Education Tech Research Dev (2011) 59:765–782
DOI 10.1007/s11423-010-9184-z

RESEARCH ARTICLE

An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom

Ahmet Baytak · Susan M. Land

Published online: 7 February 2011

Abstract This study employed a case study design (Yin, Case study research, design and methods, 2009) to investigate the processes used by 5th graders to design and develop computer games within the context of their environmental science unit, using the theoretical framework of *constructionism.* Ten fifth graders designed computer games using *Scratch* software. The results showed students were able to design functional games, following a learning-by-design process of planning, designing, testing, and sharing. Observations revealed that game design led to opportunities for informal knowledge building and sharing among students. This, in turn, encouraged students to test and improve their designs. The findings support the conclusion that elementary students can develop programming concepts and create computer games when using graphical programming software developed for their level of experience. Insights into the iterative process of learning-by-game design are presented.

Keywords Children's learning · Game design · Constructionism · Programming · Environmental education · Educational technology

Gaming has traditionally been a part of children's everyday life as a method of play. Given the prominence of entertainment games for young children, interest has been correspondingly generated in the educational potential of computer games for classroom use

This work is based upon the doctoral dissertation completed by the first author at The Pennsylvania State University.

A. Baytak
Harran University, Sanliurfa, Turkey

S. M. Land
Pennsylvania State University, University Park, PA, USA
e-mail: sml11@psu.edu

A. Baytak (✉)
Muhendislik Fakültesi, Harran University, Bilgisayar Muh., 63300 Sanliurfa, Turkey
e-mail: ahmet_baytak@yahoo.com

(Dickey 2005; Kafai 2006b; Shaffer 2006). As Squire et al. (2005) stated, new technologies and games are powerful social, cultural, and technological forces that educators cannot ignore. The implementation of computer games into elementary education, however, is a more nascent area of research and practice in education (Warren et al. 2008; Squire 2006). Researchers agree that there is a lack of empirically grounded frameworks for integrating computer games into classrooms (Ke 2008; Shaffer 2006; Squire 2006; Van Eck 2006). As such, educational theory around the use of computer games is still being developed. Two different lines of inquiry have emerged. The first concentrates on the effects of educational games on learning or motivation, including investigations of games simulating authentic practices like art design or urban planning (Shaffer); entertainment games that contextualize academic content (e.g., *SimEarth*, *Civilization*) (Squire); or 3-D virtual worlds that engage students in rich, inquiry investigations in academic content areas (e.g., *Quest Atlantis*) (Barab et al. 2005). A second area centers on teaching children how to design or program educational computer games to learn more about a specific academic topic (Kafai 2006a). Kafai (p. 37) notes that far fewer researchers have emphasized "making games for learning instead of playing games for learning". The remainder of this introduction will concentrate on research efforts of learner-constructed games in educational contexts.

Learning-by-game design

Learning-by-design is neither a new concept nor one that is limited to constructing computer games. The idea of "design" represents a broad class of experiences, but a key experience is that of learning by engaging in design-and-build challenges culminating in the production of an "artifact" that represents underlying understanding (Kafai 2006b; Kolodner 2006). Design projects can include building physical models of artificial lungs (Hmelo et al. 2000), or designing and building a parachute or miniature car and its propulsion system (Kolodner). Learning-by-design is rooted in the constructionist perspective which encourages knowledge-in-use through developing physical or digital objects (Papert 1993b). Learning is presumed to be more meaningful and motivational when it results from students constructing personally-meaningful designs (Bruckman and Resnick 1995). Designing artifacts and sharing them with others make students' ideas concrete and allows them to establish a personal connection with new knowledge (Papert). This aspect is key in constructionism, and goes beyond disciplinary knowledge to emphasize the motivational or emotional aspects of learning (Kafai).

The idea of constructing artifacts through computer programming was initially explored with Papert's LOGO environment (Papert 1993a), and later with Harel's work on the *Instructional Software Design Project* (*ISDP*), where elementary-school children designed math software about fractions to teach younger children (Harel and Papert 1991). Theoretically, this work was based on the premise that more integrated understanding of both programming and fractions could be achieved by helping children program instructional software to teach others about what they were learning. Harel's work was situated in an unusual research partnership between MIT Media Lab and a public school. Within this context, teachers were trained in LOGO, and researchers and teachers collaborated closely in classrooms. Harel's research followed a class of children working daily on their fractions software over a period of 4 months, and presented multiple sources of evidence of developments in children's programming and fractions knowledge, using both pre–post test assessments and controlled comparisons.

Kafai and Ching (2001) extended Harel and Papert's work to software design using science concepts, emphasizing the relationship between science talk and design activities. Kafai found that 4th and 5th grade children could develop both science and programming concepts within a software design context. Kafai (2006b) later expanded the genre of media construction to include video-game design. Her studies brought to light new observations connected to the gaming culture, such as how boys and girls represent characters and plots differently (Kafai 1998), and the efficacy of peer mentoring by more-experienced children.

Since Kafai's work in the early 2000s, game design within schools has remained somewhat unexplored. One could presume this is due, in part, to limited availability of contexts to which the approach could scale, given the time requirement for open-ended software design projects and close teacher partnerships (Rieber et al. 1998). Related research was conducted by Rieber et al., who asked children to design games for younger children, and more recently by Van Eck (2006) who spent weekly sessions with children playing and designing games over two semesters. Both projects suggested that game design could play a role in engaging students' everyday cultures through programming. Van Eck suggested that game play and design could impact girls' attitudes toward technology and science, leading more girls to pursue interests in those areas. In both projects, teams of children conceptualized computer games; the actual games were programmed by University-staffed research teams using multimedia authoring tools (e.g., *HyperStudio* or *Authorware*). Clear ties to academic subjects were also not conclusively specified in these works.

Scholars have acknowledged that software and game design is a complex practice (Prensky 2008), and the constructionist approach requires considerable cognitive demands on the part of the learner (Land and Greene 2000; Hannafin et al. 2009). With support from teachers, materials, and peers, learners presumably advance their knowledge and programming via a "learning-on-demand" model (Kafai and Ching 2001). Some researchers have expressed caution about these complexities, questioning whether the design activity should be part of learning process (Prensky; Tiong and Yong 2008). Shaffer (2006) however, has suggested that maintaining complexity of authentic practices (e.g., the practices of game designers) is critical, as it is necessary to develop distinctive ways of knowing and doing (or "epistemic frames") of that community.

