Education Tech Research Dev (2019) 67:1–19
https://doi.org/10.1007/s11423-018-9596-8

RESEARCH ARTICLE

Teaching systems thinking through game design

Mete Akcaoglu[1] · **Lucy Santos Green**[2]

Published online: 9 May 2018

Abstract In this mixed-methods study, we examined if students benefitted from a game design course offered during an enrichment hour in terms of gains in their system analysis and design skills. Students at a rural middle school in Southeast US (n = 19) attended a 1-hour game design course offered weekly during an academic enrichment class period, for the duration of a school year, learning basics of digital game-design and practicing system design skills such as making flowcharts. The results of quasi-experimental data indicated that the treatment group's pretest–posttest system analysis and design skills, compared to the control group, which did not receive any training, further improved, $F(1,33) = 16.516$, $p < 0.001$. Results from the interviews showed that the participants were able to verbalize how they applied system analysis and design skills developed during the course to problem-solving in different contexts. We discussed the instructional aspects of learning game-design that align with systems thinking. We also explored the possible influence of initial cognitive skills on student learning outcomes from such interventions.

Keywords Game design · Problem solving · Systems thinking · System design · Design

✉ Mete Akcaoglu
makcaoglu@georgiasouthern.edu

Lucy Santos Green
llgreen2@mailbox.sc.edu

[1] Georgia Southern University, 275 COE Drive, Room 3113, Statesboro, GA 30461, USA

[2] University of South Carolina, 1501 Greene Street, Columbia, SC 29208, USA

Introduction

A system is a unit, built by nature or by humans, that functions as a whole through the dynamic and complex interactions of its parts (Salen et al. 2011). Within the context of building and understanding systems, *systems thinking,* is defined as the cognitive ability to identify connections between parts of a system; understanding how the parts come together to function as a whole (Assaraf and Orion 2005; Senge 1990). In a broader sense, systems thinking is a collection of theories and models that consider the relationships between "agents and structures operating within a bounded context" (Norman 2013, p. 1087). It involves recognizing patterns among variables that make up a system and learning how to reorganize these into more effective forms. Systemic thinkers have the ability to unearth the hidden dimensions of a system through an understanding of the whole.

In recent years, systems thinking has been lauded as an important higher-order thinking skill, believed to aid individuals in identifying methods for complex problem solving (Goodman and Stroh 2008; Salen et al. 2011). Previous research conducted in this area indicates that systems thinking, as a cognitive skill, can be taught at all levels (Cabrera et al. 2015). Problem-based learning theories (Brown et al. 1989), among others, suggest that *design* tasks provide learners with contextualized and authentic learning experiences where systems thinking skills may be nurtured and developed. In addition, completing these types of design tasks gives students the opportunity to use real world-based skills and domain knowledge that may be applied to situations in the future (de Vries 2006). Design learning gives learners the opportunity to conduct investigations, solve problems through implementation of knowledge and skills, reflect on these problem-solving experiences, and lastly, participate in self-explanation and communication exercises. Consequently, design arguably helps learners to use active and collaborative learning (Ke 2014). Unfortunately, activities that allow students to tackle complex and ill-structured problems through design are not usually a part of formal school curricula (Jonassen 2000).

As an engaging design task, game-design has recently been used in informal instructional contexts as a way to teach students thinking skills, including problem-solving, and computer science concepts (e.g., Akcaoglu and Koehler 2014; Denner et al. 2012; Hwang et al. 2013). Games are interactive systems (Fullerton 2008) and designing games requires building systems. Recently, researchers identified connections between designing digital games and cognitive and motivational outcomes (Kafai and Burke 2015) such as the ability to learn programming (Baytak and Land 2010, 2011), computational thinking concepts (Denner et al. 2012), perceptions of learning and problem-solving (Hwang et al. 2013), improved attitudes toward math (Ke 2014), and solving problems, including complex ones (Akcaoglu et al. 2016; Akcaoglu 2014; Akcaoglu and Koehler 2014). This body of work strongly suggests that the game-design process requires implementation of "systems thinking, problem solving, critical and creative thinking, storytelling, programming, and visual literacy" (An 2016, p. 565).

The purpose of the study was to explore the interplay between systems thinking as a problem-solving skill, and design tasks through an investigation of the cognitive outcomes resulting from a game-design enrichment course offered to middle school students at a rural school in the Southeast region of the United States. Specifically, we examined if completing a game-design course with explicit connections to designing and building systems, impacted students' ability to design and analyze systems—their systems thinking skills. In the next section, we will first define problem solving. Following this, we will situate system analysis and design as a problem-solving skill, detail systems thinking, and finally, explore the connections between game-design tasks and systems thinking skills.

Background

Problem solving

Problem solving is a cognitive process that occurs when the problem solver attempts to transform a "given situation" into a "goal situation," despite the lack of an immediately perceivable solution (Mayer and Wittrock 2006, p. 287). Problem solving is an important cognitive task frequently executed in our personal and professional lives (Jonassen 2000). For children, the ability to problem solve is especially crucial because it establishes a foundation for all future learning, and "for effectively participating in society" (OECD 2003, p. 154). It does so by requiring that learners work through a process that demands both retention and transfer of new knowledge, two components necessary for meaningful learning (Mayer and Wittrock 2006). The ability to problem solve is further developed whenever a child is confronted with a new challenge since "problems are not equivalent, in content, form, or process" (Jonassen 2000, p. 65).

Problems are mainly categorized by the complexity of their structure as well-defined or ill-defined. Ill-defined or ill-structured problems, are problems in which "the given state, goal state, and/or allowable operators are not clearly specified" (Mayer and Wittrock 2006, p. 288). The types of problems this article primarily concerns itself with, design problems, fall under the category of ill-defined problems because these are usually open-ended, unstructured, or "wicked" (Nelson 2003).

