Khalili, Sheridan, Williams, Clark, & Stegman, 2011; Navarete, 2013; Owston, Wideman Ronda, & Brown, 2009; Pelletier, 2008; Reynolds & Chiu, 2015; Reynolds & Harel Caperton, 2011; Robertson & Howells, 2008; Sheridan, Clark, & Williams, 2013; Sprung, Zimmermann, Nischelschwitzer, Strohmaier & Schadenbauer, 2011).

Making games requires designers to think about a meta-structure in which the game mechanics, interactions, and content are to be embedded. In particular, the notion of system thinking has received much attention, perhaps because of the growing interest in using complex system thinking as a framework to approach science learning and the notion of computational thinking as designing system. To capture the synthetic and analytical nature of design or system thinking, some work has also referred to these skills as what Games (2010) called the “designer mindset.” Studies have investigated these design skills in the context of game making. These design skills are also present in context where students do not use a programming language but rather a scripting context or design tool. Most prominent here is the work on GameStar Mechanic (Salen, 2007), an environment that was specifically developed for kids to make and share game designs that can be fixed by others—unlike the programming languages that can be used to program many things, games being just one app. Although we can argue to what extent GameStar Mechanic engages makers in some form of programming, it is clear that GameStar Mechanic engages students in explicit design and system thinking. It is perhaps here that the distinction between programming and design/system thinking becomes the clearest: Programming engages learners into making a system, whereas GameStar Mechanic involves learners into using a system (Games, 2010). Notwithstanding such internal debates, the design/system thinking category highlights that in making games students also develop representational, or structural, competencies that are not tied directly to code alone.

Social Dimensions

The social dimensions of constructionist gaming examine the collaborations and communities in which game making can take on various forms, ranging from small-scale collaborative programming in pairs to involving classes in schools and districts, setting up national competitions, and engaging online communities with thousands of programmers. However, few of these different social designs have been the focus of extensive and comparative research such as the studies on pair programming designed by Denner and Werner (2007) that, inspired by the success of pair programming on the college level, brought the paired approach to younger students' game making. In fact, most studies in this section present models of different collaborative arrangements and illustrate their applications in a test case but do not provide the detailed analysis of academic learning outcomes listed in the previous section. We suspect that some of this focus on measuring learning outcomes solely in terms of academic content and technical skill stems from researchers' attempts to legitimize game making as worthy of schools' attention. The social affordances of children making video games for themselves and their peers is decidedly less of a metric of evaluation, which is unfortunate given schools mixed record in facilitating meaningful group work and peer collaboration.

An exception is Werner and Denner's research with children using pair programming activities to help support children's making of games with the program Alice and its offshoot Storytelling Alice (Denner & Werner, 2007). Originally implemented among college-level students as a learning technique (Williams & Kessler, 2002), pair programming—sometimes referred to as "peer programing"—is rooted in the belief that learning is an inherently social activity. Working in pairs at a single computer, students code together with one student taking the role of "driver" and generating the code while the other student takes the role of "navigator" reviewing each line of code for accuracy. Denner and Werner (2007) took the pair-programming premise and found that college students were not the only ones who could benefit from such an approach. Working with 126 middle school girls (ages 10–14) over the course of a summer program entitled "Girls Creating Games," they found that the girls were not only more successful in their capacity to program their own games in Flash but also significantly more able to articulate when they had found a problem and then able to use their partner to help debug the issue. This, in turn, made the girls more likely to persist in programming before asking the instructor for external help or even giving up altogether. These results here were subsequently mirrored 2 years later with middle school girls programming their own video games using Storytelling Alice rather than Flash (Werner, Denner, Bliesner, & Rex, 2009).

Many of the game design projects discussed in the previous sections have students individually design and program a game while leveraging the presence of other class members to serve as the audience and informally play these games. Working with fourth graders from nine public elementary schools in Ontario, Canada (18 classes total), Owston and colleagues (2009) found not only that the children were motivated to create quiz-based video games for the sake of their peers playing them but also that their spelling, grammar, and punctuation in devising such questions was significantly improved for the sake of their peers being able to effectively read (and play) the game as it was intended. These findings from pair programming and interactions studies confirm earlier work on the importance of peer pedagogy and apprenticeship (Ching & Kafai, 2008) in learning coding and project design. Thus students' opportunities to develop such collaborative and coding skills and grow more independent were delayed in the early weeks of the study because the range of their activities was so delineated by grade-level seniority.

These success of learning both coding and content through peer-to-peer collaborative game making has inspired the integration of such activities into the regular curriculum to engage not just whole classes but also connections across districts. Using industry-employed languages such as Flash and ActionScipt, Globaloria is national game-making curriculum that, to date, has supported more than 8,000 middle and high students from across 14 states to collaboratively design their own school-made video games as part of curricular activities at their schools (Reynolds & Harel Caperton, 2011). Results are promising in not only getting students more effectively collaborating around personally meaningful projects but also increasing students' self-efficacy with digital technology. Reynolds and Chiu (2015) studied two classes of middle school students (sixth and seventh grade, respectively), as well as two groups of middle school students at local Boys & Girls Clubs, using Globaloria activities over the course of 1 year. They found that participants' sense of self-efficacy improved across all four settings based on pre-and postintervention surveys. Interestingly, young game makers who reported their parents had less post-secondary education reported increased rates of self-efficacy than those participants who came from homes where one or both parents had attended college. Likewise, participants from the formal, school-based settings reported significantly more gains in self-efficacy than those within the informal, after-school club environment. Certainly, developing a working knowledge of Flash and ActionScript was a key component in boosting children's confidence here. But even more instrumental was the fact that the students had developed content that had meaning and importance outside of school. Sponsoring an annual "Globey's Award" across multiple state and grade levels for top game prototypes and presentations, Globaloria facilitates this spirit of collaboration by interjecting a spirit of competition among young game makers.

Driving peer-to-peer collaboration through this competitive spirit is indeed becoming more common. Bridging online and school-based game-making efforts, the White House has sponsored the STEM National Video Game Challenge (www.stemchallenge.org) for 5 years running. Encouraging children to design their own educational video games using a range of free programs (e.g., Scratch, Kodu, Stagecast, Gamestar Mechanic, among others), the STEM Video Game Challenge is not prescriptive in terms of the tools but rather focuses on rewarding those young creators who teach a crucial concept through engaging game play. Every year, 15 middle school and high school students from around the country are selected as winners, and the number of entries has grown from a modest 600 in 2011 to more than 4,000 this past year. Having participated in the inaugural challenge in 2011 with a class of 17 middle school students using Scratch, we found in postinterviews and surveys that the social aspect figured most prominently as the impetus for the participants to persist in completing their video games—even when they hit considerable walls in terms of coding and design (Kafai, Burke, & Mote, 2012). In postclass surveys as well as in follow up interviews, participants reported that more than academic grades and more than potentially placing as a finalist in the Challenge, they were motivated by the approval of their peers during the final session in which the entire school was invited into the classroom to play and offer feedback on their video games. Many online communities have followed the lead of the STEM National Video Game Challenge and begun to engage their members by regularly issuing community-based challenges and competitions around game making. The Scratch website (www.scratch.mit.edu) issues annual "collaborative challenges" and "collab camps" (Kafai & Burke, 2014; Kafai, Fields, Roque, Burke, & Monroy-Hernandez, 2012) as does Microsoft's Kodu site with the "Kodu Cup." Although each site has its own rules and regulations for the respective competitions, all the competitions foster the collaborative spirit by encouraging their challengers to post their ongoing projects for feedback from their peers and utilize discussion boards and forums to search out fellow team members and solicit advice on the game-making process.

