Computers & Education 73 (2014) 26–39

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing

Fengfeng Ke*

Educational Psychology and Learning Systems, College of Education, Florida State University, Tallahassee, FL 32306-4453, USA

ARTICLE INFO

Article history:
Received 1 August 2013
Received in revised form
20 December 2013
Accepted 23 December 2013

Keywords:
Learning by design
Game-based learning
Mathematical disposition
Thinking mathematically
Computer game making

ABSTRACT

This mixed-method case study examined the potential of computer-assisted, math game making activities in facilitating design-based math learning for school children. Sixty-four middle school children participated in Scratch-based, math game making activities. Data were collected via activity and conversation observation, artifact analysis, interviewing, and survey. The study findings indicated that participants developed significantly more positive dispositions toward mathematics after computer game making. The study also found that experience-driven game design processes helped to activate children's reflection on everyday mathematical experiences. Mathematical thinking and content experience were intertwined within the process of computer game authoring. On the other hand, children designers were involved in game-world and story crafting more than mathematical representation. And it was still challenging for them to perform computer game coding with abstract reasoning.

1. Introduction

Constructionists argue that learners actively construct knowledge out of their experiences, especially when they are engaged in building objects (Kafai, 1995; Papert, 1980). That is, learning occurs when the learners' active exploration (i.e., artifact design and creation) makes them develop a knowledge representation of their experience or discover an inconsistency between their current knowledge representation and their experience. Attributed to social constructivism, learning usually occurs within a social context in which interactions between learners and peers will activate collaborative exploration, articulation, reflection, and hence assimilation or accommodation for improved knowledge representation (Piaget, 1957; Vygotsky, 1978). Thus collaborative design can be considered as a *mindful* approach to learning or a meaningful environment of knowledge construction (Langer, 1997). The previous projects on learning by design also provided empirical evidence for design-based math and science learning (Kafai, 1995, 2006; Kolodner et al., 2003).

However, questions remain as to how design thinking and content-specific reasoning or knowledge development interact with each other during collaborative design, especially for novice designers. Therefore, this case study examined an implementation of design-based learning through educational computer games making by middle school children. In particular, it explored the presence and contexts of math learning of middle school children during the processes of collaborative design and computer-assisted game making.

2. Literature review

2.1. Learning by design or making

According to the theories of problem-based learning, case-based reasoning, and situated learning (Brown, Collins, & Duguid, 1989; Kolodner, 1993; Shank, 1999), design creates contextualized and authentic learning in that design tasks can make students resort to real-life skills and domain knowledge in doing project-like work, thus making knowledge and skills acquired in such situations more

---

* Tel.: +1 850 644 8794; fax: +1 850 644 8776.
E-mail address: fke@fsu.edu.

0360-1315/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compedu.2013.12.010

transferable to future situations (de Vries, 2006). The prior research on learning by design and making suggest that design, as a formal and problem-solving process, should get learners to carry out inquiries or investigations, perform problem representation and solving by applying knowledge and skills, reflect on design and problem solving experiences, and engage in self-explanation and communication activities. It is argued that design will get learners in active and collaborative learning.

Earlier works on design-based learning examined the potential of using artifact designing and making as either a learning inquiry that contextualizes curricular activities (e.g., Fortus, Dershimer, Krajcik, Marx, & Mamlok-Naaman, 2004; Kolodner et al., 2003), or a meaningful and engaging context that allows the designers to implement and ground content knowledge and critical thinking skills such as representation, reasoning, and creative thinking (e.g., Kafai, 1995, 2006; Li, 2012; de Vries, 2006). Kafai (1995), taking the perspective of constructionism, conducted a longitudinal study on a class of fourth-grade students who used *Logo* to program computer games to teach fractions. The study indicated that students got to construct mathematical representations during the design process, and game design can act as a tool for knowledge reformulation and personal expression. Kolodner et al. (2003) developed and examined a middle-school science program called *Learning by Design*™ (LBD) that integrated the theoretical propositions of case-based reasoning into the educational practice of problem-based learning. A structured sequence of small-group and whole-class investigation, iterative design, construction, and presentation activities was orchestrated and ritualized to reinforce design-based scientific discovery learning. The study suggested that LBD students consistently learned earth and physical science content as well or better than comparison students, and performed significantly better in scientific experiment design, analysis, collaboration, and metacognitive skills. Fortus, Dershimet, Krajcik, Marx, and Mamlok-Naaman (2004) examined the effect of design-based science pedagogy on the development of scientific knowledge and real-world problem solving skills of high-school students. Like Kolodner et al. (2003), they made the design inquiries and design-based problem-solving as the heart of the science curricular activities. Their study findings indicated that knowledge-test-based learning occurred for both high and low achievers after their participation in the design-based learning program. Recently, Li (2010, 2012) examined design-based learning using 'enactivism' – a theoretical framework that argues for a close connection between the affordance of a learning environment and a learner's capacity of action and perception, and a participatory culture in knowledge development. She proposed that gaming and game design should afford embodied and contextualized enactment of knowledge authoring. In an exploratory case study, Li (2012) reported that teachers' game design and building experience enabled them to re-conceptualize pedagogy and teaching practice.

On the other hand, the prior research underlined a variety of issues to be tackled when constructing and implementing a design-based learning program. In particular, it is challenging to connect design activities to the targeted domain content. That is, learners can be involved in visual or technical design and development rather than content exploration or the development of formal knowledge (Kafai, 1995, 2006; Kolodner et al., 2003). Besides, research by de Vries (2006) and Kolodner et al. (2003) highlighted the challenge of executing practices of design in a proficient way in the school classroom setting. Integrating studio-like, iterative design processes into fixed school sessions and the traditional classroom environment is in need of careful consideration and examination. As such, it is warranted to conduct a study examining how design and artifact-making processes invite and sustain content-specific reasoning and knowledge construction, representation, or reconceptualization in the school setting.

2.2. Learning by computer game making

Computer game making has been considered and examined as a "powerful learning environment" to stimulate active, autonomous learning via rich contexts and authentic tasks of composition and construction (Robertson & Howells, 2008; Smeets, 2005). The practice of having students as designers of computer games can actively engage learners and offer them opportunities to exercise the skills of digital storytelling, computational thinking, and creative thinking (Habgood, Ainsworth, & Benford, 2005; Kafai, 1995, 2006; Kelleher, Pausch, & Kiesler, 2007; Leng, Ali, & Baki, 2010; Robertson & Good, 2005; Vos, van der Meijden, & Denessen, 2011). Educational game making that requires content application can be applied as a "microworld" in which designers or learners get to explore, represent, and test their domain knowledge and skills and integrate them into the game designed (Mitchell, Kelleher, & Saundry, 2007; Roblyer & Edwards, 2000; Shaffer, 2005).

Based on the prior research and theoretical discussions on design-based learning in general and computer game making in particular, this study speculates that the following three meaningful interactions in computer-assisted math game making should be critical for the processes of creative composition and construction to be transformed into mathematical reasoning and understanding.

2.2.1. Interaction between students and math game design

As students design a math game, they will need to explore and represent their understanding of a math concept or interpretation of a math problem via scenarios and objects in the game world. The process should help students articulate, self-check, and constantly accommodate their prior mental framework (Shaffer, 2005). Thus design will act as a vehicle for the articulation and application of what they have learned about math. Prior research on design-based math learning also reports that students can "formulate mathematical conjectures during, and as a consequence of, their design activity" (Kafai, 1995; Shaffer, 2005, p. 7). These mathematical conjectures, representing students' ability to "form inferences about general principles from specific observations", are significant for mathematical understanding (Davis & Hersh, 1982; Shaffer, 2005, p. 7).

2.2.2. Interaction between students and design-based computing

Recent studies on computer game making by youth reported that design-based computing, or game programming, is motivating, reinforces esteem, and develops young people's higher-order thinking – computational thinking in particular (Habgood et al., 2005; Robertson & Howells, 2008). A variety of visual programming and game design applications, such as *Scratch* by MIT Media Lab, *Kodu* by Microsoft Research, *Game Maker* by YoYo Games, *Alice* by Carnegie Mellon University, have been used and investigated in previous learning-through-game-making studies, with evidence suggesting their positive effect in reinforcing computational thinking (Hayes & Games, 2008).