Design problems are well-suited for the growth and development of systems thinking skills. A design problem requires that a student analyze a system, or design a solution, instead of selecting a 'right answer.' When confronted with a design problem, a designer cycles through five phases: (1) identification and analysis of the problem, (2) inquiry and research of the problem, (3) identification and ranking of design priorities, (4) design testing, and (5) evaluation (Nelson 2003). Within design problems, a system analysis reveals a dynamic and flexible relationship between the variables identified (OECD 2003). Jonassen (2000) explains "human problem solvers construct a mental representation (or mental model) of the problem, known as the problem space…[however] the real problem-solving activity involved with ill-structured problems is providing a problem with structure when there is none apparent" (pp. 65–67). The very act of defining and establishing a system, becomes the first problem to be solved before designing (and its subsequent challenges) can be addressed. Systems thinking as problem solving reveals itself to be an even richer learning experience when one considers that design problems cannot be prescribed, but instead leads to emergent problem solving (Nelson 2003).

Systems, design, and systems thinking

A system is "a group of interacting, interrelated, or interdependent elements forming a complex whole" (Salen and Zimmerman 2004, p. 50). According to Salen and Zimmerman, systems are composed of four elements: (a) objects or variables, (b) attributes, (c) internal relationships, and (d) context. For example, in a game (i.e., system) such as PacMan, objects are the characters (e.g., PacMan, ghosts) and other items in the game (e.g., fruits, pellets). These objects are in an interaction with each other. In PacMan, different objects have different attributes which lead to differences in outcomes in the gameplay (e.g., eating fruits can give you immunity for a certain amount of time from the enemies in the game). As for internal relationships, these exist in different formats: spatially, socially, psychologically, or emotionally. In PacMan, the underlying game-system, as well as the state of the player who is playing the game, constitute the internal relationships. For example, the increasing difficulty of the game from one level to the next is determined by the player's skills, as well as the underlying algorithm of the game system. Finally, systems exist within contexts. In PacMan the "play" context contains both the software system and the players. As argued by Salen and Zimmerman, "[Systems] do not exist in a vacuum but are affected by their surroundings" (p. 51). Systems can be natural or artificial (e.g., human-made): human body versus robots. Systems can be simple (e.g., can opener) or complex (e.g., language). Regardless, when a system operates, its components work together to reach the desired goal (Fullerton 2008).

Systems thinking is defined as the cognitive ability to identify relationships between objects (or variables) of a system and to synthesize these as a whole (Assaraf and Orion 2005). Systems thinking includes other component cognitive skills such as (a) understanding dynamic relationships between variables (e.g., feedback loops), (b) understanding that multiple relationships exist among variables and these can be impacted and changed by time, (c) understanding that, in addition to overt relationships, there can be hidden (possibly nonlinear) relationships among system variables, (d) being able to analyze a system's efficiency, and (e) scientific thinking, which involves hypothesizing and testing theories (Assaraf and Orion 2005; Salen et al. 2011; Sweeney and Sterman 2000).

Systems thinking is an important cognitive asset for professional success in the fields of science, technology, engineering, and math. These fields require individuals to design and implement systems, find and solve complex and ill-structured problems, estimate and predict outcomes, and conduct experiments (Dym et al. 2005). Systems thinking is also listed as a key skill within most 21st century skills frameworks (e.g., Partnership for 21st Century Learning).

Teaching systems thinking can be difficult due to the complexity inherent in understanding and designing most systems (Assaraf and Orion 2005; Hung 2008). The difficulty in mastering systems thinking stems from the abstractness, and complexity (e.g., nonlinear) of inter-causal relationships among variables. Jonassen (1997) explains that expert systems thinkers build additional complex "schemas which they can apply in a more proceduralized or automatized manner" (p. 67). In contrast, beginners refer to schemas lacking in rich detail, so that they are forced to rely on generalized approaches. This lack of cyclic or dynamic thinking can deter learners of all ages from becoming effective systems thinkers (Assaraf and Orion 2005; Hung 2008).

Despite these challenges, educators have reported success in the employment of instructional innovations that target improving systems thinking (Cabrera et al. 2015). For example, Hung (2008) found that a semester-long course on systems theory and its practical applications, where simulation modeling software was used, helped students increasingly use "systems thinking by reasoning through the interrelationships, causal relationships and feedback processes" (p. 1110). Hung argued that systems thinking can be taught through explicit instruction, suitable cognitive tools (e.g., software should be user-friendly), and a gradual progression from modelling simple to more complex systems. In another study, Kali et al. (2003) found that through a short (4-hour) knowledge integration activity, students were able to make modest improvements in their systems thinking skills. The authors suggested that different components of a system and then, gradually, integrating these back into the larger system can be an effective instructional strategy to teach systems thinking. Finally, Assaraf and Orion (2005) found that through instruction that is carefully sequenced (hierarchically) systems thinking can be taught. They argued that students' individual cognitive abilities, as well as their level of active participation during knowledge integration activities were two factors that led to differences among student performances.

Game design and systems thinking

Games are complex systems, composed of various elements that interact with one another in many different and complex ways (Crawford 1984, 2003; Fullerton 2008; Salen and Zimmerman 2004). The process of designing games requires game designers to plan, sketch out, and create, often very complex systems. According to Crawford (1984), "the central problem in designing the game structure is figuring out how to distill the fantasy of the goal and topic into a workable system" (p. 55). Jonassen (2000) further argued that "because the criteria for acceptable solutions are not always obvious [in a design problem], designers must construct personalized systems for evaluating their products" (p. 80). Both emphasize the identification and establishment of a system as a core component of the game-design process. During this process, designers experience how systems work, which helps them to learn how to "think analytically and holistically, to experiment and test out theories, and to consider other people as part of the systems they create and inhabit" (Salen et al. 2011, p. 56).

Games in educational contexts have attracted the attention of both educators and researchers (An 2016; Kafai and Burke 2015; Mayer 2016). However, the outcomes from using commercially-available educational games for teaching and learning purposes yielded mixed results (Kafai and Burke 2015; Mayer 2016). According to a recent review by Mayer (2016), the conditions in which the effectiveness of educational games increase, widely vary, warranting further investigation. In contrast, getting young learners to design their own digital games, rather than simply playing commercially-available products, provides students with the opportunity to develop skills in system analysis and design, problem solving, and computer programming (Akcaoglu 2016; Kafai and Burke 2015). Using software such as Agentsheets, Alice 3D, Gamemaker, Kodu, or Scratch, children create and play digital games; a process that has been found to be an effective method for teaching not just computer programming (Werner et al. 2014) and problem solving (Akcaoglu 2014; Akcaoglu and Koehler 2014), but also content knowledge in areas such as health and math (Baytak and Land 2010; Ke 2014).