Advancements in graphical or visual programming software that can be learned without programming experience (see for example, Overmars 2004) have enabled even young children to engage authentic game-design practices. Constructionist research on game design has evolved most recently in connection with Mitchell Resnick's Lifelong Kindergarten Group at MIT. The group has developed a media-rich programming environment based on LOGO called *Scratch*, which is a tool designed to make programming accessible to young children through use of dragging command blocks (Peppler and Kafai 2007a). *Scratch* has been implemented as part of The MIT Computer Clubhouse community project, which is a network of informal, after-school programs, typically directed to inner-city youth (Maloney et al. 2008). With *Scratch*, students can design a wide variety of media products, with a low barrier to entry, since it utilizes visual programming.

Current research on *Scratch* has been conducted within the informal context of the Computer Clubhouse, where children use the software in unstructured ways to create video games, animations, greeting cards, or music videos (Maloney et al. 2008; Peppler and Kafai 2007a). Explicit ties to formal school concepts are not part of this program, and the artifacts created by participants reflect personal interests or entertainment. One line of research is largely ethnographic in nature, focusing on the informal media culture and identities of participants in relation to media literacy (Peppler and Kafai 2007b). Other research is based on artifact analyses of student-created games for evidence of developing programming skills (Maloney et al.). Another study collected preliminary usage data from an online repository of user-created *Scratch* products (Monroy-Hernández and Resnick 2008). Taken together, these studies show that participants can develop programming concepts informally over time using *Scratch* and participate in a culture of creative media production; these studies provide few implications, however, for game-design pedagogy in school contexts.

Implications for the present study

Our research builds on the constructionist perspective to software design for learning (Harel and Papert 1991; Kafai 2006b) and describes how 5th grade children in a joint technology and science classroom engage in game-design practices using *Scratch*. Prior research from the late 1990s demonstrated the efficacy of student software design in classrooms, given extensive time, research teams, and mentoring (Harel and Papert 1991; Kafai and Ching 2001). Recent research has examined the next generation of computational tools for enabling constructionist pedagogy using *Scratch*, but within informal, after-school contexts (Peppler and Kafai 2007a). Consequently, questions about the potential of this pedagogy in primary school classrooms remains relatively unexplored, despite continued relevance of the theoretical construct to the learning sciences community (Kafai 2006a, b).

This study extends previous work in that it investigates a 21-day learning-by-game design experience using *Scratch* within a science classroom context. We employed case-study methodology (Yin 2009) to systematically describe the class, concentrating our descriptions on student practices of game design: (a) planning; (b) designing, testing, and debugging of games; and (c) knowledge sharing and social interaction. This framework represents key characteristics of the software design for learning perspective and highlights the processes required for students to develop sharable artifacts (Kafai 2006a, b). The goals of this research were to gain a more detailed understanding of the problems and processes experienced by children to design their gaming artifacts, while operating within an authentic science classroom, and using a graphical programming tool that has yet to be explored in such a context. Peppler and Kafai (2007b) suggest that extending *Scratch* into classroom settings is advantageous to study, as classrooms are potentially more amenable to systematic reflection on design practices, a process that is less commonplace in informal learning spaces. Hence our research goals were exploratory and descriptive, rather than conclusive, with an eye towards extending current constructionist perspectives through description of theoretically relevant practices within the present context.

Method

Participants and context

The research site for this case study was an elementary and middle charter school located in a college town in the Northeast, USA. The case was purposively chosen as it closely represented the pedagogy under study, but it was a class of convenience consisting of 10 5th graders (6 girls and 4 boys). To protect student identities, all names presented are pseudonyms. All students had been previously introduced to basic features of *Scratch* to make animations for prior class project. Three boys, Larry, Kyle, and Eli had prior experience in game design (*Gamemaker*) and comparatively more experience with *Scratch* through after-school, elective programming.

The study was conducted in a joint class of technology education and science. The unit focused on environmental problems and was introduced in the final weeks of the school year. This project was a supplemental science topic that occurred after all required science units had been covered and was not tied to required science standards for the 5th grade. Students were asked to identify a problem to explore, based on a textbook chapter on the environment, and then to design a computer game using *Scratch* to teach that topic to 2nd graders. Two planning sessions took place in the science classroom, and all subsequent game-design programming took place in the school's computer lab. Classes were structured to encourage students to move around and share knowledge and strategies informally. Students collaborated when sharing game ideas, but each student was assigned to a computer to work on his/her design individually. Each student received a Tablet PC laptop computer with a server personal account access.

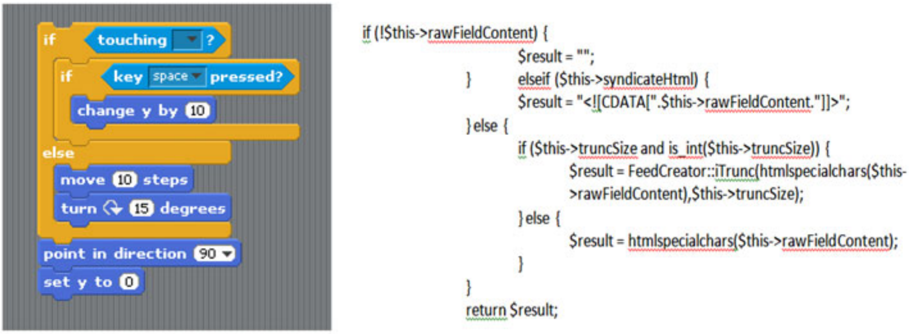
The instructional design for the study was rooted in the overall framework of "constructionism" (Kafai 2006b), incorporating design strategies from the Instructional Software Design Project (ISDP) (Harel and Papert 1991; Kafai and Ching 2001), albeit on a significantly compressed timeframe (21 days for our study, vs. 4–6 months). We asked the class to design video games about environmental problems to teach younger children in their school. Based on the strategies outlined by Kafai (2006b), our design entailed four primary strategies: (a) students make personal choices about the instructional games (artifacts) and representations that they create for their younger learners; (b) students keep a log of their planning and designs; (c) prospective users are invited for feedback sessions; and (d) software and content representations are reviewed and discussed during sessions guided by the teacher.