Computational thinking, being a universally applicable attitude and a fundamental reasoning skill, refers to "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Wing, 2011). The core process of computational thinking is to create and make use of different levels of abstraction in order to model and solve problems more effectively. Notably, computational thinking encompasses *mathematical thinking* in that one needs to think algorithmically and has the ability to apply mathematical concepts such as induction to develop effective problem solutions (Wing, 2011). Mathematical thinking (e.g., Powell, Francisco, & Maher, 2003; Schoenfeld, 1992) is a critical lens for mathematics education. According to Schoenfeld (1992), there are five aspects of mathematical thinking: the knowledge base, problem solving strategies or heuristics, monitoring and control, disposition toward mathematics, and engagement in mathematical practices.

An examination of the literature on the two thinking skill sets indicates a significant overlap. In particular, both computational and mathematical thinking have been considered essential aspects of everyday reasoning. Both require reasoning abstractly and quantitatively, representing or modeling problems via mathematical symbols and algorithms, attending to precision, looking for structure and regularity, and productive disposition toward computational practices. Thus the activity of computer math game making, requiring design-based programming and hence computational thinking, can be a natural embodiment of mathematical thinking (e.g., Mitchell et al., 2007; Roblyer & Edwards, 2000; Shaffer, 2005).

2.2.3. Interaction between students and peers during collaborative design

In the literature of math education, self-explanation was identified as an important factor for enhancing and correcting students' understanding of mathematics concepts (VanLehn, Jones, & Chi, 1992; Wong, Lawson, & Keeves, 2002). The prior research on design-based learning reported student-designers' performance of self-explanation and communication of domain knowledge via a composite representational system, including design drawings, discourse, and artifact prototypes (Kolodner et al., 2003; de Vries, 2006). Previous learning-by-design programs frequently involved structures of design-based collaboration and peer critique (Kolodner et al., 2003; Shaffer, 2005). Students in a math-game design group are expected to co-explore alternatives of a gameplay challenge to embed what they have learned about mathematics, describe and demonstrate specific design moves, and present a particular issue to be solved, thus getting opportunities to elaborate more conceptual insights with each other. Hence self-explanation and communication in mathematics become necessary and motivated during game composition and construction processes.

In spite of its promise, research on learning through computer game making, especially that for math, is still sparse. Besides, a recent survey on instructional practices of learning through computer game making suggested that existing instructional strategies and authoring software generally stress the goal of computational thinking while placing little or no emphasis on educating design thinking (Hayes & Games, 2008). Previous studies (e.g., Habgood et al., 2005; Kafai, 1995; Kafai, Franke, Ching, & Shih, 1998) reported that young learners, in spite of receiving concentrated training on game programming toolkits, could only achieve a low level of sophistication or content integration in the design of computer learning games. Particularly, the studies found that when given freedom in determining the centrality of the target subject matter, the learning games that children made typically replicated gameplay mechanics from prior entertainment games and integrated the learning content only as an extrinsic feature or an afterthought to the gameplay process. Although game design can be used as a tool for both personal expression and knowledge reformulation (i.e., using game design as a sounding board to test and evaluate their knowledge representation), the latter process was found to be a lot more challenging and still in need of further examination (Hayes & Games, 2008; Kafai & Peppler, 2012). As such, it is important to examine how design thinking, design-based computational thinking, and content learning (e.g., mathematical thinking) interact with each other during game making.

3. Method

This descriptive, mixed-method case study (Stake, 1995; Yin, 2008) examined the process and nature of learning by design within the contexts of computer game making and math learning. In particular, this study addressed the following research questions: (1) Did participating in computer game design and development foster positive dispositions toward mathematics for school children? (2) How did computer game design and development processes foster mathematical thinking for school children?

3.1. Participants

The study was conducted in a rural pueblo school and an urban school with a high percentage of Hispanic students. Sixty-four middle grade students participated in the study, with around 20% of participants being Native American, 80% being Hispanic, and 43% being girls. During the study, these participants were randomly assigned into ten design groups comprising six to seven members, with each group comprising both girls and boys and students with different math competence levels based on their school records. In this study, a design group was considered as the major unit of case, with an individual student designer acting as the secondary unit of data coding and analysis. Before game making, all participants had played a collection of computer math games at their computer classes. Those games designated middle school math curriculum and were provided by credible educational organizations or resource sites. The game-play experience should have awarded the participants with some initial reflection on and understanding of game quests that could be both fun and educative.

3.2. Setting

The computer game design and development activities lasted for six weeks with two 1-h sessions each week. They took place during participants' computer classes at the schools' computer labs. The activity structure of game making was open-ended and following the protocol of participatory game-design studios (e.g., Muller & Kuhn, 1993; Sotamaa, 2005) and that of previous learning-by-design programs (e.g., Kafai, 1995; Kolodner et al., 2003; Shaffer, 2005). Specifically, participants of each design group were requested to collaboratively identify math concepts or skills that they would like to teach to their younger siblings. They then collaboratively made a mini-game that would explain math concepts or afford the practice of the targeted math skills, and can resemble the computer gaming features that they experienced and enjoyed. The game making process comprises paper prototyping and then computer-assisted game design refining and testing.

During paper prototyping, students of a design group were seated together and brainstormed and negotiated their game design ideas via dialogs, drawings, and design notes. Individual members were encouraged to: (a) explain preferred design goals and math topics to each other, (b) describe the design moves (e.g., story, character, scenarios, gameplay rules, and how math is embedded), and then (c) ask design questions to be answered. Other group members, along with adult mentors, acted as critics by asking questions on the related math content, sharing or extending design thoughts, and then exploring answers to design questions.

When a paper-based design prototype was ready, student design groups moved to computer stations to model and program their game design prototypes using *Scratch*. Students were involved game world and character crafting, information and materials search on the Internet, action scripting. During the last design session, every design group uploaded their mini-games to the Scratch website, presented their design products to others in the class, and shared comments and critiques. Design debriefing, facilitated by adult mentors, took place in groups in addition to design activities at the end of each week in order to encourage students to reflect on and connect their design moves with learning experiences.