Of course, students are drawn to these collaborations even without the impetus of external competitions. Outside of schools, game-making activities are a driving force in many online communities. This is well illustrated with Aragon and colleagues' (2009) analysis of the collaborations at the Scratch website. Through a series of case studies, their analysis points out that one of the primary reasons that children are drawn to Scratch as a tool is the potential to find like-minded game makers at the Scratch website. They highlighted one particular collaboration entitled "Gray Bear Productions." Described as a "company," Gray Bear was founded in 2008 by three young game makers—ages 8, 13, and 15; they soon created a video game called *Pearl Harbor*, which functioned like a digital version of the classic *Battleship*. The sophistication of the game's graphics and the ease of game play attracted hundreds of views and downloads on its initial release. Multiple remixes of the project soon followed. As Gray Bear Productions explains on its website, "We had a lot of people who wanted to join us," and membership jumped to 18. Soon Gray Bear Productions created games such as *Forest Frenzy* and *A Night at Dreary Castle*, which reflected the designers' growing sophistication in creating graphics, plotlines, and game play. *Forest Frenzy* had 19 versions over multiple months before a final glitch-free version was completed. Luther and Bruckman (2008) investigated how such collaborations form and what keeps them together, as quite frequently members have never once met in person and must solely rely on web-based interactions. Their research focused on the Newgrounds website (www.newgrounds.com), a site that like Scratch hosts self-generated content in terms of video games, animations, artwork, and music. They found that these collaborations are rarely successful. Those that successfully created a game and/or animation were a rarity—less than 20% of collaborators actually complete a project.

Gee (2008) explained that collaborative games often mimic professional practices, and these types of games "already give us a good indication that even young learners, through video games embedded inside a well-organized curriculum, can be inducted into professional practices as a form of value-laden deep learning that transfers to school-based skills and conceptual understandings" (p. 38). Although Gee is referring here to game playing, his sentiment is certainly appropriate for game making as well. As evident with the studies previously cited, whether game making is integrated into classroom curricula or occurs "in the wild" at youth-oriented media sites such as Scratch, when children make games, they are making first and foremost for the sake of playability. Conceptual understanding of subjects such as mathematics and science, as well as the dynamics of teamwork and task prioritization, are not learned as ends in and of themselves but put expressly toward the purpose of creating genuinely playable games, resulting in more genuine—and collaborative—learning experiences.

Cultural Dimensions

Although children developing communities around game making and game playing certainly have the potential to mimic professional practice, there is also the wider question as to who actually participates and who can participate in these aforementioned communities. Gaming (Jenkins & Castell, 1998), but also coding communities at large (Margolis et al., 2008; Margolis & Fisher, 2002), have a long history of not engaging girls and minorities, and the reasons are multiple: On one hand, there is the lack of interest, lack of experience, and lack of skill from females, and on the other hand there is the persistent stereotyping of women in these same three areas, which is then compounded by a lack of female player and minority roles and the prevalence of violence in games. This larger issue of gender differences is not germane to gaming alone: It is one that has plagued programming and STEM in the learning sciences at large. Yet despite these persistent issues, constructionist gaming approaches have been seen as a possible remedy for addressing the gender divide so present in the technology culture at large. Although some of the studies presented in this section have examined the influence on computational perspectives such as STEM career interests, low and behold controlled comparisons between different groups in terms of other learning outcomes have been rare.

An early study of game making revealed no significant gender differences in learning programming and disbanded with conventional wisdom at the time believed to be true: Girls could be interested in programming and be interested in gaming if they were just given the opportunity to make their own (Kafai, 1995). The success of girls in constructionist gaming became the launch pad for a whole series of tool developments (such as Storytelling Alice) and research initiatives to use game design to broaden girls' participation in computing (Hayes & Games, 2008; Kafai, 1998; Robertson, 2012). Werner and Denner (2009) research with Storytelling Alice precisely focused on engaging a wider range of children in game making by viewing the process of game design in terms of "variable stories." Working with 22 middle school children (12 boys, 10 girls) over the course of a 2-week camp, Werner and Denner found that 100% of participants were able to successfully make a interactive project through this lens of storytelling. Participants reported feeling empowered by the process of making a project that was interactive, and 61% of participants were able to transition their initial story-based projects into games that, unlike stories, had no fixed outcome.

Yet this push for students—and particularly girls—to make video games over digital stories or interactive art projects also stirs some levels of resentment among certain circles. Although there has been more recent success with game making to bring girls into the so-called clubhouses of computing and gaming, the push for girls making games also revealed a problematic aspect: Why did girls have to design games to prove they were, in fact, tech savvy? This issue received little attention, even from the feminist side, which mightily and justifiably lamented about the reification of stereotypes in girls making games (Jenson & deCastell, 2007). Furthermore, findings from a recent study have began to question to what extent engagement with game making also leads girls to more positive engagement with computing careers (Robertson, 2012). Although girls learned as much about coding as the boys participating in the same research project, they did not express any further career interests. Clearly this is an issue that needs further investigation. However, there may very well be a reason that games are given such a priority over other digital projects such as interactive stories and art. Adams and Webster's (2012) analysis of more than 300 middle school and high school projects in Scratch and Alice suggests that games, more than other project type, best capture certain programming features such as conditional statements and variables. This of course prioritizes coding and computational design over other decidedly "softer" skills such as narrative sequencing. But Adams and Webster's analysis suggests that games by their variable nature necessitate a wider range of technical skills and, perhaps even more important, the capacity to think systematically in terms of inputs and outputs as a recurring relationship.

Moving beyond gender concerns, a small number of studies dealt specifically with the race and ethnicity of their participants and how this relates to their prior experiences. For instance, working with African American high school boys testing video games for design errors through the Glitch Game Testers program speaks well to the challenges of consistently engaging youth from struggling schools (DiSalvo et al., 2014). Working with approximately 10 to 12 students every summer and then part-time throughout the academic year, DiSalvo and colleagues pointed out that the boys' interest in video games usually proves to be an excellent "hook" to get them interested in the program. It quickly grew apparent every summer, however, that they also needed to take into account the financial needs of those youth in perpetuating these children's interest in gaming as beyond a recreational activity (DiSalvo & Bruckman, 2014). Lameman and Lewis's (2011) video-game-making efforts connected with First Nations (Mohawk) youth through the "Skins" pilot workshop. Using the Unity 3D game engine, Lameman and Lewis had the 10 teenage participants choose tribal stories from their own childhood to serve as the starting point for their game narrative. Because none of the participants had prior game-making experience, introducing the "new" in terms of the "old" became a key component of the Skins model. Repenning and colleagues (2015) reported that Agentsheets was also used in tribal communities, engaging American Indian students with success in making games. Likewise, Denner et al. (2012), with Latina middle school girls through the "Girl Game Company" in rural central California, found in postworkshop surveys that it was the opportunity to design their own avatars that got the girls to sign up for the club—even more than the prospect of making a video game.

What is troubling overall from the reviewed body of research on children making games is how little participant background is actually reported and reflected in study designs. The fact that only 66% of studies reported participants' gender and less than half (49%) reported racial or ethnic demographics speaks to a wider body of research (Moreno & Mayer, 2005; Sun & Rueda, 2012) that collectively points to the commonsense (but nonetheless routinely ignored) precept that students' sense of confidence with digital technology is inextricably tied to their own personal identities. Whether an activity taps into these identities and allows for personal expression plays no small role in whether a child will be attracted and persist with any such activity. This focus on other factors when recruiting underrepresented students and youth does not negate the tremendous pull that video games have on children. Indeed, video games are one of the great equalizers in modern society—played by virtually everyone regardless of race, gender, and now age. It is an open question what impact this widespread use has on designing and researching constructionist gaming activities that address various cultural aspects of the students' identities. Research is beginning to map out the intersectionality of race, gender, and ethnicity in gaming (Kafai, Richard, & Tynes, in press).