Three teachers worked with the students: a science teacher, a science co-teacher, and a Technology Education teacher, with the last being the principle investigator of this research. The science teacher and co-teacher were responsible for classroom management and science content expertise, and had no knowledge of *Scratch* or of game design. The technology teacher was primarily responsible for overseeing the study and helping students with the game design software. During this unit, science teachers took on a secondary, supportive role to the technology teacher.

The game design software

The game design software *Scratch* was used by the children to make their computer games. *Scratch* is an open-source programming environment written in the Squeak programming language (http://llk.media.mit.edu/). *Scratch* was selected for this study because it entails a user-friendly interface for children with visual code blocks, reducing syntax errors. It also was free of charge and contained a library of sample designs that could be modified if students needed that level of support to get started. Figure 1 shows sample command blocks and their programming.

There are four main sections on a *Scratch* program: Command blocks, Script area, Stage, and Sprites. Users drag and snap command blocks together to build scripts. There are four different sections in the script area for coding, costumes for drawing sprites, sounds for recording or importing sound files, and coordinates to set the directions of the selected sprite. Users can design their own sprites with the *Scratch* editing tool, use



Programming with *Scratch* **Programming with PHP**

Fig. 1 Similar controlling structures in Scratch programming and php programming (the php code is from Joomla)

Scratch-ready images for the sprites, or export any image to the *Scratch* platform. Students also used *Moodle* course management system for storing their plans and reflections each day, and uploading their games to the server. Uploading games helped the researcher to archive the games over time.

Research design, data sources, and procedures

This research used a multiple-participant case study as the research methodology (Yin 2009). Case study methodology was appropriate for this study, as it allowed us to make first-hand observations in a natural setting (Yin 2009) to address our descriptive research purposes about what happens when children design games about science using *Scratch*. Our analyses were directly tied to our research purposes of informing design and theory of constructionist learning-by-game design, for which this case is an example.

Both qualitative and quantitative data were collected in this study. To enhance the trustworthiness of the study (Merriam 1988; Yin), the research relied on multiple data sources, selected according to those that have been used in similar game-design research (e.g., Kafai 1998, 2006a, b): students' archived games (100 versions over the course of the project), interview transcripts of pre and post interviews, observations and field notes, and journal entries. Observations were based on field notes and video recordings as students worked on design activities. One video camera was set up to capture full-class interactions; another was set up to capture conversations and activities of three girls who were seated together in lab. These girls (Adria, Megan, Isabella) were selected based on: (a) signed releases for videotaping; (b) representativeness of a variety of skill levels; and (c) proximity and rapport with each other. Videos of the small group were transcribed by the first author and examined for interactions related to the three game design categories investigated in this study. After transcribing the videotapes, reports were constructed and discussed with the second author to aid in identifying themes that illuminated our key research foci. A semi-structured pre- and post interview was conducted individually with students about their experience with the game design process. The post-interviews were recorded and conducted outside class time within 2 weeks during free time and during an after-school program. Students' games were occasionally shown during post interviews to prompt questions about design decisions. Finally, students were asked to make daily plans for their designs prior to each design session and after testing sessions. The plans were structured with prompting questions that students responded to online. The general analytical technique was one of triangulating data from multiple, independent evidentiary sources, and pattern matching converging lines of evidence with theoretically derived categories of learning-by-game design (Yin 2009).

This research was conducted over 21 consecutive school days. Data were collected during class sessions as well as after-class sessions for interviews. Each class session was 45 min but 5 min of each session was devoted to attendance, class preparation, and starting computers. The study followed four main phases: (a) planning (3 sessions); (b) design and development of *Scratch* games, including structured peer review sessions (16 sessions); (c) 2nd-grade game testing sessions (2 sessions); and (d) pre- and post-study interviews.

Results

Description of the students' games

All students developed games that integrated environmental science concepts. Most of the students' games represented adaptations of real-world scenarios, with some unrealistic features. Amber for example, selected the ocean as her game world, but some of the game characters were representations such as gray shapes for trash or starfish collecting the trash icons. Isabella used air pollution with a similar concept of a bird collecting gray air smoke icons with a shiny background to represent day, and dark background to represent night. Tanya also used the ocean as her game world where a scuba diver collected oil spills. Megan portrayed the context of a kitchen with several appliances and tools using energy and water, and asked players to turn off switches and water at certain times to conserve energy. Adria created a maze game where each environmental issue was shown with a representative icon. Kyle created a background of New York City and a river to represent his idea that big cities have water pollution that causes bacteria. Larry's game asked players to collect trash on a street view. Nick's game was an ocean view in which the player evaporates oil, and the oil grows similarly to a real world case. Last, Eli made a puzzle game that portrayed coral bleaching. The background was a sea shore, and he used chemical element shapes to show different chemicals that could be used to undo the bleaching of the coral.

The Results section is organized around the following: (a) a single case description; (b) description of the overall student game design process; (c) analyses of programming concepts incorporated.

A single case description of the game design process

This section provides an overview of a single student's game-design process (Adria), to illustrate how the process unfolded holistically for one student. Adria's case was prototypical in terms of the overall design process and end-product game quality, but was unique in that her confidence and skill level were below average at the beginning of the study. Adria started the game-design project with little prior experience in gaming and computing. She had explored *Scratch* through technology class, but said she was never able to design something that functioned properly. She reported that she expected to have difficulty with the game-design project because of her low confidence and experience with using *Scratch*.

During the first few design sessions, Adria started and stopped a new game each day. On the first day, she noted in her online planning journal that "I'm doing rainforest destruction. My idea for the game is to have them walk around the forest putting tree trunks on the stump." However, once design sessions began, Adria stopped work on this idea. When queried about this by the researcher, she said she planned to design a game about rainforest destruction but was not sure what to do. She replied that she did not have sufficient skill with *Scratch* to make a game about rainforest destruction, even though she was able to use the image editor to add and edit game sprites, which was the first step prior to programming the game. After assistance from the teacher, she agreed to draw some main sprites of the game she planned to design. However, her planning journal for the second day said that, "I changed my mind now, you have to walk around a beach picking up stuff, and you have to put it in a recycling bin or a trash can." When asked why she switched to a new game, she mentioned that making a game about rainforest destruction would be too hard to design, so she decided to a make a different one. She spent one design session drawing sprites for her new game.