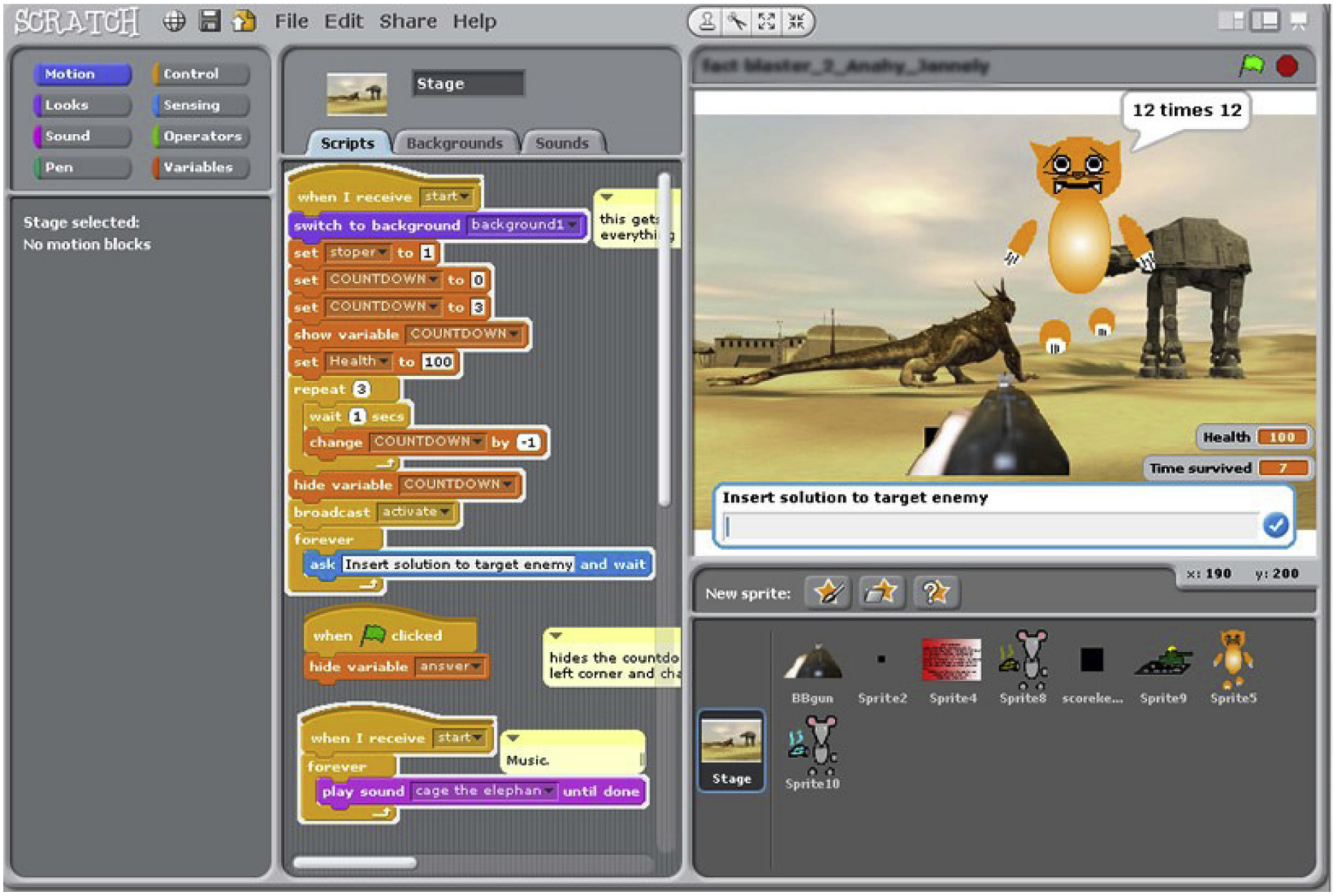
3.2.1. Game making tool – Scratch

Scratch, a programming environment developed by the MIT Media Lab, was used as students' game authoring tool for this project (see Fig. 1). *Scratch* has been implemented and reported on in multiple recent studies that involve computer-supported learning and learning-though-game-design workshops for school students (e.g., Brennan & Resnick, 2013; Kafai & Peppler, 2012). Prior research indicated that *Scratch* is user-friendly for school children who can learn mathematical and computational ideas by creating and sharing Scratch projects. People can upload their games to the developer community site to be played, downloaded, and customized.

Participants of this study received three 1-h training sessions on using Scratch. Scratch game developer community site, with a big archive of Scratch-based games shared by other Scratch users, was also introduced to participants. Participants were encouraged to self-explore existing Scratch games after school, in order to develop a better understanding of the salient functions of Scratch and potentials of Scratch-based games.

3.2.2. Scaffolding

Five graduate students, majoring in education and enrolled in an educational game design course, facilitated all game design/making sessions. They answered questions on math content and game design, gave feedback between design moves, prompted for explanations, and gave praise or encouragement to keep students' confidence high. They also provided children participants with in-design mentoring on Scratch-based visual programming.



```
when I receive start
switch to background background1
set stoper to 1
set COUNTDOWN to 0
set COUNTDOWN to 3
show variable COUNTDOWN
set Health to 100
repeat 3
    wait 1 secs
    change COUNTDOWN by -1
hide variable COUNTDOWN
broadcast activate
forever
    ask Insert solution to target enemy and wait
```

this gets
everythi

```
when clicked
hide variable answer
```

hides the countdo
left corner and ch

```
when I receive start
forever
    play sound cage the elephan until done
```

Music.

Insert solution to target enemy

Fig. 1. Scratch-based game: Calculation blaster.

3.3. Data collection and analysis

Data were collected through in-field observation of design discourses and behaviors, interviewing, and a pre and post-game-making mathematics attitudes survey. Peer debriefing and member checking were part of the data collection (Lincoln & Guba, 1985). Peer debriefing consisted of formal reviews of data among the three coders of the study data. Member checks were conducted with student participants informally during the process of observation and interviewing.

3.3.1. Activity and conversation observation and analyses

Design groups' activities as well as their conversations during every session were observed, with a random selection of them also video recorded. Qualitative task analysis with the observed activities and the content analysis with the activity-associated conversation transcripts were conducted and integrated for the examination of salient design or development events and the potential presence of mathematical thinking within these events. The video-recorded group design activities and conversations were coded every 3 min. Occasionally, a group design conversation or act focusing on the same specific topic or object could extend slightly beyond 3 min; the event was then coded based on its natural ending in order to avoid an interrupted event observation. A total of 156 design events and 336 computer game development events were coded. We started by coding the critical properties of salient actions or instances of design- and computing-related mathematical thinking, and classifying them into *aggregations* that signify the presence and nature of design-based learning (Yin, 2008). Via a systematic coding method (Marshall & Rossman, 2006), we then reduced and summarized the coded data based on the aggregations emerging from the data. Three coders compared, negotiated, and finalized the aggregations and codes after achieving a 100% inter-rater agreement. The coding on the design and development events and the math presence elements is outlined in Table 1. The analyses of the design activities and conversations contributed an initial list of themes depicting nature of design- and computing-related learning events, which were then refined and extended via a constant pattern matching and corroboration with the data collected from interviewing and the design product analysis.

3.3.2. Interviewing

Student participants were randomly selected and interviewed in group after each session, and all of them were interviewed after game making. Semi-structured interviewing focused on participants' perceptions about math game making and associated learning experiences. Examples of interviewing questions included but were not limited to:

> "What was your game about?" "What did you do?" "What would you add or change about this game?" "What do you want your players to learn from this game? Do you think this game will teach? Why or why not?" "Do you think knowing math is important for game creation? Why or why not?" "How do you feel about your game creation experience? What impressed you most? What frustrated you most?" "What have you learned by creating this game? Could you elaborate or give any examples?"

A qualitative thematic analysis (Miles & Hurberman, 1994) was conducted with the interviewing scripts across cases to examine recurring themes, which were then compared and congregated with the findings of in-field observation and conversation analysis.

3.3.3. Artifact analysis

An artifact analysis was conducted with design groups' products – the paper prototypes and Scratch-based games created. The analysis focuses on examining the presence and the way that math content was integrated into game prototypes and how computational and mathematical thinking got reflected in the games' scripts. The analysis with games created was assisted by *Scrape*, an open-source Scratch code analyzer (http://happyanalyzing.com/downloads/) that elucidated the types and frequencies of programming functions and variables used in a Scratch game's action scripts. Specifically, *Scrape* helped to summarize the number of variables, sprites (moving objects), stacks of scripts, and frequency of each type of programming functions (e.g., control, looks, sensing, operators, motion) used in a single game and across a batch of games. The data helped to illustrate the complexity of computational thinking and the presence of mathematical thinking (e.g., a math-related variable or function) demonstrated by the final design product – a game created. Findings from the artifact analysis were triangulated with the observation and interviewing data to present a rich description on the processes of math game design, computing, and design-based math learning.

3.3.4. Survey

All student participants completed the *Attitudes towards Math Inventory* (ATMI, Tapia & Marsh, 2004) right before and after the 6-week game design/making activities. This 40-item, five-point Likert-scaled inventory has been used and validated in prior research on game-based math learning (Ke, 2008). This survey corresponds to the expectancy-value mode in conceptualizing motivation as comprising expectancy (or perceived competence), value (goals and beliefs about the importance and interest of the task), and affective (emotional reactions to the task) (Pintrich & De Groot, 1990, p. 33), and was used to examine a potential change in participants' attitudes or dispositions toward mathematics according to four identified factors: self-confidence, value, enjoyment, and motivation. The KR-20 reliability of the inventory in this study was 0.87.