The critical reality is that children at lower income and predominantly minority schools (Margolis et al., 2008; Warschauer & Matuchniak, 2010) receive considerably less access to and engagement with computing activities that focus on actual creation of digital content. Computers at these lower income schools are less tools for making content than tutors, skilling-and-drilling students on particular academic content. Such a fact needs to be taken into consideration when implementing a game design course or workshop. The wide gap in how computers are implemented into school-day curricula may be also one of the reasons why many researchers have examined motivation beyond classroom learning, turning instead to examine career aspirations as a result of making games for learning. Repenning's (2013) and Ryoo and colleagues' (2013) research on game design specifically focuses on how such active, productive engagement with digital media actually shifts children's attitudes toward computing and opens up the possibility of computing as a career. Making games for learning not only increases motivation for learning but, equally important, also changes students' attitudes toward the goals of learning, allowing children to better grasp the long-term benefits of computing and digital design in terms of a potential career pathway.

DISCUSSION

What we can learn from this review? We pointed out findings across a diverse set of studies with students of different ages, inside and outside of school, and of game-making activities with different tools conducted over the last 20 years. No matter which programming tool was used for game making—whether AdventureAuthor, Agentsheets, Alice 3D, Flash, Greenfoot, Kodu, Logo, Scratch, or Storytelling Alice, to name but a few—no matter which context (school, after-school club, or online community) and no matter which age group (from elementary to high school and college students), making games proved to be a compelling context for learning computational concepts and practices and broadening participants' perspectives on computing and STEM overall. Some studies framed their outcomes within the constructionist framework, whereas others used making games for learning as a context to study problem solving, academic subject matters, or other skills. Even within assessing students' learning of programming, we could see a wide variety of approaches focusing on either particular computational concepts or practices such as debugging and remixing, whereas others examined the nature of problem solving or planning involved in making games. In the following sections, we first discuss the challenges and then articulate directions for serious gaming that connect instructionist and constructionist approaches.

Challenges for Constructionist Gaming

The results from our review on making games for learning, although overwhelmingly positive, also raise several concerns: (a) the involvement in game design, (b) the framing of learning with and about computation, (c) the collection of data, (d) the lack of negative findings, (e) the few studies focused on collaborative learning, and (f) the absence of online opportunities.

One of the obvious challenges in pulling together findings from such a diverse set of studies is the wide variety of contexts, tools, age of students, and time periods used to make games. An unsurprising consequence of such diversity is that the games generated and the learning benefits conferred vary significantly in form and substance. Obviously a game designed in 2 hr compared to one that took weeks of development results in different learning outcomes and relies on different research designs measuring what qualifies as learning. Although the shorter time frames offer opportunities for experimental designs that assess particular outcomes (e.g., Vos et al., 2011), the longer periods offer opportunities for understanding the development of individual children's interests, knowledge, and skills (e.g., Peppler & Kafai, 2007). We hope that both formats continue to be employed in future research to further our understanding of learning in game making.

Second, what researchers conceptualized as learning differed as well. Although we employed a meta-framework developed by Brennan and Resnick (2012) to capture the learning of computational concepts, practices, and perspectives in game making, there is still considerable work needed on how we want to conceptualize and assess computational thinking (see also Werner, Denner, Campe, & Kawamato, 2012) or whether we want to adopt a more expansive view of computational participation (Kafai & Burke, 2014) that focuses not just on personal but also includes social and cultural dimensions of learning—two aspects that proved to be undervalued in the reviewed research but are critical to understanding and designing productive and supportive learning opportunities in game making.

Third, we also note that the standards of reporting varied; in many instances basic demographic data on subjects simply were not provided, details on the implementation of game-making activities were absent, and the nature of the data analysis was not included or fully articulated. We suspect that omission in many instances is not a matter of neglect but rather reflective of how reporting requirements differ between computer science and education. As the new field of computer science education is being constituted and professional organizations such as the American Education Research Association and the International Society for Technology in Education increasingly incorporate CS education, we expect to see improvements here.

Fourth, few negative findings were observed in our review. The most prominent negative outcome was the lack of success in constructionist gaming to raise girls' interest in STEM careers identified in Robertson's (2012) large-scale study, which is contrary to the many other studies that used game making to broaden participants' perspectives on and interest in computing and STEM. This finding is perhaps less surprising considering that gaming and technology cultures have traditionally been unaccepting and even hostile to female individuals. Thus, why would we expect that girls are induced by such activities into joining the club (Margolis & Fisher, 2008)? Other researchers (e.g., Fadjo et al., 2009; Lameman & Lewis, 2011) noted technical difficulties, a finding that we would expect to see reported more often given that many students, boys as well as girls, had little to no background in programming before starting the game-making activities. We are obviously not looking for discounting the value of the constructionist gaming approach. Rather, we are interested in having a more fine-grained understanding of what works for whom and in which context so to better design effective and supportive learning opportunities.

Following up on this point, another area ripe for more documentation and investigation is how collaborative arrangements can enhance and further learning opportunities in constructionist gaming. We know from research on game play how important collaborations are to motivate and sustain player's efforts to move ahead in the game. With the exception of the research on pair programming in making games (Denner & Werner, 2007; Werner, Denner, Bliesner, & Rex, 2009), we saw very few other studies include collaborative learning in their designs. Many studies obviously took informal advantage of collaboration between game designers by having them play one another's games and provide feedback, but they did not explicitly investigate outcomes as Huang, Hong, and Chen (2014) did in their study examining the impact of peer feedback. We also know from related studies on children's collaborative programming of simulations that teams can be productive learning groups but team members' learning is impacted by prior programming and experience (Ching & Kafai, 2008). We have only recently begun to understand how such collaborative programming tasks can be designed for mutual interdependence and integration of challenging computational concepts (Fields, Vasudevan & Kafai, 2015). How such collaborations in game making can be constructed and supported is a wide territory for further research.

Finally, related to this absence on collaboration in game making, we also noted an absence in research studies specifically identifying online opportunities to make and share games. Programs like Scratch, Alice, and Kodu—which were once considered "tools"—have redefined themselves as "communities." They have migrated from stand-alone software packages to real-time online applications and, in the process, have reshaped contemporary literacy practices in DIY communities, extending computational thinking into computational participation (Kafai & Burke, 2014). Collaborations such as the Gray Bear collaborative (Aragon, Poon, Monroy-Hernandez, & Aragon, 2009) are not the norm at websites, as research by Luther and Bruckman (2008) documented on the Newgrounds site. Of interest, those that do rely on effective social skills as much as technical prowess. Fostering collaborations and communities of game designers, online and offline from the early classrooms where kids first designed their individual games to the massive online communities where games are some of the most popular designs shared is another key challenge that moves children from making into sharing—a far too rare occurrence in today's kids DIY sites (Grimes & Fields, 2015). Gaining access to a wide and appreciative community means that players have the opportunity to leverage that community as an extension of the tool itself, with meaningful feedback serving to help fledgling designers gain a foothold into what works in game design, whereas more experienced designers can grow in proficiency and create increasingly intricate games. But before this community can be leveraged, young makers must first have the confidence to share their own work, as well as learn the nature and respect the role of constructive feedback. This is no small process and one that future research is responsible for documenting.