Adria started a totally new game idea again during the third design session. For this game, she decided to make a maze game (Fig. 2) where a knight tried to stop a dragon from heating the earth. During a conversation with science teachers, she mentioned that the dragon represented global warming which increases the Earth's temperature, and the knight represented humans trying to stop global warming. After peer and teacher feedback, Adria added more science details and associated gaming functions to her game. For example, she added a car to the maze that the knight would touch to solve air pollution problems resulting from car exhaust. After she settled upon this game idea, every design session became more productive for her. She routinely asked the teachers and peers for help in adding new features and commands, and no longer expressed her weakness in using the software, as she did frequently before.

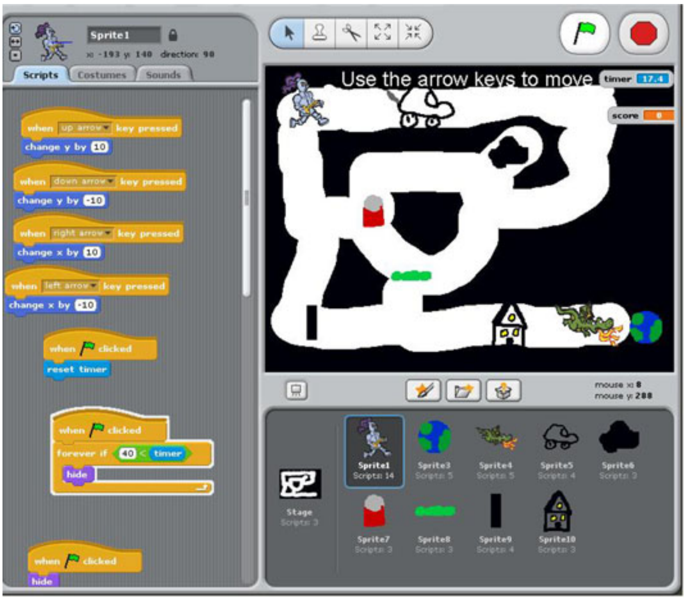


Fig. 2 A screenshot from Adria's final game design platform

Adria incorporated several *Scratch* commands that she learned from other classmates. She frequently asked Megan and Isabella who were sitting next to her to try her game. After each try, her classmates provided constructive feedback for her to improve the game. Later, during a testing session with 2nd graders, she found that they wanted more 'fun' in the game, so she added background sounds and actions to make the game more enjoyable. These changes involved significant increases in the number and type of programming concepts she used in her game.

Overall, the main obstacle for Adria during the project was conceptualizing and operationalizing the initial idea as a game. She gained traction only after deciding first on the type of game genre that she was comfortable making. Once this idea stabilized, she continued increasing her skill in design and programming.

Results of game design processes

This section presents findings using the class as the unit of analysis, and describes the game design process of the participants. These findings reflect observed patterns and themes, using the theoretical framework of learning-by-game-design as the analytical lens.

Planning

The goal of the planning phase was to provide three sessions for participants to research information on the environmental content of the game and to operationalize these concepts into a plan for designing their educational game. Students used the class textbook and the internet to research their environmental science topic. Hence, students served dual roles of both learner and instructional/game designer. Participants were asked to post their findings and design plans on the course management system. Analysis of student plans revealed that the students began the process by identifying a connection between causes and effects of the environmental issue they were exploring. This helped them to identify ideas for icons, game worlds, and goals for the players. Most students initially focused on broad causes (e.g., pollution), rather than individuals' responsible actions. Several class sessions and teacher-directed prompts for planning journals were required to help students begin to integrate their environmental content focus with the features of their game. The planning process in some cases was constrained by the level of technical knowledge. As presented earlier in Adria's case, topics changed multiple times according to how readily she believed she could program it.

Prior research has allotted substantially more time to planning phases (Harel and Papert 1991; Kafai and Ching 2001). Our data suggest that students were able to jump into programming after only 3–4 days of planning. But, it was not without starts and stops for some. Our findings points to a possible trade-off to compressing planning times—students may be able to engage game programming sooner, but may be limited by what they feel capable of programming at that point in time, and what they understand about the topic under study, rather than based on a well-conceived design.

Design, testing, and debugging

Once students' solidified their overall idea, they searched for new information on their topic as they added new features to their games. For example, it was observed in an early version of Kyle's game that he included images of moving shapes without any clear purpose. When asked by the science teacher what those represented, he said they were bacteria that came from polluted water which were bad for people. The teacher then asked him if all bacteria are bad. Kyle paused and replied that not all. He then decided to search for more information on bacteria. It was later observed that Kyle had two types of bacteria shapes with different purposes. One type of bacteria would not affect the players' game life whereas the other type of bacteria took game lives of the player when it was touched. Similarly, Nick had oil spills in his game, but there were no clues for the 2nd graders to understand how the oil spilled into the water. When questioned by the researcher about this, Nick conducted more research online and found that sometimes cargo ships or factories spill oil into the water. Finding out this information helped him to conceptualize components of the oil spill and how it could be portrayed visually and realistically in the game. These examples provide some insights into how and when the design process prompted a reflection-in-action cycle (Schön 1983) that extended from efforts to represent science with the gaming features. The teacher played an important role in the process by prompting students to think about new ideas or justify their ideas represented in their games. These conversations were essential for sparking reflection, which led to new actions that showed a more comprehensive representation of the problem under study. Kafai and Ching (2001) reported that such guiding discussions could also be facilitated by near-age peers with prior design/programming experience, suggesting that peers can serve a similar guiding role when grouped by experience.

Not all reflection was due to prompting by the instructors. Testing or debugging games was another common practice. Students tested their games several times not only for debugging but for trying out and enjoying their own games. Changes and improvements to the games were frequently paired with these testings, seemingly in response to a "need to know" that emerged from trying out and wanting to improve their games. For example, after testing her game, Tanya asked how she could add a new feature to her game where other sea creatures were affected after touching the oil spills. This question prompted a discussion and led her to search for what happens to sea creatures in real life when there is oil in water.