The process of designing digital games is beneficial for several reasons. First, digital game-design requires young learners to not only analyze and understand systems, but to create these from scratch (An 2016). Second, there are personal, social, and cultural benefits from engaging in game-design tasks (Kafai and Burke 2015). From this perspective, during making, children create visual representations of their worlds as they see it, and this creation, this vision, is publicly shared through their games. The ability to articulate one's vision to an external audience not only aids in a more holistic assessment of learning (Akcaoglu and Koehler 2014; OECD 2003); but serves a key function in the problem-solving process: clearly articulating the solution to those unfamiliar with the problem in the first place (Jonassen 2000).

Third, game-design tasks are good contexts to practice and strengthen problem solving skills since the initial steps in game design echo the initial steps in problem solving including analyzing and understanding the underlying patterns and mechanisms in a given situation (Akcaoglu 2014; Akcaoglu and Koehler 2014; Ke 2014; Polya 1957). Game design is a problem "represented in its natural complexity and modality" (Jonassen 2000, p. 69) that draws on "prior knowledge, merging content areas and requiring the integration of concepts, representations and processes" (OECD 2003, p. 156). In doing so, it causes students to struggle with the construction and deconstruction of a new system, and with the design tribulations it brings, in an authentic setting (Jonassen 2000).

The current study

Designing and building systems is a key skill for science, technology, engineering, and math careers. Through this design process people learn to identify problems, plan solutions, and use higher-order thinking skills (Honey and Kanter 2013). Designing digital games has been established as a successful instructional context and activity for teaching and learning academic skills such as computer programming and problem solving. Although we know designing games is an engaging activity, which involves designing complex systems, we do not know if this practice leads to improved system design and thinking skills. The purpose of this research study was to understand if middle school students who attended a game design course showed improvements in their system analysis and design skills. More specifically, our research questions were:

1. Does attending a game-design course impact students' system analysis and design skills, compared to a control group?
2. How do students describe their approach to problem solving within the context of game-design?

Methods

In order to answer the research questions, data was collected and analyzed in two ways: quantitative and qualitative. To answer the first research question, a quasi-experimental design was used, where students who attended a game-design course were compared to students who did not attend the game-design course. To answer the second research question, semi-structured interviews were conducted with game-design students, and the transcripts were qualitatively analyzed to identify common themes.

Participants and context

Participants in this study were 6th grade students (age average = 11) at a middle school located in the Southeast region of the United States. The students who attended the game-design course (i.e., the experimental group) were students who scored in the top 25% of their class in math and science (n = 20, male = 12, female = 8). Although the researchers were not specifically looking for an academically talented group, the school teacher for this group indicated an interest for a game-design course during the school's enrichment period. The enrichment period was a once-a-week, hour-long session built into a middle school's master schedule. This block of time allowed teachers the opportunity to provide enrichment and remediation for all students in each academic subject. Students in our experimental group attended the game design sessions during this period throughout the school year. The students in the control group (n =19, male = 10, female = 9) were randomly picked from students in the same class as the experimental group, who showed an interest and volunteered to participate in the study. Due to missing data, the analysis was conducted with 16 control group and 19 experimental group students. All students involved with this study obtained prior parent approval for participation.

Game design and learning course

During the game-design courses, Microsoft Kodu, a freely available software, was used. Kodu, like its peer software (e.g., Scratch) allows for visual programming. In fact, Kodu's main purpose is to "help users learn computer science concepts through game creation" (Stolee and Fristoe 2011, p. 100). Compared with other software serving similar purposes, Kodu is more visually appealing due to its 3D environment, its capacity for various levels of complexities during design in terms of computer programming concepts, and its affordance in letting users create different types of games (e.g., adventure, arcade, racing) and simulations (Akcaoglu 2016).

A total of 10 game design sessions (10 h) were offered to the experimental group during the Fall semester. The course was built based on previous research (i.e., Akcaoglu 2016) and followed a similar format: purely game-design sessions, problem-solving sessions, troubleshooting sessions, and free-design sessions. However, during this study, sessions devoted to troubleshooting activities were not implemented. The first author was the lead instructor of the game-design courses, while the second author provided support by helping with classroom organization and answering student questions during their free-design time. A more detailed description of the Game Design and Learning courses, as well as its theoretical underpinnings can be found elsewhere (Akcaoglu 2016).

During the game-design course, due to the short nature of each session (around 50 min), six sessions were spent on teaching the basics of game design, which included covering topics related to main components of games (Goals, Rules, Assets, Scoring, Mechanics, and Spaces). During the first six lessons, students recreated popular game types: Apple Hunter, PacMan, and Kodu Adventure. Coding concepts, from basic to more advanced, were also introduced during these weeks. For example, while the students learned about and practiced basic coding structure within the game design software (if–then statements), they learned about nesting codes and creating different behaviors for different characters based on the game state. The following sessions in the semester were dedicated to planning and designing games. These games were usually versions of the earlier games that students created during the course, or games that they were familiar with based on their experiences outside of school. In order to engage students with real-life design experiences, students were also asked to create flowcharts for their games before and after each game-design activity (Fig. 1). During these flowcharting exercises, the students tried to depict the connections between the objects in their games to understand how these objects and connections worked as systems, giving students a chance to practice their systems thinking skills.

Data sources and instruments

Two types of data were collected during this study. Quantitative data was collected by using an internationally validated test (Programme for International Student Assessment: PISA) of system analysis and design developed and distributed by the Organisation for Economic Co-operation and Development (OECD 2003). Qualitative data was collected (recorded and transcribed) through semi-structured interviews conducted by the lead authors of this study.