Opportunities for Connected Gaming

Many of the benefits that we observed for constructionist gaming can also be found in instructionist gaming, where students play educational games for learning. But what distinguishes constructionist gaming most from instructionist gaming is its focus on engaging students in design. How can the particular software be programmed to optimize game play and how does one map the wider gaming narrative to be consistently dynamic and also fair? These are the difficult questions game makers must regularly ask, and the learning benefits from asking such questions cannot be ignored. These outcomes are perhaps what most distinguish making games from playing games for learning, but we also see here the greatest potential for integration. We already know that metagaming (Gee, 2003) is an important part of gaming culture: There are game players who venture into scripting their own fan sites, building complex spreadsheets to understand system designs, and some even venturing into making their own level extensions and games. We see here connection points between instructionist and constructionist gaming that can help move serious gaming into new and productive directions, a direction that we call *connected gaming* (Kafai & Burke, in press).

Connected gaming purports that learning to play and make games is ultimately part of a larger gaming ecology in which the traditional roles of the "player" and the "maker" are no longer treated as distinct entities. We can already see instances where this is happening, not the least in commercial games. Some of the most popular games on the market today include level and character modding as a central feature (El Nasr & Smith, 2006; Hayes-Gee & Tran, 2015). Such games encourage modding precisely because it brings the end user closer to the game. Connected gaming opportunities need to be designed for serious gaming. For instance, the well-known game SimCity and the newly released Scratch 2.0 program each offer an apt example of instructionist and constructionist approaches merging together into this notion of connected gaming. From the instructionist gaming side, SimCity illustrates how playing a game can contribute to a better understanding of the constantly shifting dynamics of a simulated world (Salen, 2013). From the constructionist gaming side, new features in Scratch 2.0 environment allow for writing programs that survey information from participants at the site to better understand who is sharing online and what they are sharing (Dasgupta, 2013). These are two different approaches, but both have the same goal of "looking under the hood" for understanding what happens in the massive and interconnected community. Although the tools in SimCity are programmed by experts, the tools in Scratch are programmed by players themselves. Going forward, there is no reason that SimCity couldn't offer programmable tools that would allow end-users to customize their investigations, whereas preprogrammed tools in Scratch can be incorporated for those wanting to experience an actual simulation before designing their own. In fact, the latter approach already exists.

In bringing together instructionist and constructionist approaches in connected gaming we open up new perspectives for participation in serious gaming. We are blurring the lines between what it is "to play" and "to make" digitally, which, we have argued, is an inherently welcome merger, because in commercial gaming these boundaries have been crossed a long time ago. In fact, Gee (2003) himself made this connection between playing and making when he reflected toward the end of *What Video Games Have to Teach Us About Learning and Literacy* that "[i]n fact, it is a crucial learning principle that people learn best when they have an opportunity to talk (and write) about what they are learning. . . . I may well have learned more by writing this book than anyone has by reading it" (pp. 215–219).

Of course, he was commenting on writing his book here. But writing, like programming, is a maker activity where people construct an artifact. It does not matter whether the artifact is digital, material, or a hybrid of the two. Most of the current discussions on serious gaming have focused on playing games for learning, and most return to Gee's learning principles as the touchstone of what can be learned from the process. We have argued here that those personal, social, and cultural principles that are rooted in playing games for learning likewise exist when making games for learning. And why not? Gee's main point was that games were great examples of learning environments. But of course not everything has to be a game played; it can also be a game made.

CONCLUSION

When the field of serious gaming started, attention focused on proving the effectiveness of instructionist gaming (R. E. Clark, 2007) and "researching learning in popular gaming cultures, designing learning environments based on those principles, and reconceptualizing educational practice for an interactive age" (Squire, 2007, p. 51). Constructionist gaming was not part of either discussion in building the field of serious gaming. But if we want to realize the potential of serious gaming, we need embrace a broader agenda that recognizes that opening access and participation in serious games is not solely a matter of making better games for learning but allowing students themselves to make the games they would like to see and play. Ultimately, our goal is to promote environments that are good for learning, and it is here where constructionist approaches join instructionist efforts and where we can make a case for "connected gaming," an approach that doesn't draw boundaries between players and designers as participants of digital media culture but rather sees them as complementary to each other, as Papert (1995) already envisioned:

> If one does belong to a culture in which video games are important, transforming oneself from a consumer to a producer of games may well be an even more powerful way for some children to find importance in what they are doing. (p. iii)

FUNDING

This analysis and writing of this paper were supported by a collaborative grant from the National Science Foundation (NSF#1450877) to the first author. The views expressed are those of the authors and do not necessarily represent the views of the National Science Foundation, University of Pennsylvania, or College of Charleston.

REFERENCES

Abelson, H. & DiSessa, A. (1980). *Turtle geometry: The computer as a medium for exploring mathematics*. Cambridge, MA: MIT Press.

Adams, J. C. & Webster, A. R. (2012). What do students learn about programming from game, music video, and storytelling projects? In *Proceedings of the 43th ACM Technical symposium on computer science education* (pp. 643–648). New York, NY: ACM.

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. *Educational Technology Research and Development*, *62*, 583–600.

Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the game-design and learning (GDL) after-school program. *Computers & Education*, *75*, 72–81.

Al-Bow, M., Austin, D., Edgington, J., Fajardo, R., Fishburn, J., Lara, C., & Meyer, S. (2009). Using game creation for teaching computer programming to high school students and teachers. *ACM SIGCSE Bulletin*, *41*, 104–108.

Allsop, Y. (2015). A reflective study into children's cognition when making computer games. *British Journal of Educational Technology*. Advance online publication. http://dx.doi.org/10.1111/bjet.12251

Aragon, C., Poon, S., Monroy-Hernandez, A., & Aragon, D. (2009). A tale of two online communities: Fostering Collaboration and Creativity in Scientists and Children. In *Proceedings of the Creativity and Cognition Conference* (pp. 9–18). New York, NY: ACM Press.

Atwood, J. (2012, May 15). Please don't learn to code. *Coding Horror*. Retrieved from http://blog.codinghorror.com/please-dont-learn-to-code/.

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM Technical Symposium on computer science education* (pp. 245–250). New York, NY: ACM.

Baytak, A., & Land, S. M. (2010). A case study of educational game design *by* kids and *for* kids. *Procedia—Social and Behavioral Sciences*, *2*, 5242–5246.

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the annual meeting of the American Educational Research Association, Vancouver, British Columbia, Canada. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Buckingham, D., & Burn, A. (2007). Game literacy in theory and practice. *Journal of Educational Multimedia and Hypermedia*, *16*, 323–349.

Burke, Q., & Kafai, Y.B. (2014). A decade of programming games for learning: From tools to communities. In H. Agius & M. C. Angelides (Eds.), *The handbook of digital games* (pp. 689–7009). New York, NY: Wiley & Sons.

Burke, Q. & Kafai, Y. B. (in press). DIY zones for Scratch designs in class and club. *The International Journal of Learning and Media*.

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, *55*, 1098–1111.

Ching, C. C. & Kafai, Y. B. (2008). Peer pedagogy: Student collaboration and reflection in learning through design. *Teachers College Record*, *110*, 2601–2632.

Clark, K. & Sheridan, K. (2010). Game design through mentoring and collaboration. *Journal of Educational Multimedia and Hypermedia*, *19*, 125–145.

Clark, R. E. (2007). Learning from serious games? Arguments, evidence, and research suggestions. *Educational Technology*, *47*(3), 56–59.