Students were also asked to engage in structured peer-testing sessions, where they tried out each other's games. This sharing process may have led to the spread of strategies and ideas throughout the group. Sometimes, new features appeared in the students' designs that were attributable to ideas gained from peers. For example, Tanya reported the following features that she learned from reviewing others' games:

> *Researcher*: "*After trying these games, did you learn something from them to improve your game?*"
> *Tanya*: "*Well, with the letters, I learned that from Lacy's game that I want to put that in, too. Timing, I got from Isabella's game.*"

Formative testing also occurred when the target audience (2nd graders) was given a chance to try out the games. Each game was tried out by at least three second-graders. After testing, the second graders filled out an evaluation form. Despite the fact that the 2nd graders did not directly interact with the 5th-graders during testing (due to scheduling logistics), several 5th-graders made changes to their games as a result of this testing. For example, Kyle noted after the 2nd graders feedback: "*I think I should make them learn more about bacteria, and I should also have better directions.*" He then added information for each game character, such as "*I am bacteria 2. I am really dangerous try to avoid me*" and "*Hello! I am medicine if you get on me I will heal you*". Similarly, after the 2nd grade testing, Eli added labels to the coral shapes to provide clearer navigation, and Lacy revised the text of her game directions to improve readability. It was also noted that many students extended their timers based on the second graders' comments. Adria and Eli used background sounds to better appeal to the 2nd graders.

Sharing and peer interactions

This section describes how participants interacted with each other during the game development process in structured and unstructured ways, how they shared ideas and strategies, and how these interactions affected their designs. Findings are based on interviews, video observations, and observations of game artifacts.

Unstructured interactions

During the individual game design tasks, there were a number of unplanned collaborative interactions. These interactions were observed in video recordings where students frequently asked questions to one another and the teachers. To illustrate, the following conversation occurred in response to Amber asking Adria for help:

> Adria: "*Do you want the person to disappear*?"
> Amber: "*Yeah.*"
> Adria: "*I cannot do that but I can do*, *when the starfish touches one* (referring to the pollution) *it becomes blue*" (The original color was gray and the background of her game was blue, in other words it disappears.)

The teachers were also able to encourage informal sharing of strategies by referring some students' questions to others who were knowledgeable in that area. For instance, Isabella asked the technology teacher about adding a timer to her game. After he looked at the *Scratch* settings and could not determine a solution, he asked the rest of the class if anyone could help her. Larry then showed the teacher and Isabella how to set a timer in her design. Similarly, Adria asked the teacher how to add the ability to move different directions using arrow keys. The teacher then asked Isabella to help Adria, since he had just taught her that same feature. Isabella successfully helped Adria add the move function to her game.

Interestingly, students also learned optimization strategies from each other. During one of the earlier design sessions, Amber asked the teacher for help with adding a point calculation feature to her game. The teacher suggested using a long Boolean expression. Her final game, however, showed that she replaced this Boolean expression with a short script. She was asked about this change in the interview:

> Researcher: "*Did you take this* [*Boolean*] *command out*?"
> Amber: "*Well*, *I took this out because a friend told me that you do not have to make it so long. You can just have* [*this*] *other* [*script*]."
> Researcher: "*Who said so*?"
> Amber: "*Megan.*"
> Researcher: "*But*, *does it still do the same function*?"
> Amber: "*Yeah.*"

Similar to learning patterns seen in informal learning and gaming cultures (Sefton-Green 2003), these interactions illustrate how game design affords opportunities for self-directed learning or *upskilling* through observation, imitation, and peer teaching at the point of demand.

Structured interactions

Participants engaged in structured opportunities to share their designs. One session took place after the 5th design session, after most students already started a game. The students were divided into two groups, and each student took turns presenting their environmental issue and game idea. Video observations showed that discussing students' games often created opportunities for talking about science and how to represent science concepts in their games. The following excerpt shows how subtle prompting to connect science and game features helped students to refine ideas:

> Adria: "*Well my game is about global warming, the way that I show global warming in a fancy way so that it will be more fun for kids, because you are a knight with armor trying to go kill the dragon which is heating fire on earth to try to like make it blow up. And if you do not get there in time the earth blows up.*"
> Science co-teacher: "*So you are using a character appealing to the kids?*" …
> Adria: "*Yeah.*"
> Science co-teacher: "*…but we are not learning about the dragon, right? We are learning about what?*"
> Adria: "*Global warming.*"
> Science co-teacher: "*How are you teaching them?*"
> Adria: "*I am teaching them because, I do not know if they already know this, but when you use electricity, it heats it. I can make the dragon something.*"
> Larry: "*The dragon could be named pollution.*"
> Science co-teacher: "*That is a good idea.*"
> Larry: "*Or pollutionator.*"
> Adria: "*Yeah.*"
> Larry: "*His name [the knight] should be recycler.*"

Although their talk is simplified (the conversation above includes some statements about recycling that are somewhat tangential to global warming), what is valuable here is the attempt by students to make sense of science in a context that is very real and important to them. While one could argue the place of the "pollutionator" and "recycler" in this scenario, they were important to Larry's understanding (his game was about trash and recycling) and to moving Adria in a direction of more integration of her game and science problem. However, it also points to a need for more structured opportunities to external science thinking, as it can be challenging for teachers in spontaneous interactions to identify all opportunities to deepen thinking and game features.

What was learned?

Since this project was a supplemental, enrichment opportunity, formal assessments were not established. Teachers used an evaluation rubric at the middle and the end of the project to provide feedback about environmental content and richness of its integration (mean score 3.7 out of 4). Scores reflected what teachers saw represented in the games, as well as conversations with students. Teachers provided a written report that described how each student addressed the environmental problem in the games and listed if a student misrepresented an environmental concept. However, this process was challenging for teachers, and was amplified by the novelty of the technique and technology. The games designed incorporated both educational and entertainment value, so it was common to see elements of the imaginative and fantasy in games, rather than standard scientific representations.

In addition, designing for students 3 years younger seemed to encourage students to treat concepts more simply than perhaps they were capable. Given these challenges, we make limited claims as to the nature of the science knowledge represented within the students' games. However, it is important in future work to more formally address such questions.