Based on Mayer and Wittrock's (2006) conceptualization of problem-solving, the system analysis and design section of the 2003 PISA problem-solving assessment focuses on a student's ability to identify relationships in a system, and his or her ability to design a

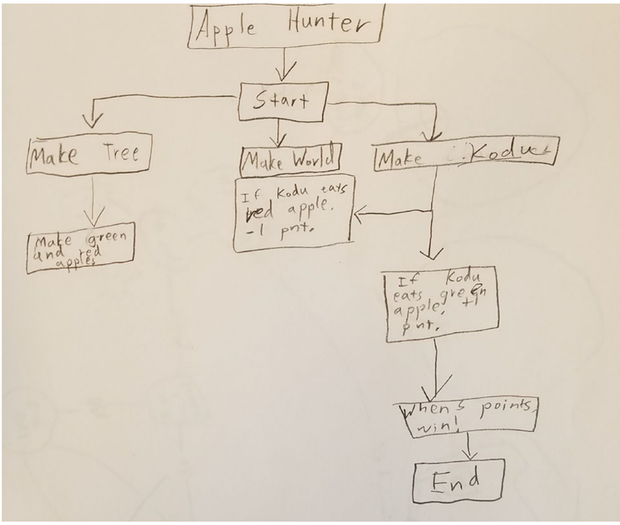


Fig. 1 Example student flowchart of the game "Apple Hunter."

system where there are dynamic and complex relationships among its parts. System analysis and design is characterized as tasks that "require understanding the complex relationships among a number of interdependent variables, identifying their critical features or applying given representation, analyzing a complex situation or designing a system so that certain goals are achieved" (OECD 2003, p. 167).

The section of the test used in this study contained seven questions. There were two multiple-choice questions, one short answer question, and four open-ended questions that required students design, develop, or demonstrate understanding of a system. As an example, one open-ended question, the *library system* question, asked students to draw the library system flowchart of an imaginary school based on loan rules of the library. Sample correct and incorrect student responses can be seen in Fig. 2. In the first picture, we see an incorrect answer: the student does not fully show *Greenwood High School*'s library system. In the second picture, the student's correct answer fully depicts how the library system at the school works, representing the relationships among the rules and components of the loan system. As demonstrated in this example, the test items were designed to assess (a) the ability to identify relationships between objects (or variables) and synthesize these as a whole by understanding dynamic relationships between variables (e.g., through flowcharts), (b) understanding that multiple relationships exist among variables and these can be impacted and changed by time, (c) understanding that, in addition to overt relationships, there can be hidden (possibly nonlinear) relationships among system variables, (d) being able to analyze a system's efficiency, and (e) scientific thinking, which involves hypothesizing and testing theories (Assaraf and Orion 2005; Salen et al. 2011; Sweeney and Sterman 2000).

Procedures

Quantitative data (problem-solving testing) for this study was collected during the first and the last sessions of the game-design course. The problem-solving tests were administered as a paper and pencil test, taking students approximately 30 min to complete. After

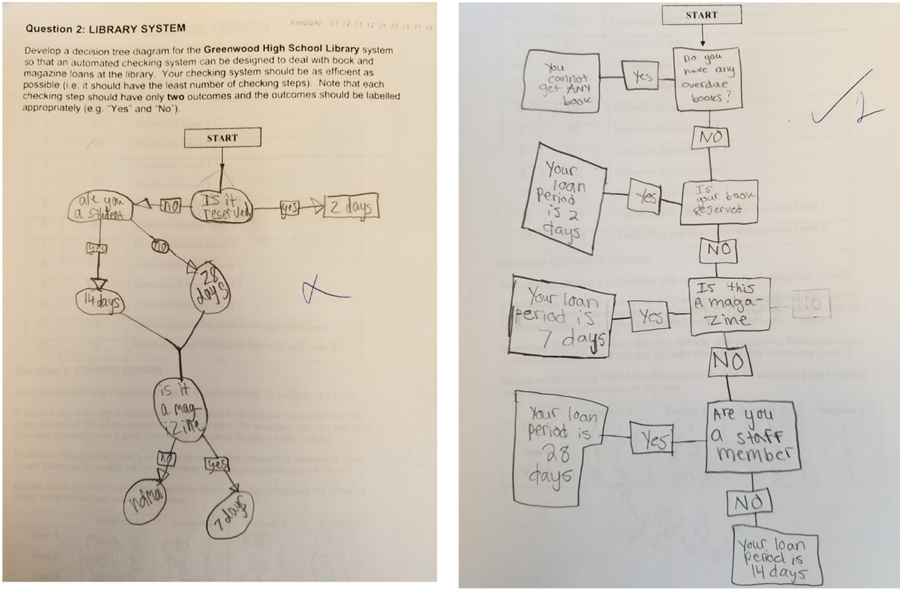


Fig. 2 Incorrect (left) and correct (right) responses to the library question

completing the test, the students carried on their game-design work for those sessions. The control group did not attend any game-design sessions. They took the same problem-solving test as the game-design group on the same days.

Qualitative data, in the form of semi-structured interviews, was collected following the completion of the game-design courses. Students with the most and the least gains in problem solving, as determined by the problem-solving test scores, were purposefully selected to be individually interviewed. Each interview, conducted by the authors, lasted approximately 20 min. Interviews were digitally recorded and transcribed for analysis. Students were asked to describe and reflect on their experiences and thoughts about their approaches to problem solving, game design, programming, and their overall experience as participants.

Data analysis

A graduate assistant graded the students' performance assessment using an answer key with established interrater reliability and validity by OECD (2005). The students' group membership (treatment vs. control) was blinded during the grading process. In order to provide a better interpretation of the students' performances (linearly), transformation was applied to students' performance scores to convert them into percentages of a maximum possible (POMP) score that could be attained (Cohen et al. 1999), where 0 was the lowest possible score and 100 was the highest possible score. Since scores were linearly transformed, the results of further statistical analysis with the data remained constant (Cohen et al. 1999). The reliability of the test was acceptable, $a = 0.72$.

Based on their performance on the problem-solving assessment, seven participants from the experimental group were purposefully selected for interviews. More specifically, in order to collect a diverse set of experiences and responses, based on their gains in the performance assessment, participants who showed more than one standard deviation (n = 4), less than one standard deviation (n = 1) improvement, and students who showed no improvement (n = 2) were targeted. During the interviews, the goal was to understand how students approached solving problems and if their experiences in the game-design courses helped their approach to problem solving, their likes and dislikes regarding the course and the software, their approach to programming their game design process, and their interest in game designing outside of school. The interview questions can be found in Appendix.

Once transcribed, the two lead authors independently applied qualitative content analysis to the interview data, thus engaging in a "subjective interpretation of the content of text data through the systematic classification process of coding and identifying themes or patterns" (Hsieh and Shannon 2005, p. 1278). Since there was not an existing theoretical framework to guide our coding efforts, we used an inductive coding approach, completing the first three steps identified by Mayring (2000): (a) revisiting the research question, (b) a determination of categories, and (c) development of inductive categories from the material. Following this, the researchers met to discuss their initial codes. The researchers then completed the last three steps of inductive category development: (d) revision of categories, (e) final working through text, and (f) the interpretation of results. During the final step, the researchers discussed any discrepancies until a 100% agreement was reached, and thematic content analysis completed.