4. Results

4.1. Mathematical dispositions

A pairwise *t*-test was conducted with the ATMI responses to compare student participants' attitudes towards mathematics before and after game-making activities. The *t*-test indicated a significant result, $t(62) = -2.56, p = .01$. Participants reported significantly more positive attitudes, including self-confidence, value, enjoyment, and motivation, towards mathematics after participating in the computer game making activities.

Table 1
Aggregated categories of game design/development events and mathematical presence/practice.

Code	Category	Description
Game Design Event ($n = 156$)		
Des 0–5	**Design Act**: • Off-task (Des0), 12.2% • Passive (Des1), 10.2% • Management (Des2), 8.2% • Design discussion (Des3), 46.9% • Sketching (Des4), 18.4% • Case reviewing (Des5), 4.1%	• Action in which a design group was involved in social chat or other game-design-irrelevant behaviors. • Action that comprises passive design moments, such as silent thinking or mentor-led explanation. • Action that serves the purpose of team or project management, such as the discussion on team roles and responsibilities. • Brainstorming and negotiation on features and assets of the game to be designed. • Paper drawing and design narrative write-up. • Searching and reviewing exemplary Scratch-based games and other game examples
DesO 0–2	**Design Object**: • N/A (DesO0), 20.4% • Game world (DesO1), 55.1% • Gameplay (DesO2), 24.5%	• Not focusing on game features • Focusing on the design of game world assets, including the backdrop story, setting, game characters and objects. • Focusing on the gameplay mechanism design, such as game challenges, gameplay actions, and rewards.
Des-Math 0–3	**Math Presence***: • No presence (Des-Math0), 69.4% • As a side note (Des-Math1), 18.4% • Presentation (Des-Math2), 10.2% • Representation (Des-Math3), 2.0%	• No presence of mathematical discussion • Math mentioned lightly and typically as a contextual cue. • Presentation of math questions or a math topic is explicit during design (e.g., listing numerical questions to be presented at a cut-screen within the game). • Mathematical representation of a math concept or a math expression is explicit during design (e.g., describing the *XY* coordinates in the map feature).
DesI 0–4	**Design Inspiration**: • Other (DesI0), 36.7% • Gaming experience (DesI1), 8.2% • Ed. gaming experience (DesI2), 16.3% • Everyday life experience (DesI3), 16.3% • Identity expression (DesI4), 22.4%	• Design ideas that are not obviously related to designers' experiences or identity expressions. • Design ideas related to prior gaming experiences. • Design ideas related to prior educational gaming experiences. • Design ideas related to designers' everyday life experiences and activities. • Design ideas related to designers' self-image or personal preference expression.
Scratch-based Computer Game Development Event ($n = 336$)		
Dev 0–6	**Development Act**: • Off-task (Dev0), 2.4% • Resource searching (Dev1), 4.8% • Art crafting (Dev3), 28.6% • Audio inclusion (Dev4), 7.1% • Script analysis (Dev4), 6.0% • Scripting (Dev5), 27.4% • Test play (Dev6), 23.8%	• Action in which a design group was involved in social chat, play, or other game-development-irrelevant behaviors. • Searching for materials through Internet, such as visuals, sound effects, and other materials for game authoring. • Editing the visuals in the Scratch game, such as costumes of a sprite. • Selecting and including sound effects to the game. • Reviewing and analyzing Scratch commands and scripts to plan coding • Coding or scripting of actions and animations in the game. • Test play the game actions or scenes made.
Dev-Math 0–3	**Math Practice***: • No practice (Dev-Math0), 50% • Numerical calculation (Dev-Math1), 9.5% • Analytical reasoning (Dev-Math2), 25% • Quantitative reasoning (Dev-Math3), 15.5%	• No obvious evidence of mathematical practice or thinking • Practicing numerical calculations during game making or test play. • Demonstrating analytical reasoning (e.g., planning the interaction and sequence of diverse animations, inductive and deductive reasoning during script analysis and debugging). • Demonstrating mathematical conceptual comprehension or application, and/or model a real-world problem using math symbols and expressions.

Note: * Math presence or practice was coded as a rank or ordinal variable. A higher-level category may include occurrences of lower-level categories. For example, a game-development event demonstrating quantitative reasoning may also include the processes of numerical calculation and/or analytical reasoning.

During interviewing, approximately 91% of participants reported that they have enjoyed making computer games. The frequently-reported engaging part of game making, citing children participants, included "playing our own game," "making it with the friends," "getting different ideas and putting them into a game," and even "messing around with the scripts, figuring out which controls to use, and moving the things (game objects) the way you want!" Only 52% of participants, however, mentioned math learning, especially numerical calculation, as part of game design and development.

The qualitative data indicated two salient patterns of participants' mathematical dispositions during game making: (a) Lack of a connected representation of math, and (b) a tendency to engage and persevere in effortful analytical reasoning.

4.1.1. Lack of a connected representation of math

Multiplication or division of integers was a predominant math topic depicted in 50% of the paper prototypes and 70% of the final games made. During interviewing, participants typically reported that they felt most confident about integer calculation and also considered it "most useful" in everyday life. Other math concepts, such as *XY*-coordinate system, area, and perimeter, were mentioned in participants' paper prototypes but did not get integrated into the games created. Less than 20% of design prototypes or products involved the topics of fraction, proportion, or percentage even though they were highlighted in math games that participants played at their computer classes.

Math content in games made was typically presented as a non-situated practice of mathematical equation solving. A total of 10 mini-games were made by 10 design groups. Five of the games put mathematical representation, in blunt symbols and expressions, as an extrinsic feature appended to sprites or a segmented post-action of game play. For example, in a game called "Calculation blaster (Fig. 1)," a list of integer calculation questions (e.g., "12*4 = ?") were matched with and showed on the surface of a set of shooting targets; the player would need to type the answer to each calculation question to shoot the question carrier. The faster and the more questions were solved, the more targets hit and the safer the game agent was. Another game called "Got caught" made the player maneuver the arrow keys to move a Pac-Man-like character through the maze (i.e., the school in this case) while avoid being caught by the enemy (i.e., the teacher). If caught, the player would need to answer a cut screen of numerical calculation questions before he/she can continue the game. As demonstrated by these game examples, math was depicted as a negative occurrence – hurdles to jump over. Their designers commented, "You will need to solve a math question before the game can keep going…Before you can kill, you have to do equation."

Two of the games created by participants provided a real-world task setting for the quest of numerical calculation. Shopping at the mall or playing as a cashier, for example, were presented as the background for the cut-screen numerical calculations by multiple design groups. Only one game represented math as a contextualized, real-world phenomenon. A game called "Car distance (Fig. 2)" simulated the distance that racing cars would cover when varied values for the cars' time and speed were put in; the gameplay was to decide on the set of time and speed to get a car to the destination.