Others have identified similar challenges to assessing what is learned in constructionist environments (Ioannidou et al. 2003; Tangdhanakanond et al. 2006). Since students' learning trajectories differ based on their individual designs, it is not always possible to anticipate the same outcomes from all students. Each student's learning is highly differentiated, and assessments of a core set of items might not accurately reveal what is learned from all students (Squire 2006). Kafai and Ching (2001) have noted that given that students are not always at the same place at same time, assigning "sweeping statements about trends or 'progressions' turns out to be more complicated given the complex nature of the software design project work (pp. 358–359)". Portfolio-based assessment (Tangdhanakanond et al.) relies upon multiple layers of evidence such as written explanations and justifications as a way of clarifying understandings embedded within a created artifact. Other studies have used concept mapping as a method to visualize and quantify students' deepening knowledge after participating in design tasks (Shaffer 2006). Artifact assessments should ideally include both the artifacts themselves and ways of explaining conceptual relationships that have been constructed.

A limiting factor in assessing for understanding involves time. Constructionist approaches are complex learning environments that occur in contexts with finite time and resources (Carver 2006). When extensive time and multiple assessments have been used, significant gains in understanding have been reported in constructionist research. Harel and Papert's (1991) early work with the ISDP project utilized a multi-tiered approach to evaluation that relied on (a) case studies for illustrating and comparing individual approaches and processes within the classroom culture; (b) pre–post test assessments to identify gains in programming and fractions knowledge (typically through interviews); and (c) cross-class comparisons to measure learning gains on knowledge tests using control groups. In shorter timeframes, however, one might not expect to see differences on knowledge tests, either between classes or on pre–post assessments. Assessment in learning sciences contexts requires multiple opportunities for students to demonstrate competence and should be aligned with goals for deep understanding (Carver).

We also analyzed the various types of programming concepts that were accessible to students through game design with *Scratch*. To analyze programming concepts used, the students' games were analyzed based on Maloney et al. (2008) and Malan and Leitner (2007) who classified each *Scratch* command as different programming concepts. These studies categorized command blocks of *Scratch* and counted how many times the students used these commands in their designs. Similarly, we classified the *Scratch* programming concepts based on Malan and Leitner's scheme as the following: *statements*, *Boolean expressions*, *conditions*, *loops*, *variables*, *threads*, and *events*. Table 1 gives a detailed counting of the programming concepts that students used in their final games (Table 2).

As Maloney et al. (2008) indicated, students could use fewer commands to have the same function in their games. Hence, a large number of commands does not necessarily indicate more complex programming. Instead, the use of complex commands such as *variables* and *events* show that they were able to use advanced programming concepts in *Scratch*. It was interesting that boys rarely used the *broadcasting* feature (mean = 0.04) which is an advanced feature of *Scratch* (Broadcasting is categorized under *Events* by Malan and Leitner). Yet, most of the girls, after getting help from the teacher, included

Table 1 A summary of characteristics of the students' games

Student name[a]	Game genre	Graphics and character development	Control options	Duration of the game
Kyle	Platform	A fantasy game character, no gender specified	Arrow keys and mouse clicks	1 level with lives
Larry	Platform	Stick man, gender is not clear	Arrow keys and shortcut keys	3 levels
Amber	Platform, action	Star fish, no gender specified	Arrow keys	1 level
Isabella	Platform, action	Bird, no gender specified	Mouse move	2 levels
Lacy	Platform, action	A girl, female character	Arrow keys and some keys	1 level
Nick	Action, action	Boat, no gender specified	Mouse move and some keys	1 level with lives
Tanya	Platform, action	Scuba diver, no gender specified	Arrow keys and some keys	3 levels
Adria	Platform, maze, action	Knight, no gender specified	Arrow keys	1 level in a certain time
Megan	Educational, action	X	Mouse clicks	1 level
Eli	Educational	X	Some keys	1 level

[a] All names presented are pseudonyms

Table 2 Students' average use of different *Scratch* programming concepts

Students	Sprites	Statements	Boolean	Conditions	Loops	Variables	Treads	Events
Kyle[a]	10.2	50.5	11.7	7.1	13.8	0.6	18	0.1
Larry[a]	14.9	57.7	13.2	13.6	2.1	1.4	34.3	0
Nick	7.7	26.4	7.8	9.4	12.1	4.6	17.5	0
Eli[a]	17	20	7	4.8	3	4.6	18.2	0
Tanya	25.8	127	16.3	16.3	19.9	1.6	41.8	3.4
Amber	18.8	31.8	26.1	15.8	16.9	3.5	30.5	0
Isabella	14	42	13.9	11.8	12	1.2	15.4	10.6
Lacy	14.8	33.1	2.5	0.2	0.5	0.2	19.5	0.3
Adria	10.3	56.6	19.9	19.4	19.4	4.5	49	0
Megan[a]	11.2	41.8	10.1	5.3	7.2	1.9	20.4	13.5
Class average	23.56	78.4	20.6	16.8	17	4.13	42.9	3.88

[a] Denotes prior game design or Scratch experience

many broadcasting commands in their games (mean 7.72). With this broadcasting function, the students were able to make new game levels.

The data are consistent with prior work (Robertson and Howells 2008) that visual programming toolkits such as *Scratch* can help novice programmers easily combine commands. With these combinations, the students made scripts that ran actions as part of their games. The students of this project were successfully able to program their scripts with little mentor assistance. Despite students' minimal prior experience with *Scratch* and almost no design experience, the students were able to have a functional game after the 10th design session, and after the 6th session for the experienced students.

The data show that students, regardless of gender, were able to use different advanced level programming concepts at elementary grade level. None of the students had any difficulty with adding *statements* to their programming scripts. With the simplified *Scratch* command block, the students did not have difficulty with collisions, but they could not set other Boolean expressions without the teacher's help. The students' use of *conditions* and *loops* were limited to 'forever' or 'if' command blocks. Students rarely used 'if else' or other complex command blocks in their games.