Results

Quantitative data results

In order to understand if there were initial performance differences between the control and the treatment groups an independent-samples t test was conducted on students' pretest scores. This analysis was an important first step because the two groups were not chosen randomly (i.e., a quasi-experimental research design). The pretest average for the experimental group was higher (M = 34.1, SD = 15.9) than the control group (M = 20.1, SD = 16.5), indicating also a statistically significant difference between the two groups: $t(37) = 2.701$, $p = 0.01$.

Next, in order to understand if the two groups (i.e., control vs. treatment) showed different gains from pretest to posttest, a repeated-measures analysis of variance (RM-ANOVA), having two levels of time (pre vs. post) as within-subjects factors, and two levels of group (control vs. treatment) as between-subjects factor was conducted. The multivariate omnibus for time was significant (Wilks's $\Lambda = 0.87$), F $(1, 33) = 4.93$, $p = 0.033$, $\eta^2 = 0.13$, as was the time-group interaction (Wilks's $\Lambda = 0.78$), $F\,(1, 33) = 8.5$, $p = 0.006$, $\eta^2 = 0.21$. The results indicated that the change from pretest to posttest was significantly different for the two groups.

Table 1 Descriptive statistics for the control and experimental groups

Group (n)	Pretest		Posttest	
	M	SD	M	SD
Control (16)	22.16	16.92	19.89	16.34
Treatment (19)	33.97	16.28	50.72	22.13

Follow up between-subjects test confirmed this result, $F(1,33) = 16.52$, $p < 0.001$, $\eta^2 = 0.5$. The results favored the treatment group (Table 1). In fact, while scores for the control group went down negligibly, $t(16) = 0.47$, $p = 0.65$, there was a statistically significant gain in the treatment group's scores, $t(19) = 3.84$, $p = 0.001$, with a large effect size, Cohen's $d = 0.88$.

More specifically, it was seen that the students in the treatment group were able to analyze the depicted systems in the test and recreate them more efficiently (See Fig. 2a, b). For example, in the library system question, students more efficiently identified the checkpoints in the system and narrowed the flow down from more general attributes to more specific ones to produce a more efficient system.

Qualitative results

The following section describes the themes identified by the researchers based upon an analysis of the transcript of interviewee responses: (a) problem solving during the design process, (b) help seeking during problem solving, and (c) the desire for personalization. These three themes reflect commonalities in responses of 6th grade study participants such as overcoming challenges, applying problem solving skills to new contexts, preferences on the type of external support students sought, and the creative limitations of Kodu.

Theme 1: challenges in the design process

The first theme, challenges in the design process, relates to ways students approached and overcame design challenge problems while designing and developing their games. Problems experienced by the student designers generally fell into two camps: problems generated by unfamiliarity with Kodu and its menus, and problems generated by a conflict between design and execution. For the purposes of this paper, challenges in the design process focused on the second camp, the tension between what a student designed during the planning phase, and how the student ultimately executed his or her design plan. Sometimes, this tension became evident when students tackled smaller design features such as an object's behavior on land or in a body of water. In the following quote, the student describes how a design was modified from the original plan, so that the object would remain in one place:

> Well, like, I had this rock. I put it on top of a hill but it kept rolling down, so at first, I tried to make it unrollable [*sic*] or something like that and it still kept rolling down for some apparent reason, so I just changed it to a tree.

In other instances, students were overwhelmed by the complexity of their own designs. Their original plans called for developing a system much too robust for their initial game-design skills. Therefore, the design had to be trimmed down: "I wanted to add more levels but then you'd have to create another world and it would be really complicated to do that—just to have to create more characters and such." These quotes exemplify typical student approaches to problem solving design challenges that were observed and recorded. In the majority of these instances, students quickly modified designs by identifying and selecting alternate features and objects that behaved in a desired way within Kodu; or, students pared down the design—thus simplifying the overall system as well as its interconnected parts.

Theme one also highlights how students connected problem solving skills learned during the game design process to other contexts. Several students voiced a relationship between their experience problem solving design challenges during video game design, and their approach to problem solving questions on the problem-solving assessment. Misty (pseudonym) explained how she worked out a problem differently by thinking of it as she did the menus and submenus in Kodu, creating a flowchart to make better sense of a word problem:

> Misty: During the coding of the characters, I did have to sort of use a flowchart [air quotes] with the dragging under and the yes's and no's on there.
> Interviewer: Can you explain to me how that helped you solve this problem [points to word problem on the PISA test]?
> Misty: I solved the problem by looking at it differently. I just, like, followed the instructions but I sorted through the random pictures that I put words inside.
> Interviewer: Did you reproduce a flowchart?
> Misty: Sort of. I made it a little different. It's more complicated than this [points to her original video game design flowchart].

Another student, Marcus, pointed out:

> If you don't know what to put down, the order is only a minor part. The more I designed games, the more I noticed that if you don't know what to do, then there's no point in having an order to anything.

Facing design challenge problems when game-designing helped Marcus understand that tackling any complex problem requires knowing how each object or function behaves in a system (in the case of the system design assessment) before one can establish order, restructure or add to that system—an insight he applied to solving problems on the PISA test.

Theme 2: help-seeking patterns

The second theme, help-seeking patterns, refers to student preferences as to when and how to look for external support during problem solving and design. When engaged in help-seeking, some students followed a predictable pattern that began by attempting an alternate solution first: "I usually had a plan B on how things would work…" followed by consultation with peers, "…and if that didn't work, I usually tried to get help from one of my friends…" and finally, the instructors, "…or you guys." Other students prioritized efficiency and availability over the predictable pattern just described:

> It didn't really matter. I just needed some help with it at the time, so I just asked either my friends or you. It didn't really matter who. Whoever was available, basically, because if I interrupted someone else, it would make it harder for you guys to go around. There weren't as many teachers.

A third group of students placed importance on level of expertise, reasoning that the instructors would offer a higher level of support: "I would rather ask a teacher since they probably know more about it. They have a bigger chance of helping me."