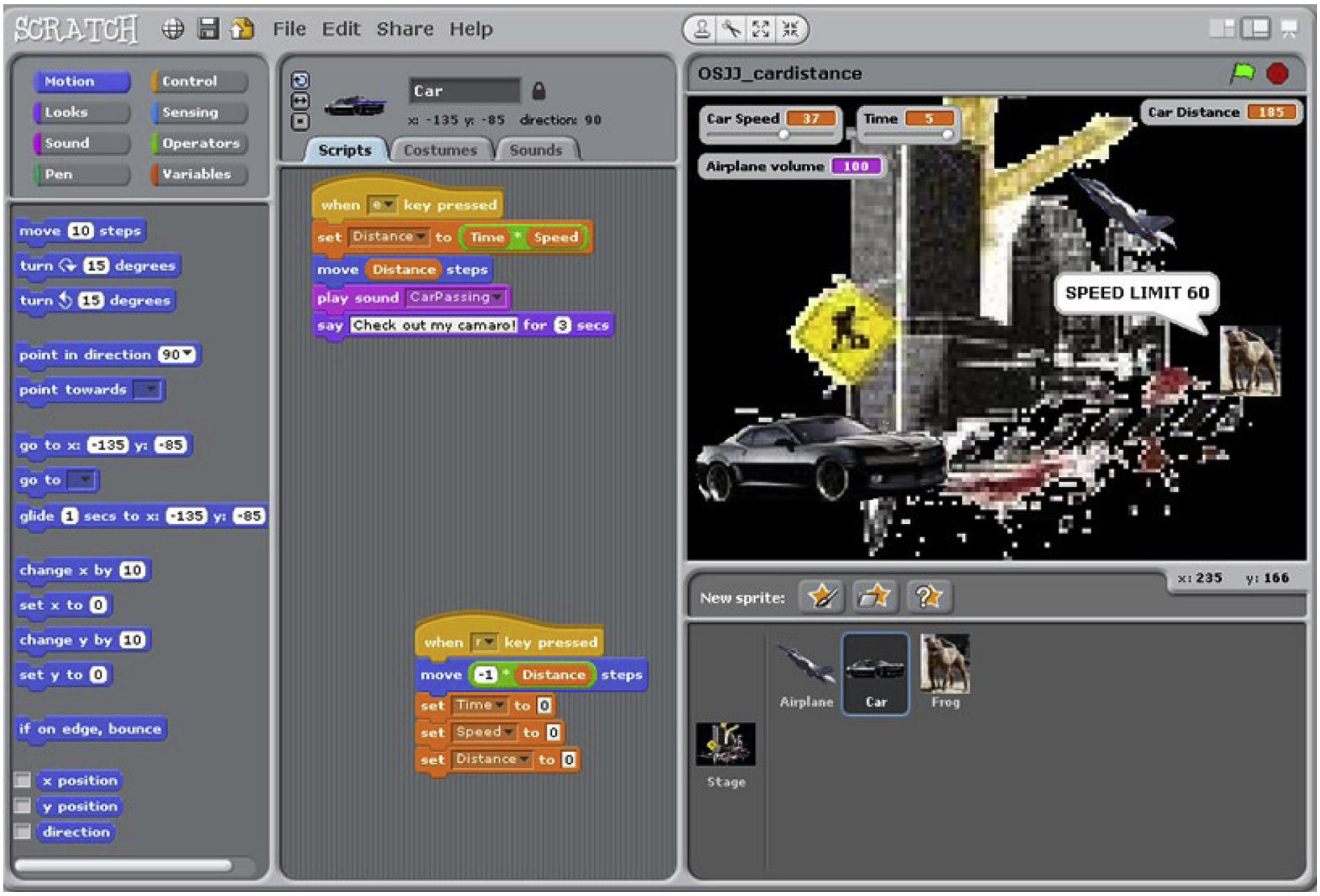
Two of the games did not integrate math content into gameplay. The design group of a game called "Plane destruction," for example, confessed that they focused on creating a fun play (i.e., using arrow keys to aim cannon to hit the plane). They did not demonstrate awareness or purposeful design effort on connecting play with math, until a facilitator cued them that they could mark the cannon's pointing angles with math symbols of 90° or 45°. For these participants, math appeared to be an afterthought during game design, especially when it involved a concept that they possibly did not have a solid understanding and hence could not connect with.

During post-session interviewing, participants in the later phase of game making, in comparison to at the beginning of game making, provided obviously more reflective comments that depict math as a recursive process of interaction between formal learning and daily life experience:

> "Math is everywhere, like math is in everything you do."

> "I learned that even though math is everywhere you still have to learn it and when you learn it you will see it more in life that it will be in everything you do. Like cooking, technology, practically everything."

> "I need to like math, because you need math in your life. Like we made a cake for my brother's birthday yesterday, we need measurement and (to) mix stuff, like that we need math."



Car
x: -135 y: -85 direction: 90

```
when e key pressed
set Distance to (Time * Speed)
move Distance steps
play sound CarPassing
say Check out my camaro! for 3 secs
```

```
when r key pressed
move (-1 * Distance) steps
set Time to 0
set Speed to 0
set Distance to 0
```

Car Speed 37
Time 5
Airplane volume 100
Car Distance 185

Fig. 2. Scratch-based game: Car distance.

4.1.2. *Tendency to engage and persevere in effortful thinking*

Participants were frequently observed engaging in effortful analytical or quantitative reasoning during game making. In around 41% of the coded game-development events (please refer to Table 1), participants demonstrated effort in reasoning analytically or quantitatively. Computer game development, usually involving an iterative process of script analysis, composition, and experimentation with error diagnosis, was described by a participant as a "hard but fun" process that was illustrated by the following examples.

> Sally[1] and her group-mates sat together before a computer and were making a game quest of "Calculation blaster." They added various shooting targets, such as zombies and monsters, onto the game stage, but ran into a bug during test play – the movements of these targets were not coordinated and hence they did not make sequential appearance as expected. To fix this issue, Sally and a partner were reviewing the action scripts associated with each target sprite (an average of 11 stacks of command). They frequently pointed their fingers at certain line of scripts on the computer screen and were involved in discussion or deliberating. Time elapsed and they were found kept testing and refining different lines of scripts. Finally, Sally giggled with her partner and claimed, "Ok, got it!" They managed to use a variable to control a sprite's movement speed and customized its value for different sprites.
>
> Joe and Matt's group wanted to add a new sound effect to the takeoff action of an airplane sprite in their racing game. They were refining and testing the Scratch scripts to synchronize the sound effect and the takeoff action. The revision and experimentation iterated so much that a mentor commented with a smile, "I have heard that airplane 50 times today!" Joe and Matt cheered when finally the sound and action got coordinated, "Oh yeah, there you go!"
>
> Elsa, sitting with an adult mentor, was modifying a shopping-mall game. Elsa was a 7th grader in the special education program for both reading and math. As reported, she had difficulty with even single-digit calculation. Elsa was reviewing the function panel in Scratch to seek the stack that controlled speech presentation. She put her finger onto the screen, reading scripts line by line. Six minutes passed by before the mentor was heard remarking excitedly, "Good going!" To her surprise, Elsa, who was smiling brightly, managed to decompose the codes to identify the target function. The mentor then pointed to the line (the "say" function) on the computer screen and probed, "Ok, let's change here." Elsa dragged a "say" function from the function panel in Scratch to the game stage, and fitting it into the stack of scripts. She then started to type her math questions (prewritten on a note paper) as the output of the function. During the remaining session, Elsa was working on two computers, with one computer displaying the current shopping game and the other one being her working station. She went back and forth between scripting and test-play, deliberately and attentively. The level of independence and self-assurance she demonstrated was never observed before.

In the above examples, participants persevered in decoding variables and functions that underlie a gaming action, seeking solutions, and planning and monitoring trials. Their engagement with effortful analytical reasoning and the act of attending to quantitative precision during design-based trouble-shooting resembled what was required for mathematical thinking. Math-disadvantaged participants, without exception, demonstrated unexpected proactivity toward effortful thinking. Elsa, for example, showed her capability of reasoning about scripting and her willingness to perform self-learning and regulation during design-based problem solving. The Scratch authoring tool supported instant result checking for scripting crafting, thus providing informative competence feedback that reinforced children designers' motivation for iterative trial and error learning. As observed, participants in general expressed the sense of achievement with every successful trial and de-bugging practice.

Another potential reason of participants' persistence with design problem solving was their sense of ownership or agency during game authoring. A prominent observation was participants' enjoyment with playing their own games. Around 26% of coded game-development events focused on test play. This observation was aligned with a recurrent comment by interviewed participants on game-making, "It was cool because you get to design it however you want and you get to play it too with your characters that you put on it."

4.2. *Activation and suspension of mathematical thinking during game design*

It was observed that game design, being experience driven, reinforced participants' reflection and exploration of everyday mathematical experience. Their design thinking, however, revolved around game-world or story crafting more than math portrayal.