Discussion

This study used a constructionist framework (Papert 1993b) to support students' understanding by designing computer games. By designing a game, students represented their thinking as an external and computational artifact and examined what they needed to know to further improve their designs (Chapman 2009). This study provides insights into the conditions under which the students' "need to know" emerged naturally as a result of engaging in the game-design process. This was most clearly evident in cases where students' programming knowledge advanced as a result of a new goal or intention. However, this cycle was also apparent in instances where students realized a need to know more precise science knowledge to increase the fidelity of their games. Future research should explore more directly such cycles of science learning that are potentially triggered during the game-design process.

Our findings describe how computer game design became a dynamic learning process where students' self-motivations about game play prompted goals, and these goals prompted additional information seeking and new interactions. Interacting with the design tool as an "object to think with" (Papert 1993a, b), learners conceptualized their ideas, and reformulated and implemented them into the design. The process of designing and testing led to continual redesign. Peer and teacher interaction played a role in the redesign process, as they not only shared games, but ideas, concepts, and strategies (Gargarian 1996). Zhang et al. (2007) noted that the spread of ideas across a community is characteristic of a productive knowledge-building community. The students did not appear to see anyone as the only expert, creating a social climate that encouraged a sense of agency (Barron et al. 1998). This was facilitated by the teachers, who directed students towards multiple resources, tools, or peers as a means to solve design problems.

The success of these children in programming their own games lends support to the conclusion that elementary students can access complex programming if the software they are using and visual interface is developed for their level of experience (Lin et al. 2005). Our study shows that all participants, without exception, finished a game in less than 3 weeks that was meaningful to them and the community around them. These results may have implications to influence the cycle of elective technology education courses and computer clubs that attract and sustain students who are already strongly interested in computer science (Repenning and Ioannidou 2008). Computer clubs or courses utilizing game design could serve to attract students who might not traditionally pursue computer-related hobbies or interests.

However, our study has several limitations, the most obvious one being related to our assessment design. We could readily identify advances in programming skills through quantitative counts and through examination of functional games. But, clarifying more clearly how or if learners deepened their science knowledge is largely descriptive and anecdotal. Additionally, video recordings of students working on their games were limited to 3 girls, and the nature of the set up within a large, busy class did not allow us to capture subtle conversational elements across more students. We also acknowledge an overarching limitation of the case study methodology we used, which is that our context and case selection were unique; thus, this potentially limits the value of our findings beyond the circumstances of this case. Specifically, students were provided with laptops, and three boys were experienced in technology and/or game design. These factors may have affected the presence of student helping behaviors and success in creating games. Finally, our study took place in a learning environment that was not tied to science standards or graded performance and is thus limited to inform classroom practices tied to such indicators. Future research might address these limitations.

A promising outcome from our study points to students' advancements in programming ability in a short time frame and through situated activity. Future work should investigate whether programming knowledge gained through game design extends beyond the production of the game artifact itself. Prior research has shown that students who took object-oriented programming courses using visual interfaces performed better at C++ and Java programming courses (Cooper et al. 2000). Researchers who have investigated use of *Scratch* have also suggested that it might be a primary step into learning more advanced programming languages (Malan and Leitner 2007; Maloney et al. 2008). These claims warrant further investigation. It is also worth investigating whether game design using visual programming tools can play a role in decreasing technophobia, particularly among girls (Van Eck 2006). In this study, a few girls began with a fear of programming with *Scratch*, but it was no longer apparent by the time they finished their games. Gender implications for interests toward math, technology, and science (Van Eck) can be more precisely explored in future studies.

In this study, measuring learning gains in disciplinary knowledge was difficult to discern, given the individualistic nature of the game design process. It was clear to us that making such assessments would require narrowing of the scope, with more full-class investigations of a more constrained topic. Barron et al. (1998) have suggested using thematically related cycles of problem solving and investigation that precede the introduction of design projects, as a way to scaffold students' conceptual understanding first. Similarly, Kolodner (2006) has reported significant knowledge gains with a learning-by-design (LBD) pedagogy that involves iterative cycles of investigation by the full class, with small group design work. The LBD pedagogy uses a system of activity structures designed to prepare students for a complex design challenge. Such approaches, however, depend on a narrower scope (e.g., students are constrained about what they are asked to design). Narrowing the scope of both content and design focus will lead to more focused assessments, and encourage a more systematic interplay between design activities and research/inquiry activities (Barron et al.; Kolodner).

This points to a tradeoff that must be considered in constructionist work—content coverage versus authenticity of the design process (Kafai and Ching 2001; Shaffer 2006). One emphasizes knowledge integration and the other personal investment and complexity. One implication may be to limit open-ended game design projects to informal contexts, like after-school clubs or to technology classes that focus solely on game design, without curricular integration. On the other hand, given gender preferences for programming (Kafai 2006a), it is difficult to ignore the engagement of the girls in our study. It is possible that these girls would not have elected to join a computer club to learn about *Scratch* or other game design software, without first being exposed to a success that came from exploring the activity in a non-elective classroom environment.

References

Barab, S. A., Thomas, M., Dodge, T., Carteaux, R., & Tuzun, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development, 5*(1), 86–108.

Barron, B. J. S., Schwartz, D. L., Vye, N. J., Moore, A., Petrosino, A., Zech, L., et al. (1998). Doing with understanding: Lessons from research on problem- and project-based learning. *The Journal of the Learning Sciences, 7*, 271–311.

Bruckman, A., & Resnick, M. (1995). The MediaMOO Project: Constructionism and professional community. *Convergence, 1*(1), 94–109.

Carver, S. (2006). Assessing for deep understanding. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 205–221). Cambridge, MA: Cambridge University Press.

Chapman, R. (2009). Encouraging peer sharing: Learning reflections in a community of designers. In Y. B. Kafai, K. A. Peppler, & R. N. Chapman (Eds.), *The computer clubhouse: Constructionism and creativity in youth communities* (pp. 81–90). New York, NY: Teachers College Press.

Cooper, S., Dann, W., & Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. In *Proceedings of 5th Annual CCSC Northeastern Conference* (pp. 28–29). Ramapo, NJ.

Dickey, M. (2005). Engaging by design: How engagement strategies in popular computers and video games can inform instructional design. *Educational Technology Research and Design, 53*(2), 67–83.

Gargarian, G. (1996). The art of design. In Y. B. Kafai & M. Resnick (Eds.), *Constructionism in practice: Designing thinking, and learning in a digital world* (pp. 125–160). Lawrence Erlbaum: Mahwah, NJ.