As the semester progressed, we observed that more and more students followed the pattern of the first help-seeking group. They began to identify classmates designing similar games, or students that had implemented a unique solution. Then, they approached these peers, asking for specific and targeted help. This strategic approach to help-seeking indicated student growth in problem solving abilities in two specific ways. First, students were able to meaningfully reflect on their previous problem-solving experiences (e.g., waiting on a teacher versus targeting a classmate, logging a solution implemented by a student designing a similar game), using these reflections to modify their help-seeking behaviors. Second, students established an active and collaborative learning space (Ke 2014), participating in self-explanation and communication (e.g., changing seats and organically grouping themselves according to similarity in game designs, playing and critiquing each other's games).

Theme 3: the search for personalization

The search for personalization, the third theme gleaned from interviews, was an emergent finding that warrants further investigation in a future study. It reflects student enjoyment of the creative aspects of Kodu, while simultaneously labeling lack of customization options as a major program limitation. While several students in the course had experience with other game design platforms such as Scratch, this was the first time any participants designed with Kodu. Some students, like Abby, derived enjoyment from the ability to select patterns and textures, as well as character traits:

> I like how in Kodu we got to use art and different patterns and more, like sort of make the mountains. I got to use the virtual world as more of my canvas. I like how we did the characters with it and how we coded the characters. Like one of them would go really slow and the other would go really fast.

For students such as Abby, the creative component was the most enthralling aspect of design: "It was more of the art in it that I wanted to use, so I just wanted to do the art and the technology more than the code."

For students like Bobby, who came to us with a broader background and interest in game design and computer programming, the creative aspects of Kodu failed to make up for perceived software limitations:

> Bobby: I didn't like that the Kodu game platform limited you a lot more than other platforms.
> Interviewer: What other platforms have you worked with?
> Bobby: Game Salad, Flash, Unity, and…Javascript.
> Interviewer: You do have a lot of solid experience there. Cool. Any specific way that Kodu limits you?
> Bobby: It's just—it doesn't have that much objects to put in and you can't create your own objects like you can in almost every other game-making platform that I know.

The desire to create and further customize characters and character behavior came up frequently. Another student declared: "I wish you could make your own character. Like, design your character and not have to pick out a character that was already programmed in the program." A third student, Ginger, described her frustration: "You couldn't do certain things. Like, if you had to run, you couldn't do it. You couldn't get out of the car, get in the car. You had to find a way."

Despite a general frustration with a perceived lack of personalization, as students described what they wanted to be able to accomplish with their game designs during their interviews, several concluded the inability to customize was mostly due to a lack of experience with Kodu. Jacob shared, "I feel like there are so many things that I didn't learn and if I did learn those I might be able to go back into my game and change a few things and make it better." Even Bobby, one of our more experienced students, recognized his lack of familiarity with Kodu as a barrier to customization and creative design:

> I could've maybe tried to do stuff more at home to understand it more, which, I can still do that, but it would be better if I did it along with the course, which I didn't. Like, in my game that I tried to make, I tried to make it to where it spawned one enemy at a time and not all the enemies. I wanted to spawn them separately and I didn't know how to do that.

While, on its face, the ability to customize a game may seem unrelated to the development of systems thinking through problem solving; student awareness of their ability limitations with Kodu, as well as the impact of these limitations on their original designs, resulted in further definition and establishment of their game design systems. As previously discussed, the establishment of a system is the first problem to be solved before designing (and its subsequent challenges) can be addressed (Nelson 2003).

Discussion

The purpose of this study was to examine if middle-school students showed gains in their problem-solving skills in system analysis and design, after participating in a game-design course. Compared to students in the control group, course participants showed statistically and practically (i.e., large effect size) significant gains in their system analysis and design skills. Previous research in this area has determined that experience with game-design is effective in improving young learners' problem-solving skills (Akcaoglu 2014; Akcaoglu and Koehler 2014), deepening understanding and execution of programming concepts (Denner et al. 2012), fostering computational learning (Werner et al. 2014), and developing content knowledge (Baytak and Land 2010). The results from this study refine and further support a possible connection between learning game-design and systems thinking skills, as a possible outcome from practicing analyzing and designing systems through activities embedded in game-design.

Interviewed participants were also able to verbalize how they applied system analysis and design skills developed during the course to problem-solving in different contexts. Consequently, both quantitative and qualitative findings suggest that the game-design course offered in this study was an effective instructional structure for teaching system analysis and design, as well as problem solving skills to middle school students, adding to the growing body of research reporting positive outcomes in game-design activities for young learners (Kafai and Burke 2015).

Instructional approaches to foster systems thinking should help students make explicit connections to systems theory (Hung 2008). During our game-design course, as evidenced in transcripts of Misty and Marcus describing the manipulation objects and other information within Kodu, students were able to see how games functioned as systems. Hung (2008) argued that tools used during teaching of systems thinking should be user-friendly and transparent, supporting the characteristics of systems thinking. We believe most game-design software, including the one used in our study (Microsoft Kodu), are user-friendly and transparent, allowing students, based on their own cognitive skills, to design systems in increasingly complex ways while developing an understanding of the relationships among system variables. As a result of these game-design platform characteristics, and as notated in student interviewee commentary, participants were able to conclude that creating games first required understanding and designing of systems and system components.

As stated by both Kali et al. (2003) and Hung (2008), another important feature of instructional settings to teach systems thinking is that teaching should follow a gradual trajectory from introduction of simple to more complex systems. This approach allows knowledge building of a system's parts so that students are able to understand systems in a more holistic fashion (Kali et al. 2003). During the game-design course implemented in this study, and while also making explicit connections to systems theory, we followed the structure used previously by Akcaoglu (2016) and organized our course so that students began the semester designing and building simple games (i.e., systems). Later, as the course progressed through the semester, students designed games that were more and more complex.

In addition to the instructional approaches to teaching systems thinking above, we encouraged students to practice visually analyzing systems by having them create causal maps before and after each game that they designed. Causal mapping refers to visually depicting the relationships among the variables of a problem (Öllinger et al. 2015). Causal mapping helps problem solvers identify potential solution pathways through the creation of external representations of the underlying structure (i.e., system) present in complex problems (Öllinger et al. 2015). Creating external representations of complex problems helps problem solvers shape their perceptions of the problem, identifying the necessary components required to solve the problem, or design the system (Zhang 1997).