4.2.1. *Reflection on mathematical experience during game design*

In the design phase, around 47% and 18% of the coded design events were design discussion and paper sketching respectively. Participants' design conversations and paper prototypes indicated that they sought design ideas from everyday experiences and games that they played and enjoyed. Of the coded design events, 25% reflected participants' memory of prior games played, 16% involved reflection on their life experiences, and 22% focused on an expression of their self-images or hobbies. The following examples illustrated events of experience-driven design, and hence experience-triggered math reflection and representation.

> In the design sketch, Joseph and Enrique drew a jungle on a map with the x-y coordinates marked. The depiction of the map resembled that of a math game that they had played in computer class. The portrayal of the game world, encompassing transportation vehicles, weapons, trees on slopes and terrains, reminded of "Call of Duty," a game that both designers favored and played after school. During design talk, Joseph's group described the game, titled "Mystery around the World," as an adventure in which the hero needs to save his/her dog. The story replicated that of "Lure of Labyrinth," a math game that they played before. Integrating the previous two game cases, this mini-game started with the hero trying to acquire weapons by "solving math problems to get money." The game credits were briefed as, "Passenger = 100 pts, tank = 250 pts, plane = 150 pts, boat = 300 pts." On another sheet, the group listed math questions or challenges in their game, "2(6x5)/10x12; 10x12; Area, $\pi = 3.14$, diameter, circumstance; 5.2x20, and 'What is the percentage of 0.02?'"

---

[1] Participants' names in the paper are all pseudonyms.

> Gus described, “The game was about shopping in the mall. You walk around and shop around. And if you want to get money, you got to do some math problems. We actually like to go to the mall.” Leroy added, “We made the stores that we usually go, like Wal-Mart and Goodwill.” They then explained why they viewed shopping as fun, “Because at their house it is always boring.” In the design sketch submitted by the group, the game was titled “Family Dude” and game actions were “changing the baby’s diaper, going to the grocery store, and doing multiplication for shopping.”
>
> Giovanna described her idea of math integration to peers, “Well, I would change (it) to like every time you jump rope like 5 times 3, you have a score.”

As shown above, children designers learned from and simulated the game features of the previous game cases (e.g., a map representation of x-y coordinates, the depiction of area and shape via landscape, and a cut-screen presentation of math equation solving in between game play). They were also found exploring and sharing observations about after-school math implementation or their daily-life mathematical experience (e.g., shopping and rope jumping) during game design. Participants’ reflection and communication on their mathematical experience in either game play or everyday activities got activated during design.

Participants’ portrayal of everyday math during game design, initially, tended to be work-related: change calculation by a cashier, tip calculation by a waitress, and food mixing by a cook. They seemed to have difficulty associating their leisure involvements (e.g., text messaging, sport games, hanging-out with friends, entertainment game-play, or listening to music) with math, as the following conversation exemplified:

> Mentor: What do you do for fun?
>
> Several boys answered: Play football, soccer, or basketball.
>
> Mentor: Great! So does that involve math? (Boys all shook their heads) Are you sure? Think more about it?
>
> Boys: I don’t know.
>
> Mentor: If you are a coach of soccer or basketball, what would you need to do?
>
> A boy finally answered: You need to count, like time.
>
> Mentor: Yes! And more than that, you also need to study scores and how players played, to judge the probability of winning…
>
> Mentor turned to another child: “So what you do you do for fun?”
>
> The child: “Listen to songs.”
>
> Mentor: “Does that involve math? Can you use music to teach math?”
>
> The child answered quickly and definitely: “No!”

The mentor then cued the child that the tempo of music, such as beatboxing, can be used to teach ratio and fraction. The child’s face shined when he heard the term beatboxing, but showed disbelief on the presence of fraction or ratio in beatboxing. This observation implied that detecting or discovering math in diverse everyday phenomena can be confined by a young learner’s math competence or prior perceptions of mathematical experience, and was still not a spontaneous occurrence.

To further facilitate design talk and creative thinking on math representation in games, mentors used probes such as “*Where* in real life have you used or observed math?” and “*What fun* activities involved math?” Obviously, participants responded to the previous question more actively, which corresponds with the finding that around 55% of the coded game-design events revolved around the design of game world whereas only 25% of them were about gameplay actions or rules. Consistently, participants’ design sketches depicted the setting and context of a math task obviously more than the task or action itself.

4.2.2. Game world and story crafting more than math portrayal

Participants spent obviously more time and effort negotiating on the design of the game world (e.g., visuals and sound effects) than exploring on the representation and integration of math content. Only around 31% of the coded game-design events made reference to math. Mathematical discussion typically occurred as an ‘appendix’ during the design talk, as the following observation note exemplified.

> George insisted that Mexico (where he was born) should be the background for the last level of their adventure game and argued with his teammates on the name of characters. Their mentor, intending to direct the group’s attention to the design of gameplay and hence math integration, probed, “So what will the characters do?” Ignoring this probe, Vanessa shifted the topic to the background music, “When I do my homework I usually have music on. The player can turn it on and off and choose music (in this game).” The others concurred. The mentor probed again, “So you have background music. But what will characters do? Will they just walk around?” After a long silence, Jasmine finally answered, “They have to find someone.” George continued, “And they are brothers.” “Or sisters,” Vanessa added. The group’s design talk resumed its vibrancy when they started elaborating on how brothers and sisters get kidnapped, how the hero would save them via helicopter, and how game characters look like. Yet nobody touched on gameplay actions or math integration. The mentor probed for the third time, “But what will be the challenges? How or what math will be used in the game?” Nobody answered and George yawned.

Apparently, children designers in the above case focused on telling a story rather than designing gameplay. Setting, story, and character had occupied the design talk. Correspondingly, the group’s design sketch predominantly carried visuals and narratives for the game world, objects (e.g., planes, cars, and weapons), characters (e.g., names, “funny” outlooks, and “different types of clothes”), and sound effects, with less or little portrayal of game challenges and math content. Overall, only a small percent (25%) of the coded game-design events focused on gameplay. At the same time, a Chi-Square analysis indicated that a design event focusing on gameplay, in comparison with that on game world, tended to have a higher possibility of being math referent, $\chi^2(1, N = 156) = 11.57, p < .01$.

At times, participants would treat game design as a channel for identity expression. They projected self-images or preferences onto the game world and characters, as the following example highlighted:

> Eunice and his teammates were doing paper sketching. Eunice drew a racing car, with a line of math equation ("2x2") on its side. He then wrote, "I would look like a famous person on my Hammer to all the people who would be curious about me." Briana provided a similar character sketch, "I would look like a famous people with the coolest car of the world." Rafael drew a Nissan Skyline car and a character named "Rafael." Among seven group members, only Julissa and Robin briefed gameplay. Julissa wrote, "Making a racing game, roads, streets of LA, gaining scores, losing if you hit a car." Robin wrote, "Racing, you unlock cars as you win missions (calculation)."

In the above case, math content was designed as 'a side note' to the story elements (e.g., the racing car numbers) that composed the imaginal identity of designers (e.g., being famous with a cool car), more than the core of gameplay (e.g., doing numerical calculation to unlock cars).

The observation on the lack of mathematical discussion, along with the preponderance of digital storytelling and identity expression during game design, may relate to the complication involved in designing a math game. During post-session interviewing, around 67% of participants reported the process of game design as challenging. Among their comments, Robin's was demonstrative, "Like you have to play, you have to make your characters, plan all of that stuff, and at the same time you have to use the math."

4.3. Situated mathematical thinking within computer game development

"I learned that most math is involved in creating games." This participant quote highlighted a prominent observation that mathematical thinking and content experience were intertwined with the process of game development (i.e., the modeling and programming of the game design ideas).