Cassell, J. & Jenkins, H. (1998). (Eds.). *From Barbie to Mortal Kombat*. Cambridge, MA: MIT Press.

Cuban, L. (2014, August 16). Tissue paper reforms: Coding for kindergartners. *On School Reform and Classroom Practice*. Retrieved from https://larrycuban.wordpress.com/2014/08/16/tissue-paper-reforms-coding-for-kindergartners/.

Dasgupta, S. (2013). From surveys to collaborative art. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 28–35). New York, NY: ACM

DeLay, D., Hartl, A. C., Laursen, B., Denner, J., Werner, L., Campe, S., & Ortiz, E. (2013). Learning from friends: Measuring influence in a dyadic computer instructional setting. *International Journal of Research and Method in Education*, *37*, 190–205.

Denner, J. & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research*, *37*, 131–150.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, *46*, 277–296.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, *58*, 240–249.

DiSalvo, B. & Bruckman, A. (2014). From interest to values. *Communications of the ACM*, *54*(8), 27–29.

DiSalvo, B., Guzdial, M., Bruckman, A., & McKlin, T. (2014). Saving face while geeking out: Video game testing as a justification for learning Computer Science. *Journal of the Learning Sciences*, *23*, 272–315.

Earp, J. (2015). Game making for learning: A systematic review of the research literature. In *Proceedings of 8th International Conference of Education, Research and Innovation (ICERI2015)* (pp. 6426–6435). Seville, Spain: International Academy of Technology, Education and Development.

El Nasr, M. & Smith, S. (2006). Learning through game modding. *Computers in Entertainment*, *4*, 1–20.

Esper, S., Foster, S. & Griswold, W. (2013). CodeSpells: Embodying the metaphor of wizardry for programming. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 249–254). New York, NY: ACM.

Fadjo, C. L., Hallman, G., Jr., Harris, R., & Black, J. B. (2009, June). Surrogate embodiment, mathematics instruction and video game programming. In G. Siemens & C. Fulford (Eds.), *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 2787–2792). Waynesville, NC: AACE.

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: Fostering participatory culture by making music videos in a Scratch high school coding workshop. *Interactive Learning Environments*, *23*(5), 1–21. http://dx.doi.org/10.1080/10494820.2015.1065892.

Fristoe, T., Denner, J., Mateas, M., MacLaurin, M., & Wardrip-Fruin, N. (2011, June). Say it with systems: Expanding Kodu's expressive power through gender-inclusive mechanics. In *Proceedings of the 6th International Conference on the foundations of digital games* (pp. 227–234). New York, NY: ACM.

Games, I. A. (2010). Gamestar Mechanic: Learning a designer mindset through communicational competence with the language of games. *Learning, Media, and Technology*, *35*, 31–52.

Games, I. A. & L. Kane. (2011). Exploring adolescent's STEM learning through scaffolded game design. In *Proceedings of the 6th International*

Conference on Foundations of Digital Games (pp. 1–8). New York, NY: ACM.

Garrelts, N. (2014). (Ed.). *Understanding Minecraft: Essays on play, community, and possibilities*. Jefferson, NC: McFarland.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York, NY: Palgrave Macmillan.

Gee, J. P. (2008). "Learning and games." In K. Salen (Ed.), *The ecology of games: Connecting youth, games, and learning* (pp. 21–40). Cambridge, MA: MIT Press.

Gee, J. P. (2010). *New digital media and learning as an emerging area and 'worked examples' as one way forward*. Cambridge, MA: MIT Press.

Girard, C., Ecalle, J., & Magant, A. (2012). Serious games as new educational tools: how effective are they? A meta-analysis of recent studies. *Journal of Computer Assisted Learning*, *29*(3), 1–13.

Grimes, S., & Fields, D. (2015). Children's media making but not sharing. The potential and limitations of child-specific DIY media websites for a more inclusive media landscape. *Media International Australia*, *154*, 112–122.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*, 38–43.

Hayes, E. R., & Games, I. A. (2008). Making computer games and design thinking: A review of current software and strategies. *Games and Culture*, *3*, 309–322.

Hayes-Gee, E. R., & Tran, K. (2015). Video game making and modding. In B. Guzzetti & M. Lesley (Eds.), *Handbook of research on the societal impact of digital media* (pp. 238–267). Hershey, PA: IGI Global.

Holbert, N. R., & Wilensky, U. (2014). Constructible authentic representations: Designing video games that enable players to utilize knowledge developed in-game to reason about science. *Technology, Knowledge and Learning*, *19*, 53–79.

Honey, M. & Kanter, D. (2013). (Eds.). *Design, make, play: Growing the next generation of STEM innovators*. New York, NY: Routledge.

Howland, K. & Good, J. (2015). Learning to communicate computationally with a Flip: A bi-modal programming language for game creation. *Computers & Education*, *80*, 224–240.

Hwang, G. J., Hung, C. M., & Chen, N. S. (2014). Improving learning achievements, motivations and problem-solving skills through a peer assessment-based game development approach. *Educational Technology Research and Development*, *62*, 129–145.

Ito, M., Horst, H., Bittanti, M., Boyd, D., Herr-Stephenson, B., Lange, P., . . . Robinson, R. (2009). *Living and learning with new media: Summary of findings from the digital youth project*. Chicago, IL: John D. & Catherine T. MacArthur Foundation.

Javidi, G., & Sheybani, W. (2010). Making youth excited about STEM education. *Journal of Computing Sciences in Colleges*, *26*, 140–147.

Jenkins, H., Clinton, K., Purushotma, R., Robison, A., & Weigel, M. (2006). *Confronting the challenges of participatory culture: media education for the 21st century* (White Paper). Chicago, IL: John D. & Catherine T. MacArthur Foundation.

Jenson, J., & deCastell, S. (2007). Girls playing games: Rethinking stereotypes. In *Proceedings of future play* (pp. 9–16). New York, NY: ACM.

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Mahwah, NJ: Erlbaum.

Kafai, Y. B. (1998). Video game design by girls and boys: Variability and consistency of gender differences. In J. Cassell & H. Jenkins (Eds.), *From Barbie to Mortal Kombat* (pp. 90–114). Cambridge, MA: MIT Press.

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games & Culture*, *1*, 36–40.

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: MIT Press.

Kafai, Y. B., & Burke, Q. (in press). *Connected gaming: What making video games can tell us about learning and literacy*. Cambridge, MA: MIT Press.

Kafai, Y. B., Burke, W. Q., & Mote, C. (2012). What makes competitions fun to participate? Developing a middle school classroom workshop for video game design. In *Proceedings of the 2012 Conference on interaction design and children* (pp. 284–87). New York, NY: ACM.

Kafai, Y. B., & Fields, D. A. (2013). *Connected play: Tweens in a virtual world*. Cambridge, MA: MIT Press.

Kafai, Y. B., Fields, D. A., Roque, R., Burke, W. Q., & Monroy-Hernandez, A. (2012). Collaborative agency in youth online and offline creative production in Scratch. *Research and Practice in Technology Enhanced Learning*, *7*, 63–87.

Kafai, Y. B., Franke, M., Ching, C., & Shih, J. (1998). Game design as an interactive learning environment fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematical Learning*, *3*, 149–184.

Kafai, Y. B., & Peppler, K. A. (2011). Youth, technology, and DIY: Developing participatory competencies in creative media production. *Review of Research in Education*, *35*(1), 89–119.