Causal mapping also helps students "overcome the difficulty of dealing with the abstractness and non-perceivable properties and relationships" (Hung 2008, p. 1101). During the game-design course, students' causal maps not only aided in their understanding of the relationships that existed in a system; but these external representations may have also helped students overcome a difficulty inherent to the design process: visualizing concrete relationships among multiple variables (Fig. 1). The process of mapping a system gave our students "an opportunity to evaluate the accuracy and the representative nature of their mental models" (Hung 2008, p. 1012).

The increasingly complex games that students created at the end of each weekly learning task also served as dynamic external representations. Nelson (2003) describes how designers evaluate each decision or move they make in three ways: by examining the consequences of a decision or move, by evaluating how a move impacts or violates earlier decisions or moves, and by perceiving what new problems or opportunities the move creates. These "working games" became visual models that gave our student participants opportunities to evaluate design decisions and potential problem solutions (or moves) in the three ways described by Nelson. Like the causal mapping process, these also required student game designers to "understand not only the individual parts and their respective inter-causal and feedback-loop relationships, but also the underlying mechanism of the system as a whole" (Hung 2008, p. 1011).

Limitations and future research

As noted in the results section, one of the biggest limitations of this study was that the students in the experimental group had significantly higher pretest scores than students in the control group. This may indicate that the students who participated in the game-design course had higher levels of cognitive abilities in problem solving and design tasks. As noted by Assraf and Orion (2005), and Hung (2008), student cognitive ability can be an important factor impacting learning experiences and outcomes. In fact, Hung (2008) found that "students who did well might have been more capable cognitively of performing systems thinking than the ones who did not do so well" (p. 1112), pointing to the possibility that, similarly, in our case, the experimental group may have been more capable and thus benefitted accordingly from the game design course. Asarraf and Orion (2005) explain that as a higher-order thinking skill, systems thinking is very sensitive to the cognitive abilities of individuals: learners with high initial cognitive skills can more easily perceive and manipulate relationships among a system's variables; while learners with low cognitive skills tend to struggle with identifying dynamic relationships in a system. Similarly, our results show that the treatment group, with initial higher cognitive abilities, might have benefitted more from the game-design course. It should be noted, however, that since the control group did not receive any systems thinking treatment, our results cannot directly explain this initial-level effect.

Another limitation of this study was the impossibility of measuring specific impacts from different learning activities on students' systems thinking skills. For example, although we believe the causal mapping activities were helpful in getting students to analyze and design systems and improve in their systems thinking, the study design and collected data did not allow for such granular analysis. Hence, empirical support for this connection was beyond the scope of this research and cannot be provided in this report. In a future study, the impact of such activities should be analyzed using experimental or quasi-experimental designs. Finally, conducting future studies with larger sample sizes can improve the study's power and generalizability.

Conclusions

Games are complex systems (Crawford 1984; Fullerton 2008; Salen and Zimmerman 2004). Designing games, therefore, requires analyzing, understanding, planning, and creating systems, making game-design learning activities may be good contexts for teaching higher-order thinking skills, such as problem solving and systems thinking. This study established a potential connection between the engaging task of game design, and the development of systems thinking skills, although it should be noted that this effect may not be as direct and uniform, as we explained in the limitations section. Adding to the body of work in this domain, (Akcaoglu 2014; Akcaoglu and Koehler 2014; Akcaoglu et al. 2016; An 2016; Denner et al. 2012; Kafai and Burke 2015), the findings of this study provide support for the increasing number of academic (cognitive and motivational) outcomes possible through the use of game-design as an instructional tool.

We suggest that future work on investigating outcomes from technology-rich interventions should benefit from using existing and agreed-upon cognitive skills, rather than targeting outcomes that bear similarities to these cognitive skills but currently, are largely unmeasurable. A focus on established higher-order thinking skills that are supported by well-established theories and research (e.g., problem solving, systems thinking) will aid in the design of effective instructional interventions that target measurable outcomes. Building on existing theories in such ways can also help us make progress toward understanding the outcomes of teaching in technology-rich environments, due to a unified understanding of these topics, as well as the methods for assessing them. In this manner, researchers and educators can design instruction that meets the goal of education: "helping students be able to apply what they learned to solve new problems" (Mayer and Wittrock 2006, p. 289).

Acknowledgements We would like to extend our sincerest gratitude to Dr. Tarcin for sharing his pearls of wisdom and unwavering support with us during the course of this research.

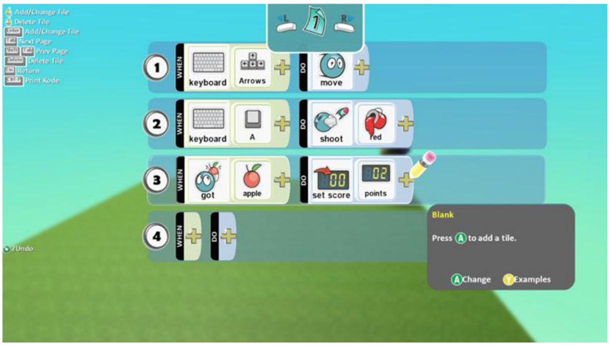
Compliance with ethical standard

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Interview questions

1. Questions about the *test*
    a. look at items and ask how and why they solved it the way they did (how did you get this answer)
    b. when you were learning how to create games, were there things that helped you in thinking how to solve problems like these?
2. Questions about their *game design, programming,* and *overall experience*
    a. what did you *like* or didn't like about the process of designing games?
    b. is there anything you wish you could *do* differently?
    c. is there anything that you wish you *knew* to make your game better?
    d. while making a game, when you are trying to *program characters in your game (SCREENSHOT ATTACHED)*, what do you usually do? (for example, if you run into a challenge, what was your first instinct: raise hand, ask friend, ask instructor, online?) Why?
    e. Do you (or did you) continue to create games outside school?
        i. was this the game that you were already working on?
        ii. use Kodu?
    f. Do you have any coding/programming experience (with any software/language)? Have you done anything like this before?



References

Akcaoglu, M. (2014). Learning problem-solving through making games. *Educational Technology Research and Development, 62*(5), 583–600. https://doi.org/10.1007/s11423-014-9347-4.