4.3.1. Experiencing math during computer game scripting

Based on an artifact analysis with the games created, there is a positive association between the complexness in game scripting (i.e., in terms of the number of sprites and types of programming functions used) and the degree of math integration (i.e., none, extrinsic, and intrinsic), $r_s = 0.63, p < .01$. During post-program interviewing, more than 52% of participants reported that they learned math (rather than design or computing skills) from computer game development. The following quotes exemplified how they perceived their usage of math concepts during computer game scripting:

> "Like you're going to get the dimensions of the maps, and you had to know the Y and the X axes to like find where to put your objects and stuff."
>
> "Like you have to go to find the value of the square and the distance, like how big the square is, and the distance of the car and how fast it could go. First I thought it was pretty hard, but now I learned a lot about it."
>
> "You have to know how to move it from one position to the other, so like negative numbers, positive, multiplication, and everything else."

The above quotes demonstrated participants' mindful experiencing with math concepts and operations that got multiple representations via concrete objects (e.g., a sprite to be moved), math symbols (e.g., *XY* coordinates), language (mathematical discourse during collaborative scripting), as well as images (e.g., a map or a square geometric shape).

During computer game authoring, participants had to comprehend and apply the arithmetic and algebraic expressions embedded in programming. Script reviewing and modification enabled children to discover connections between symbolic and concrete-object-based representations of a math concept and hence develop an integrated understanding of it. The following example demonstrated how children designers got active experiencing of the concept *variable* during game scripting.

> A mentor was guiding Seidy and her design partner to modify Scratch scripts in order to present numerical questions in the game scene. She highlighted all relevant cases of the focus *variable* in a stack of scripts to explain its definition and property. Seidy and her partner appeared focused and comprehending, quickly managed to enter a numerical question as the *value* of this "presentation" variable, and tested it by rerunning the game. They continued to customize the variable's value (i.e., questions to be presented) for different game levels. The mentor then encouraged them to change the default speed of question presentation by setting a "speed" variable in the scripts.

As shown above, game programming provided a meaningful context for the multimodal interaction with math concepts (e.g., *variable* and *value*). Participants could corroborate the symbol (e.g., a variable's symbolic representation in scripts) with instant visual or pictorial feedback (e.g., the display of an updated game scene when the coding was tested) for comprehension development.

The analysis indicated that a total of 76 variables were used in the 10 computer games made. These variables were embedded in either algebraic expressions with relational operators (e.g., a control structure like "if $x = [\ ]$, then") or numerical operators (e.g., using the line of "$x = x–5$" to set a slower speed). It could be interpreted that participants had also interacted with those algebraic expressions and operators when coding variables.

Another example for scripting-based math experiencing is the practice of $x–y$ coordinate and negative/positive number to animate a sprite, via the function "go to *x*: (value) *y*: (value)." This specific function had a total of 856 occurrences in the 10 games submitted. Notably, the *XY* coordinates of a sprite's position in Scratch is ever present on the top of the stage, and change dynamically with the sprite's movement. This feature helped students better understand and apply *XY* coordinates and positive/negative numbers in action coding, as the following observation note illustrated.

> Oscar, José, and Ramón were coding a racing-car animation. Oscar clicked on the car on the game stage and commented, "Now we have to figure out from where to where. This's where it's going to start, and that's where it's going to be. So we have our X, Y coordinates now, right?" The other two noted down the XY coordinates of the start and destination positions, with which they coded the car's movement as: "go to x: −239 y: −147; glide 3 secs to x: 201 y: −152."

4.3.2. Reasoning abstractly and quantitatively

It was observed that participants learned to reason with a problem abstractly and quantitatively when trying to model and coordinate animations and events in a game. Among the coded game-development events, around 25% of them involved the process of analytical reasoning (e.g., planning the interaction and sequence of diverse animations, testing and debugging). Another 16% also demonstrated mathematical concept application and quantitative reasoning, in which participants learned to model a real-world problem using math symbols and expressions. For example, a design group coded the problem of car-distance calculation via an algebraic expression including multiplication operator and three descriptive variables – "set 'Distance' to ((Faster) * ((Time) * (Speed)))." In the expression, "Faster" was a variable representing the real-world 'power-up' mechanism (e.g., pushing wind or stepping on gas). Participants also learned to attend to precision when they estimated, tested, and modified the assets of an animation, such as the number of steps for a sprite to move, the amount of the 'wait' time for background switching, and the sequencing or synchronization of different animations in a single game scene.

> Laura was amazed by the animation in the game (named "Motion!") made by Jair's design group and asked, "How come they (moving sprites) will not wreck into each other?" Jair and his teammates, confidently and in detail, explained the reasoning process underlying the action flow. They stated the need to calculate and set exact amount of seconds for each sprite's "slide" action. They then described how they set the "wait" time of the first background to "10 s" before switching it with the second one, and how they set the "hide" time of the second sprite also to "10 s" so it can take the stage along with the second background right after the hiding of the first sprite.

The above example actually illustrated two foundational processes of game programming –procedural (i.e., sequencing of actions) and object-oriented (i.e., setting the property and function of a game object). Both processes comprised math conceptual understanding and quantitative specification (e.g., distance, time, and position calculation) and required analytical thinking (e.g., on the relationship or interaction among sprites and actions). Thus, participants got to practice abstract and quantitative reasoning when modeling the action flow in a game.

On the other hand, not all participants got accustomed to the practice of abstract and quantitative reasoning in game programming. Around 42% of the coded game-development events did not demonstrate obvious mathematical practice or reasoning. Correspondingly, during post-program interviewing, around 52% of participants reported game programming as the most challenging part of game development, "You know what you're going to do but you don't know how to get it all together." Mentor support thus became critical to participants, as one of them commented, "As long as there is instruction and help I pretty much get it." However, the high participant-mentor ratio in the program made it difficult to provide instant, individualistic assistance, as the following example illustrated.

> José and Oscar managed to calculate the XY coordinates of the target position in the "slide" function to get the car move. But they could not figure out why "the car won't go back." A mentor explained the operation of "resetting" the sprite's position by restoring the initial XY coordinates and encouraged them to try it out. The issue got solved, but quickly José and Oscar ran into another programming problem when all mentors were busy assisting other design groups. They were waiting for help, looked frustrated.

Related to the observation on the lack of mathematical or abstract reasoning in a moderate percentage of game development events is the finding that in these game development events participants were involved in esthetics computing (e.g., customizing the costume of game sprites or including sound effects) instead of script analysis or coding for gameplay actions.

5. Conclusions and discussion

The study findings indicated that game design and development activities enabled active experiencing of math content for middle school students and engaged them in thinking mathematically. On the other hand, the predominance of storytelling and game-world crafting during game design and the demanding nature of game programming would suspend students' interaction with the math content. It is observed that the processes of design and computing could merge, and at the same time, dissect themselves with the process of knowledge construction or learning during computer-based educational game making.

5.1. Game making and math disposition development

The survey results suggested that students have developed more positive dispositions toward mathematics during computer math game making. The finding supports the report of prior research that learning by design in general and computer game making in particular reinforce positive disposition or epistemic belief toward domain knowledge (Gee, 2005; Kafai, 2006; Shaffer, 2005). Qualitative observation suggests that students, including those who are math-disadvantaged, tend to engage and persevere in analytical and quantitative reasoning involved in development-oriented problem solving. This observation is consistent with the report by Robertson and Howells (2008) and Vos et al. (2011) that children in computer game making displayed enthusiasm and motivation for learning and determination to reach a high standard of achievement. It also supports the finding of Fortus et al. (2004) that learning gains occurred for both high and low achievers after their participation in the design-based science learning program.

The literature on *need for cognition* suggested that instant competence feedback and feelings of personal satisfaction derived from cognitive challenges should reinforce an intrinsic motivation to engage in effortful cognitive endeavors (Cacioppo, Petty, Feinstein, & Jarvis, 1996). In this study, students' persistence in effortful cognitive endeavors may relate to the instant competence feedback afforded by the authoring tool Scratch. Scratch-based game coding, by enabling ever-present and dynamic visual feedback, act as a sound board or sandbox for students to test their conceptual comprehension or abstract reasoning. Specifically, student designers can instantiate the meaning of abstract concepts (e.g., variable, value, and algebraic expression) or test their reasoning (e.g., on background switching and sprites' movement management) instantly and iteratively via different objects and within multiple scenarios, thus developing feelings of enjoyment and mastery in thinking. The study also found that students enjoy test-playing their own game quests because they value the ownership and personal identity expression during game design and authoring, which may be another reason why student designers persevere in design- and development-oriented problem solving. This interpretation supports the perspective of constructionism on creating active learning via authorship and identity expression (Kafai, 1995, 2006; Harel, 1991).