Kafai, Y. B., & Peppler, K. A. (2012). Developing gaming fluencies with Scratch: Realizing game design as an artistic process. In C. Steinkuehler, K. Squire, & S. Barab (Eds.), *Games, learning, and society: Learning and meaning in the digital age* (pp. 355–380). New York, NY: Cambridge University Press.

Kafai, Y. B., Peppler, K. A., & Chapman, R. (Eds.). (2009). *The Computer Clubhouse: Constructionism and creativity in youth communities*. New York, NY: Teachers College Press.

Kafai, Y. B., Richard, G., & Tynes, B. (in press). (Eds.). *Diversifying Barbie and Mortal Kombat: New perspectives on gender, race and sexuality in gaming*. Pittsburgh, PA: ETC Press.

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, *73*, 26–39.

Khalili, N., Sheridan, K., Williams, A., Clark, K., & Stegman, M. (2011). Students designing video games about immunology: Insights for science learning. *Computers in the School*, *28*, 228–240.

Klopfer, E. & Haas, J. (2012). *The more we know: NBC News, educational innovation, and learning from failure*. Cambridge, MA: MIT Press.

Klopfer, E., Osterweil, S., & Salen, K. (2009). *Education arcade*. Cambridge, MA: MIT

Knobel, M. & Lankshear, C. (Eds.). (2010). *DIY media: Creating, sharing, and learning with new technologies*. New York, NY: Peter Lang.

Koh, A., Repenning, A., Nickerson, H., Endo, Y., & Motter, P. (2013). Will it stick?: Exploring the sustainability of computational thinking education through game design. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE)* (pp. 597–602). New York, NY: ACM.

Lakanen, A. J., Isomöttönen, V., & Lappalainen, L. (2014). 5 years of game programming outreach: Understanding student differences. In *Proceedings of the 45th ACM technical symposium on Computer Science Education (SIGCSE)* (pp. 647–652). New York, NY: ACM.

Lameman, B. A., & Lewis, J. E. (2011). Skins: Designing games with First Nations youth. *Journal of Game Design and Development Education*, *1*, 54–63.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, UK: Cambridge University Press.

Luther, K., & Bruckman, A. (2008). Leadership and success factors in online creative collaboration. In *Proceedings of Computer-Supported Cooperative Work* (pp. 343–352). New York, NY: ACM.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the shallow end. Education, race, and computing*. Cambridge, MA: MIT Press.

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT Press.

Mayer, R. (2014). *Computer games for learning*. Cambridge, MA: MIT Press.

Moreno, R. & Mayer, R. E. (2005). Role of guidance, reflection and interactivity in an agent-based multimedia game. *Journal of Educational Psychology*, *97*, 177–128.

Mouza, C., Pan, Y., Pollock, L., Atlas, J., & Harvey, T. (2014, October). *Partner4CS: Bringing computational thinking to middle school through game design.* Paper presented at Stanford University FabLearn Conference on Creativity and Fabrication, Palo Alto, CA. Retrieved from http://fablearn.stanford.edu/2014/wp-content/uploads/fl2014_submission_21.pdf

National Research Council. (2011). *Learning science through simulations and games*. Washington, DC: The National Academies Press.

Navarrete, C. (2013). Creative thinking in digital game design and development: A case study. *Computers & Education*, *69*, 320–331.

Owston, R., Wideman, H., Ronda, N. S., & Brown, C. (2009). Computer game development as a literacy activity. *Computers & Education*, *53*, 977–989.

Palumbo, D. B. (1990). Programming language/problem-solving research: A review of relevant issues. *Review of Educational Research*, *60*, 65–89.

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

Papert, S. (1991). Situating constructionsim. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–12). Norwood, NJ: Ablex.

Papert, S. (1995). Does easy do it? Children, games and learning. *Game Developer* (June 1998). Retrieved June 12, 2015, from http://www.papert.org/articles/Doeseasydoit.html

Pelletier, C. (2008). Producing difference in studying and making computer games: How students construct games as gendered in order to construct themselves as gendered. In Y. B. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat* (pp. 145–161). Cambridge MA: MIT Press.

Pelletier, C., Burns, A., & Buckingham, D. (2010). Game design as textual poaching: Media literacy, creativity, and game making. E-*Learning and Digital Media*, *7*, 90–107.

Peppler, K. (2013). Social media and creativity. In D. Lemish (Ed.), *International handbook of children, adolescents, and media* (pp. 193–200). New York, NY: Routledge.

Peppler, K., & Kafai, Y. B. (2007). What video game-making can teach us about learning and literacy: Alternative pathways into participatory culture. In A. Baba (Ed.), *Situated play: Proceedings of the Third International Conference of the Digital Games Research Association (DiGRA)* (pp. 369–376). Tokyo, Japan: The University of Tokyo.

Peppler, K. & Kafai, Y. B. (2010). Gaming fluencies: Pathways into a participatory culture in a community design studio. *International Journal of Learning and Media*, *1*(4), 1–14.

Piaget, J. (1951). *Play, dreams, and imitation in childhood*. New York, NY: Norton.

Provenzo, E. J. (1991). *Video kids: Making sense of Nintendo*. Cambridge, MA: Harvard University Press.

Repenning, A. (2013). Making programming accessible and exciting. *IEEE Computer*, *46*(6), 78–81.

Repenning, R., Webb, D. C., Koh, K. H., Nickerson, N., Miller, B., Brand, S. B., . . . Repenning, N. (2015). Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. *Transactions on Computing Education*, *15*(2). doi:10.1145/2700517

Reynolds, R., & Chiu, M. (2015). Reducing digital divide effects through student engagement in coordinated game design, online resource uses, and social computing activities in school. *Journal of the Association for Information Science and Technology*. http://dx.doi.org/10.1002/asi.23504

Reynolds, R., & Harel Caperton, I. (2011). Contrasts in student engagement, meaning-making, dislikes, and challenges in a discovery-based program of game design learning. *Educational Technology Research and Development*, *59*, 267–289.

Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education*, *59*, 385–398.

Robertson, J. (2013). The influence of a game making project on male and female learners' attitudes to computing. *Computer Science Education*, *23*, 58–83.

Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education*, *50*, 559–578.

Robertson, J., & Nicholson, K. (2007). Adventure Author: A learning environment to support creative design. In *Proceedings of the 6th International Conference on Interaction Design and Children* (pp. 37–44). New York, NY: ACM.

Ryoo, J., Margolis, J., Lee, C. H., Sandoval, C. D. M., & Goode, J. (2013). Democratizing computer science knowledge: Transforming the face of computer science through public school education. *Learning, Media, and Technology*, *38*, 161–181.

Salen, K. (2007). Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia*, *16*, 301–322.

Salen, K. (2013, March). *Agile development meets evidence-centered design: Glasslab and the design of game-based assessments*. Paper presented at the Digital Media & Learning Conference, Chicago, IL.

Schanzer, E., Fisler, K., & Krishnamurthi, S. (2013, October). *Bootstrap: Going beyond programing in after-school computer science*. Presented at SPLASH-E, Education track of the OOPSLA/SPLASH conference, Indianapolis, IN.

Schanzer, E., Fisler, K., Krishnamurthi, S., & Felleisen, M. (2015). Transferring skills at solving word problems from computing to algebra through Bootstrap. In *Proceedings of the 46th ACM Technical symposium on computer science education* (pp. 616–621). New York, NY: ACM.

Seaborn, K., El-Nasr, M. S., Milam, D., & Young, D. (2012). Programming, PWNed: Using digital game development to enhance learners' competency and self-efficacy in a high school computing course. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 93–98). New York, NY: ACM.

Shaffer, D. (2007). *How computer games help children learn*. New York, NY: Palgrave MacMillan.