Hannafin, M. J., Hannafin, K. M., & Gabbitas, B. (2009). Re-examining cognition during student-centered, web-based learning. *Educational Technology Research and Development, 57*(4), 767–785.

Harel, I., & Papert, S. (1991). Software design as learning environment. In I. Harel & S. Papert (Eds.), *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school* (pp. 41–85). Norwood, NJ: Ablex.

Hmelo, C. E., Holton, D. L., & Kolodner, J. L. (2000). Designing to learn about complex systems. *The Journal of the Learning Sciences, 9*(3), 247–298.

Ioannidou, A., Rader, C., Repenning, A., Lewis, C., & Cherry, G. (2003). Making constructionism work in the classroom. *International Journal of Computers for Mathematical Learning, 8*(1), 63–108.

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

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

Kafai, Y. B. (2006b). Case-based reasoning. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 225–242). Cambridge, MA: Cambridge University Press.

Kafai, Y. B., & Ching, C. C. (2001). Affordances of collaborative software design planning for elementary students' science talk. *The Journal of the Learning Sciences, 10*(3), 323–363.

Ke, F. (2008). A case study of computer gaming for math: Engaged learning from gameplay? *Computers and Education, 51*, 1609–1620.

Kolodner, J. L. (2006). Case-based reasoning. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 225–242). Cambridge, MA: Cambridge University Press.

Land, S. M., & Greene, B. A. (2000). Project-based learning with the World Wide Web: A qualitative study of resource integration. *Educational Technology Research and Development*, *48*(1), 45–66.

Lin, J., Yen, L., Yang, M., & Chen., L. (2005). Teaching computer programming in elementary schools: A pilot study. Presented at NECC, Taiwan. Retrieved March 25, 2009, from http://www.stagecast.com/pdf/research/Lin_NECC2005_Paper_RP.pdf.

Malan D. J., & Leitner, H. H. (2007) *Scratch* for budding computer scientists. *SIGCSE 2007*, Covington, KY.

Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. *SIGCSE 2008*, Portland, OR, USA. Retrieved March 20, 2009 from http://info.scratch.mit.edu/Research.

Merriam, S. B. (1988). *Case study research in education: A qualitative approach*. San Francisco, CA: Jossey-Bass Publishers.

Monroy-Hernández, A., & Resnick, M. (2008). Empowering kids to create and share programmable media. *Interactions, 15*(2), 50–53.

Overmars, M. H. (2004). Learning object-oriented design by creating games. *IEEE Potentials, 23*(5), 11–13.

Papert, S. (1993a). *Mindstorms: Children, computers, and powerful ideas* (2nd ed.). New York, NY: Basic Books.

Papert, S. (1993b). *The children's machine: Rethinking schools in the age of the computer*. New York, NY: Basic Books.

Peppler, K. A., & Kafai, Y. B. (2007a). *What video game making can teach us about learning and literacy: Alternative pathways into participatory cultures*. Tokyo, Japan: The Digital International Game Research Association Meeting.

Peppler, K. A., & Kafai, Y. B. (2007b). From SuperGoo to Scratch: Exploring creative digital media production in informal learning. *Learning, Media and Technology, 32*(2), 149–166.

Prensky, M. (2008). Students as designers and creators of educational computer games: Who else? *British Journal of Educational Technology, 39*(6), 1004–1019.

Repenning, A., & Ioannidou, A. (2008). Broadening participation through scalable game design. *SIGCSE Bulletin, 1*(40), 305–309.

Rieber, L.P., Luke, N., & Smith, J. (1998). Project KID DESIGNER: Constructivism at work through play. *Meridian*: *Middle School Computer Technology Journal, 1*(1). Retrieved May 12, 2009, from http://www.ncsu.edu/meridian/jan98/index.htm.

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

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York, NY: Basic Books.

Sefton-Green, J. (2003). Informal learning: Substance or style? *Teaching Education, 13*(1), 37–52.

Shaffer, D. W. (2006). Epistemic frames for epistemic games. *Computers and Education, 46*(3), 223–234.

Squire, K. (2006). Changing the game: What happens when video games enter the classroom? *Innovate*: *Journal of Online Education*, *2*(3). Retrieved August 15, 2010, from http://www.innovateonline.info/pdf/vol1_issue6/Changing_the_Game_What_Happens_When_Video_Games_Enter_the_Classroom_pdf.

Squire, K., Giovanetto, L., Devane, B., & Durga, S. (2005). From users to designers: Building a self-organizing game-based learning environment. *TechTrends, 49*(5), 34–74.

Tangdhanakanond, K., Pitiyanuwat, S., & Archwamety, T. (2006). A development of portfolio for learning assessment of students taught by full-scale constructionism approach at Darunsikkhalai School. *Education, 13*(2), 24–36.

Tiong, K., & Yong, S. (2008). Learning through computer game design: Possible success (or failure) factors. *The 16th International Conference on Computers in Education* (pp. 947–951). Retrieved May 12, 2009, from http://www.apsce.net/ICCE2008/papers/ICCE2008-paper293.pdf.

Van Eck, R. (2006). Using games to promote girls' positive attitudes toward technology. *Innovate*: *Journal of Online Education, 2*(3). Retrieved August 11, 2010, from http://www.innovateonline.info/pdf/vol2_issue3/Using_Games_to_Promote_Girls'_Positive_Attitudes_Toward_Technology.pdf.

Warren, S. J., Dondlinger, M. J., & Barab, S. A. (2008). A MUVE towards PBL writing: Effects of a digital learning environment designed to improve elementary student writing. *Journal of Research on Technology in Education, 41*(1), 113–140.

Yin, R. K. (2009). *Case study research, design and methods* (4th ed.). Newbury Park, CA: Sage Publications.

Zhang, J., Scardamalia, M., Lamon, M., Messina, R., & Reeve, R. (2007). Socio-cognitive dynamics of knowledge building in the work of 9- and 10-year-olds. *Educational Technology Research and Development, 55*, 117–145.

Ahmet Baytak is a professor of education and computer at Harran University.

Susan M. Land is a professor of education at the Pennsylvania State University.