Akcaoglu, M. (2016). Design and implementation of the game-design and learning program. *TechTrends, 60*(2), 114–123. https://doi.org/10.1007/s11528-016-0022-y.

Akcaoglu, M., Gutierrez, A. P., Sonnleitner, P., & Hodges, C. B. (2016). Game design as a complex problem solving process. In R. Zheng & M. Gardner (Eds.), *Handbook of research on serious games for educational applications* (pp. 217–233). Hershey: IGN Publishing.

Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. *Computers and Education, 75,* 72–81. https://doi.org/10.1016/j.compedu.2014.02.003.

An, Y. J. (2016). A case study of educational computer game design by middle school students. *Educational Technology Research and Development, 64*(4), 555–571. https://doi.org/10.1007/s11423-016-9428-7.

Assaraf, O. B. Z., & Orion, N. (2005). Development of system thinking skills in the context of earth system education. *Journal of Research in Science Teaching, 42*(5), 518–560.

Baytak, A., & Land, S. M. (2010). A case study of educational game design by kids and for kids. *Procedia—Social and Behavioral Sciences, 2*(2), 5242–5246. https://doi.org/10.1016/j.sbspro.2010.03.853.

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educational Technology Research and Development, 59*(6), 765–782. https://doi.org/10.1007/s11423-010-9184-z.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32–42.

Cabrera, D., Cabrera, L., & Powers, E. (2015). A unifying theory of systems thinking with psychosocial applications. *Systems Research and Behavioral Science, 32*(5), 534–545.

Cohen, P., Cohen, J., Aiken, L. S., & West, S. G. (1999). The problem of units and the circumstance for POMP. *Multivariate Behavioral Research, 34*(3), 315–346.

Crawford, C. (1984). *The art of computer game design*. Berkeley: McGraw-Hill/Osborne Media.

Crawford, C. (2003). *Chris Crawford on game design*. Indiana: New Riders.

de Vries, E. (2006). Students' construction of external representations in design-based learning situations. *Learning and Instruction, 16,* 213–227.

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 and Education, 58*(1), 240–249.

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. *Journal of Engineering Education, 94*(1), 103–120.

Fullerton, T. (2008). *Game design workshop*. Boston: Elsevier.

Goodman, M., & Stroh, D. P. (2008). Importance of systems thinking. http://www.appliedsystemsthinking.com/.

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

Hsieh, H.-F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research, 15*(9), 1277–1288.

Hung, W. (2008). Enhancing systems-thinking skills with modelling. *British Journal of Educational Technology, 39*(6), 1099–1120.

Hwang, G.-J., Hung, C.-M., & Chen, N.-S. (2013). Improving learning achievements, motivations and problem-solving skills through a peer assessment-based game development approach. *Educational Technology Research and Development*. https://doi.org/10.1007/s11423-013-9320-7.

Jonassen, D. H. (1997). Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology Research and Design, 45,* 65–94.

Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Design, 48,* 63–85.

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist, 50*(4), 313–334.

Kali, Y., Orion, N., & Eylon, B. S. (2003). Effect of knowledge integration activities on students' perception of the Earth's crust as a cyclic system. *Journal of Research in Science Teaching, 40*(6), 545–565.

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 and Education, 73,* 26–39.

Mayer, R. E. (2016). What should be the role of computer games in education? *Policy Insights from the Behavioral and Brain Sciences, 3*(1), 20–26. https://doi.org/10.1177/2372732215621311.
Mayer, R. E., & Wittrock, M. C. (2006). Problem solving. In P. A. Alexander & P. H. Winne (Eds.), *Handbook of educational psychology* (pp. 287–303). Mahwah: Lawrence Erlbaum Associates.
Mayring, P. (2000). Qualitative content analysis. *Forum: Qualitative Social Research*. https://doi.org/10.17169/fqs-1.2.1089.
Nelson, W. A. (2003). Problem solving through design. *New Directions for Teaching and Learning, 95,* 39–44.
Norman, C. D. (2013). Teaching systems thinking and complexity theory in health sciences. *Journal of Evaluation in Clinical Practice, 19*(6), 1087–1089.
OECD. (2003). *PISA 2003 Assessment framework: Mathematics, reading, science and problem solving knowledge and skills*. Paris: OECD. https://www.oecd.org.
OECD. (2005). *PISA 2003 technical report*. Paris: OECD Publishing.
Öllinger, M., Hammon, S., von Grundherr, M., & Funke, J. (2015). Does visualization enhance complex problem solving? The effect of causal mapping on performance in the computer-based microworld Tailorshop. *Educational Technology Research and Development, 63*(4), 621–637.
Polya, G. (1957). *How to solve it*. Garden City: Doubleday/Anchor.
Salen, K., Torres, R., Wolozin, L., Rufo-Tepper, R., & Shapiro, A. (2011). *Quest to Learn: Developing the School for Digital Kids. Digital Media.* The John D. and Catherine T. MacArthur Foundation Reports on Digital Media and Learning.
Salen, K., & Zimmerman, E. (2004). *Rules of play*. Cambridge: MIT Press.
Senge, P. M. (1990). *The fifth discipline: The art and practice of the learning organization*. New York: Doubleday.
Stolee, K. T., & Fristoe, T. (2011). Expressing computer science concepts through Kodu game lab. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 99–104). ACM.
Sweeney, L. B., & Sterman, J. (2000). Bathtub dynamics: Initial results of a systems thinking inventory. *System Dynamics Review, 16*(4), 249–286. https://doi.org/10.1002/sdr.198.
Werner, L., Denner, J., & Campe, S. (2014). Children programming games: A strategy for measuring computational learning. *ACM Transactions on Computing Education, 14*(4), 24:1–24:22. https://doi.org/10.1145/2677091.
Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science, 21*(2), 179–217. https://doi.org/10.1207/s15516709cog2102_3.

Mete Akcaoglu is assistant professor of instructional technology in the Department of Leadership, Technology, and Human Development at Georgia Southern University. His research focuses on cognitive and motivational outcomes from innovative and technology-rich learning environments, such as after-school game-design courses.

Lucy Santos Green is Associate Professor of Library and Information Sciences at the University of South Carolina. She researchers and frequently publishes on librarians as instructional partners, the design of digital learning environments, and technology-enabled learning.