Sheridan, K., Clark, K., & Williams, A. (2013). Designing games, designing roles: A study of youth agency in an urban informal education program. *Urban Education*, *48*, 734–758.

Sprung, G., Zimmermann, L., Nischelschwitzer, A., Strohmaier, R., & Schadenbauer, S. (2011, July). *Creating games approach: Models and implications for the usage of game design as a teaching method*. Paper presented at the International Conference on Information Communication Technologies in Education, Rhodos, Greece.

Squire, K. (2007). Games, learning, and society: Building a field. *Educational Technology*, *47*(5), 51–55.

Squire, K. (2010). From information to experience: Place-based augmented reality games as a model for learning in a globally networked society. *Teachers College Record*, *112*, 2565–2602.

Squire, K. (2011). *Video games and learning: Teaching and participatory culture in the digital age*. New York, NY: Teachers College Press.

Steinkuehler, C., & Duncan, S (2008). Scientific habits of mind in virtual worlds. *Journal of Science Education Technology*, *17*, 530–543.

Sun, J. C., & Rueda, R. (2012). Situational interest, computer self-efficacy and self-regulation: Their impact on student engagement in distance education. *British Journal of Educational Technology*, *43*, 191–204.

Taylor, R. P. (1980). *The computer in schools: Tutors, tools, tutees*. New York, NY: Teachers College Press.

Turkle, S., & Papert, S. (1991). Epistemological pluralism: Styles and voices within the computer culture. *Signs*, *16*, 128–157.

Vattel, L., & Risconscente, M. (2012). MathMaker: Teaching math through game design and development. In C. Martin, A. Ochsner, & K. Squire (Eds.), *Proceedings of GLS 8.0* (pp. 313–321). Pittsburgh, PA: ETC Press.

Vogel, J. J., Vogel, D. S., Cannon-Bowers, J., Bowers, C. A., Muse, K., & Wright, M. (2013). Computer gaming and interactive simulations for learning: A meta-analysis. *Journal of Educational Computing Research*, *34*, 229–243.

Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, *56*, 127–137.

Warschauer, M., & Matuchniak, T. (2010). New technology and digital worlds: Analyzing evidence of the equity in access, use and outcomes. *Review of Research in Education*, *34*, 179–225.
Webb, D., Repenning, A., & Koh, A. (2012). Toward an emergent theory of broadening participation in computer science education. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 173–178). New York, NY: ACM.
Werner, L., Campe, S., & Denner, J. (2005). Middle school girls + games programming= information technology fluency. In *Proceedings of the 6th Conference on Information Technology Education* (pp. 301–305). New York, NY: ACM.
Werner, L., Campe, S., & Denner, J. (2012). Children learning computer science concepts via Alice game-programming. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 427–432). New York, NY: ACM.
Werner, L., & Denner, J. (2009). Pair programming in middle school: What does it look like?. *Journal of Research on Technology in Education*, *2*, 29–50.
Werner, L., Denner, J., Bliesner, M., & Rex, P. (2009, April). Can middle-schoolers use Storytelling Alice to make games? Results of a pilot study. In *Proceedings of the 4th International Conference on foundations of digital games* (pp. 207–214). New York, NY: ACM.
Werner, L., Denner, J., & Campe, S. (2014). Using computer game programming to teach computational thinking skills. In K. Schrier (Ed.), *Learning, education and games: Volume 1, curricular and design considerations* (pp. 37–53). Pittsburgh, PA: ETC Press.
Werner, L., Denner, J., Campe, S., & Kawamato, D. C. (2012, February). The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 215–220). New York, NY: ACM.
Wilensky, U. (1995). Paradox, programming, and leading probability: A case study in a connected mathematics framework. *The Journal of Mathematical Behavior*, *14*, 253–280.
Williams, L., & Kessler, R. (2002). *Pair programming illuminated*. Boston, MA: Addison-Wesley.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.
Wouters, P., van Nimwegen, C., von Oostendorp, H., & van der Spek, E. D. (2013). A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, *105*, 249–265.
Yang, Y. C., & Chang, C. L. (2013). Empowering students through digital game authorship: Enhancing concentration, critical thinking, and academic achievement. *Computers & Education*, *68*, 334–344.
Young, M. F., Slota, S., Cutter, A. R., Jalette, G., Mullin, G., Lai, B., . . . Yukhymenko, M. (2012). Our princess is another castle: A review of trends in serious gaming for education. *Review of Educational Research*, *82*, 61–89.

APPENDIX
Summary of Key Features of Reviewed Studies

Study	*N*	*Learner*	*Time Frame*	*Context*	*Tools*	*Findings*
Adams & Webster (2012)	300	Middle school G = n/a D = n/a	Several hours per week	Summer camp	Alice Scratch	CC
Akcaoglu (2014)	18	Middle school G = 40%F, 60%M D = n/a	10 days: 5 hr each day	Summer camp	Kodu	CL
Akcaoglu & Kohler (2014)	44	Middle school G = 25% F/50% F D = n/a	5 weekends: 3 hr each day	Weekend camp	Kodu	CP
Al-Bow, Austin, Edgington, et al. (2009)	17	High school G = n/a D = n/a	2 weeks: 2.5 hr each day	Summer camp	Greenfoot	CL
Allsop (2015)	30	Elementary school G = n/a D = n/a	6 months	School	Alice 3D	CL
Aragon, Poon, Monroy-Hernandez, & Aragon (2009)	1	Middle + High school G = 35% F, 65% M D = n/a	3 months	Online	Online Scratch	CP
Basawapatna, Koh, Repenning, Webb, & Marshall (2011)	32	Middle school G = n/a D = n/a	8 weeks	School	Agentsheets	CP
Baytak & Land (2010)	10	Middle school G = n/a D = n/a	8 weeks: 2 sessions of 45 min	School	GameMaker	CP; CO
Carbonaro, Szafron, Cutumisu, & Schaeffer (2010)	50	High school G = 48% F, 52% M D = n/a	12 hr	School	ScriptEase	CL
Clark & Sheridan (2010)	139	Middle + high school G = 93%–69% M D = 80% AA, 9% C, 2% M, 9% no report	Semester: 10 weekends sessions	After school	Alice, Maya, GameMaker, Scratch, Flash, 2D 3D	CC; CP; CPe

(Continued on next page)

APPENDIX *(Continued)*

Study	*N*	*Learner*	*Time Frame*	*Context*	*Tools*	*Findings*
DeLay et al. (2013)	160	Middle school G = 36% F, 64% M D = 46% C, 14% L, 28% AA	20 hr	After school	Alice; Storytelling Alice	CC; CL
Denner, Werner, Campe, & Ortiz (2014)	320	Middle school G = 36% F, 64% M D = 46% C; 37% L	20 hr	Elective at school (38%); afterschool (62%)	Alice, Storytelling Alice, StageCast	CC; CP
Denner, Werner, & Ortiz (2012)	59	Middle school G = 100% F D = 72% L, 22% C	14 months	After school + Summer	Stagecast Creator	CC; CP
DiSalvo, Guzdial, Bruckman, & McKlin (2014)	35	High school G = 100% M D = 100% AA	3 years	Summer + Weekends	Game Testing	CP, CPE
Esper, Foster, & Griswold (2013)	17	Elementary, middle, & high school G = 65% F, 35% M D = n/a	1–2 hr	After school	CodeSpells	CP
Fajdo et al. (2009)	n/a	Middle school G = n/a D = n/a	Not specified	After school	Scratch	CC; CP; CO
Fristoe, Denner, Mateas, MacLaurin, & Wardrip-Fun (2011)	n/a	Middle school G = 100%F D = n/a	2 weeks 12 hr	After school	Kodu	CL
Games & Kane (2011)	36/12	High school G = n/a D = n/a% diverse	3 months	School	Kodu Flash	CP
Holbert & Wilensky (2014)	11	Elementary + middle school G = n/a% F, n/a% M D = n/a	2 1-hr play sessions	After school	Sandbox NetLogo	CO; CL
Howland & Good (2015)	55	Middle school G = 53% F/ 47% M D = n/a	8 weeks, 2 sessions each week	School	Flip	CC; CP
Hwang, Hung, & Chen (2014)	167	Middle school G = n/a D = n/a	10 weeks, 50-min session	School		CO; CL
Javidi & Sheybani (2010)	78	Middle school G = 40% F 60% M D = % n/a low income	36 months 2-week summer, 10 weekend	After school/ Summer camp	Kahootz, Scratch, Alice, Flash	CC; CP; CPe
Kafai (1995)	50	Elementary school G = 50% F, 50% M D = 49% L, 20% AA, 11%C, 20% mixed	12 weeks; 1 hr each day	School	Logo	CC; CP; CO
Kafai & Peppler (2012)	5XX	Elementary + middle + high school G = 45%F, 55% M D = 100% L + AA	2 years	Community Center	Scratch	CC
Ke (2014)	64	Middle school G = 43% F, 57% M D = n/a	6 weeks; 2 hr each week	School	Scratch	CPe
Khalili, Sheridan, Williams, Clark, & Stegman (2011)	16	High school G = 38% F, 62% M D = 100%AA	4 weeks, every day	After school	GameMaker	CO; CL
Koh, Repenning, Nickerson, Endo, & Motter (2013)	46 schools	Middle school G = n/a D = 52%C, 43% L, 16% AA, 11% AI	8 weeks, several games	School	Agent-sheets	CC
Lakanen, Isomöttönen, & Lappalainen (2014)	462	Middle + high school G = 7% F 93% M D = n/a	5 years: 1 week with 5 hr each day	Summer	Jypeli C#	CPe

(Continued on next page)

APPENDIX *(Continued)*

Study	*N*	*Learner*	*Time Frame*	*Context*	*Tools*	*Findings*
Lameman & Lewis (2011)	10	High school G = n/a D = 100% AI	2 weeks (80 hr total)	After school	Unity 3D	CPe
Luther & Bruckman (2008)	17	High school G = n/a D = n/a	Not specified	Online	Flash	CPe; CL
Mouza, Pan, Pollock, Atlas, & Harvey (2014)	14	Middle school G = n/a D = 68% C 12% A, 11% AA, 4% L 5% M	8 weeks	After school	Scratch	CC; CPe
Navarrete (2013)	12	Middle school G = n/a D = 85% L	Academic year (daily class)	School	Education Games Central	CPe
Owston, Wideman, Ronda & Brown (2009)	311	Elementary school G = 52% F, 48% M D = n/a	10 weeks daily	School		CP; CO
Pelletier, Burns, & Buckingham (2010)	29	Middle school G = n/a% mixed D = n/a	5 weeks, 9 sessions	School	Mission Maker	CP; CL
Peppler & Kafai (2007)	1	High school G = 45% F 55% M D = 100% AA & L	2 years	Community Center	Scratch	CC; CP
Peppler & Kafai (2010)	643	Elementary + middle + high school G = 45% F, 55% M D = 100% AA & L	2 years	Community Center	Scratch	CC
Repenning et al. (2015); see also Koh et al. (2013) and Webb et al. (2012)	268	Middle school + (college) G = 45% F, 55%M D = 52%C, 43% L, 16% AA, 11% AI	Semester: 2 weeks; School: 8-week unit with multiple games	College/ School	Agent-sheets	CP
Reynolds & Chiu (2015)	242	Middle + high school G = n/a D = n/a Low SES	Year-long 100 hr	School	Flash	CL
Reynolds & Harel Caperton (2011)	93	Middle school G = 41%F, 59%M D = 59% L, 15% C, 13% A, 11% AA	Several weeks	School, Club	Flash	CL
Robertson (2012)	25	Middle school G = 42% F, 58% M D = n/a	6 weeks, 18 days	School	Adventure Author	CC CP
Robertson (2013)	225	Middle + high school G = 47% F, 53% M D = n/a	6 weeks	School	Adventure Author	CPe
Robertson & Howells (2008)	30	Elementary school G = n/a D = n/a	8 weeks, 20 hr total	School	Never-winter	CP; CL
Robertson & Nicholson (2007)	19	High school G = 47% F, 53% M D = n/a	Camp: 5 sessions of 2–3 hr; School: 8 sessions	Summer /School	Adventure Author	CL
Schanzer, Fisler, & Krishnamurthi (2013)	1400 total	Middle school G = n/a D = n/a% Low SES	10 session, 90 min each	After school	Bootstrap	CO

(Continued on next page)

APPENDIX *(Continued)*

Study	*N*	*Learner*	*Time Frame*	*Context*	*Tools*	*Findings*
Seaborn, El-Nasr, Milam, & Young (2012)	n/a	High school G = n/a D = n/a	5 months	School	GameMaker	CC; CPe
Sheridan, Clark, & Williams (2013)	415	High school G = 93%-69% M D = 80-94% AA, 9-6% M, 2-6% other	4 years	After school	Alice, Maya, GameMaker Scratch, Flash	CL
Sprung, Zimmermann, Nischelschwitzer, Strohmaier, & Schadenbauer (2011)	143	Middle + high school G = n/a D = n/a	n/a	School	Scratch	CL
Vattel & Riconscente (2012)	10	Middle school G = n/a D = 64% AA, 36% L	10 weeks 4 sessions	School	Math Maker	CPe; CO; CL
Vos, Van der Meijden, & Denessen (2011)	235	Middle G = n/a D = n/a	2 sessions	School	Memory Spleen	CL
Webb, Repenning, & Koh (2012)	1,420	Middle school G = 45% F, 55%M D = 52%C, 43% L, 16% AA, 11% AI	2 years; 8-week unit with multiple games	School	Agentsheets	CPe
Werner, Campe, & Denner (2005)	33	Middle school G = 100% F; D = n/a	6 weeks: 4 times a week/12 weeks, 2 hr	Summer/ After school	Flash	CC
Werner, Campe, & Denner (2012); see also Werner, Denner, & Campe (2014)	325	Middle school G = 37% F, 63% M D = 45% C, 37% L	Semester 20 hr	School	Alice; Storytelling Alice	CC; CP
Werner & Denner (2009)	126	Middle school G = 100% F D = 58% C, 31% L	12 weeks	School	Flash	CP
Werner, Denner, Bliesner, & Rex (2009)	22	Middle school G = 45% F, 55% M D = 27% C, 68% L, 9% AA	2 weeks (2 hours per day)	Summer	Storytelling Alice	CC; CL
Werner, Denner, Campe, & Kawamato (2012)	325	Middle school G = 36% F, 64% M D = 52% C; 37% L	20 hr	After school	Alice; Storytelling Alice	CP
Yang & Chang (2013)	67	Middle school G = 49% F, 51% M D = n/a	19 weeks	School	Flash	CP; CPe

Note. Gender: F = female; M = male; Race: AA = African American; C = Caucasian; L = Latina/o; A = Asian; AI = American Indian; M = mixed;

CC = computational concepts; CP = computational strategies; CPe = computational perspectives; CO = content, academic subject; CL = learning, disposition, problem solving.