The interviewing results suggest that math game making have helped to strengthen students' awareness of mathematical representations in daily life. At the same time, students still predominantly discuss and depict mathematics as mainly numerical calculation during game making. They appear to perceive mathematical practices as work-related and formula- or equation-confined. They also lack mathematical awareness about math concepts and procedures that were novel or abstract, and hence have difficulty representing or reformulating them in the design products. These views of math of student designers are reflected in the math games that they made – integer calculation is heavily depicted as math in the game, and it was inserted or presented more as an extrinsic add-on to the game-play actions (e.g., in a cut scene that is segmented from game scenes). Certain games even lack obvious connection to math content, which corresponds with the interviewing finding that students have difficulty connecting math with their 'play' activities. Prior research has argued that mathematical awareness is the source and basis for the choice of mathematical practice or procedure, and the links between the requirements of math in school and in real world should be made explicit (Boaler, 1993; Mason & Davis, 2013). This observation helps to explain the finding of the current and previous study that learning content has been integrated more as an afterthought when youth make an educational computer game (Habgood et al., 2005; Kafai et al., 1998). An implication is that it is important for a mentor or facilitator in the design-based-learning program to integrate the support for mathematical awareness development during in-the-moment design facilitation.

5.2. Mergence and dissection of design and math learning

This study found that students' design thinking is usually experience driven, which stimulates them to deliberate on their observations and experiences of math-related phenomena and problem solving in previous life or gaming activities. On the other hand, it seems that students tend to highlight the locales rather than activities of everyday math. It is not natural for them to identify math representation in leisure involvements and to extract math reasoning processes that underlie daily activities. Cognitive scientists have highlighted the role of knowledge in design thinking in that one needs knowledge in order to see how to apply or extend it creatively" (Chi, Glaser, & Rees, 1981; Sternberg, 2006). From this perspective, content-specific design thinking that integrates math problem representation can be challenging for children designers, especially when they lack the awareness to formulate daily problems using math as a thinking or representation tool. An implication of this finding for learning by design is to provide purposeful training on students' awareness and capability of identifying math in diverse activities and involvements. The study findings also imply that the way students depict those math concepts in game design relates to how they perceive math in daily classrooms and prior educational games. Compatibly, prior research on educational game design by children reports that young learners demonstrate a low level of sophistication in their game design because their games replicate gameplay mechanics from entertainment games or games that include math content as an addendum (Habgood et al., 2005). A suggestion for the future learning-by-game-design projects is that children designers should be encouraged to play and reflect on a selection of exemplary educational games that highlight meaningful and intrinsic integration between learning content and core gameplay mechanics. Constructive experiences with good game cases, as the study finding implicated, can be transferred to children designers' future design solutions.

The study findings indicated that children designers tend to focus on storytelling and personal identity or preference expression during game design, which is consistent with the perspective of constructionism that learning by design enables personal expression (Kafai, 1995, 2006). Correspondingly, design thinking and talk are oriented around game world and narrative rather than gameplay mechanics and content integration. In other terms, design has acted as a vehicle more for artistic expressions than that of prior mathematics mental framework as expected. As such, there should be a fine-tuned trade-off between using design as identity expression and using it as 'mathematical conjectures' expression in future learning-by-design projects (Shaffer, 2005). Future studies on learning by design should examine the scaffolds, activity structures, and other project features that would intrinsically integrate mathematical expression into identity expression during creative design.

5.3. Mergence and dissection of computing and math learning

The study indicated that children designers get to experience a connected representation of math concepts during game construction. The computer-based game authoring environment affords both symbolic and concrete, object-based representation of math concepts, thus helping children to develop connections between formal and informal conceptual understanding. Moreover, the practice of investigating, composing, and modifying the Scratch scripts for animation making enable children designers to abstractly represent a process and reason with a problem via mathematical symbols and expressions. In other terms, computer-based game authoring has acted as a *manipulative* for the development of mathematical thinking. The study findings support the claim of the prior research that computer game making can act as a powerful learning environment or a "microworld" for children to actively explore, represent, and test their domain knowledge and skills (Habgood et al., 2005; Kafai, 1995, 2006; Robertson & Howells, 2008; Shaffer, 2005; Smeets, 2005).

On the other hand, the study found that Scratch-assisted game programming is still a daunting task for children designers. Children designers have demonstrated less competence in coding game dynamics (e.g., interactions and animations), which can prevent them from fulfilling their initial game design plan, cause frustration, and hence impede their involvement in more complicated game dynamics authoring and mathematical thinking. A potential solution, as Kolodner et al. (2003) suggested and the mentors in this study reported, is to make children designers 'modify' a game before letting them develop a game from the beginning. Specifically, the project can offer a collection of template games that provide the basic ingredients and differ in the degree of complexity in scripting (e.g., number of sprites and stacks of codes), and have them act as the design prototypes to be modified or further developed by children designers.

5.4. Summary, limitation, and future research

This descriptive, mixed-method case study mainly intends to describe the phenomena of design, computing, and content learning in a game-design-based learning project. This study indicated that it is important to explore particular processes and features of design-based learning that can engage students in making meaningful integration between the targeted content knowledge and the products designed. However, the findings are generally descriptive and qualitative in nature, and hence lack power to be generalized to a broader population. The data on learning are collected via process evaluation instead of pre- and post-program knowledge tests. As such, future research on learning by design or making should further investigate the relationship among design thinking, design-based computing, and content learning via design-experiment-based intervention studies. Research should also explore on design-based learning aids that will help the learner better experience and practice content knowledge during design or making.

References

Boaler, J. (1993). The role of contexts in the mathematics classroom: do they make mathematics more "real." *For The Learning of Mathematics, 13*(2), 12–17.

Brennan, K., & Resnick, M. (2013). Imagining, creating, playing, sharing, reflecting: how online community supports young people as designers of interactive media. In *Emerging technologies for the classroom* (pp. 253–268). New York: Springer.

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

Cacioppo, J. T., Petty, R. E., Feinstein, J. A., & Jarvis, W. B. G. (1996). Dispositional differences in cognitive motivation: the life and times of individuals varying in need for cognition. *Psychological Bulletin, 119*(2), 197.

Chi, M. T., Glaser, R., & Rees, E. (1981). *Expertise in problem solving*. pp. pp-7. Pittsburgh: Learning Research and Development Center, University of Pittsburgh.

Davis, P. J., & Hersh, R. (1982). *The mathematical experience*. Boston: Houghton Mifflin.

Fortus, D., Dershimet, R. C., Krajcik, J., Marx, R. W., & Mamlok-Naaman, R. (2004). Design-based science and student learning. *Journal of Research in Science Teaching, 41*(10), 1081–1110.

Gee, J. P. (2005). Learning by design: good video games as learning machines. *E-Learning, 2*(1), 5–16.

Habgood, M. P. J., Ainsworth, S. E., & Benford, S. (2005). Endogenous fantasy and learning in digital games. *Simulation & Gaming, 36*(4), 483–498.

Harel, I. (1991). *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school*. Norwood, NJ: Ablex Publishing.

Hayes, E. R., & Games, I. A. (2008). Making computer games and design thinking a review of current software and strategies. *Games and Culture, 3*(3–4), 309–332.

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

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

Kafai, Y., Franke, M., Ching, C., & Shih, J. (1998). Game design as an interactive learning environment for fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematical Learning, 3*(2), 149–184.

Kafai, Y. B., & Peppler, K. A. (2012). *21 developing gaming fluencies with scratch: Realizing game design as an artistic process*. In *Games, learning, and society: Learning and meaning in the digital age* (Vol. 355).

Ke, F. (2008). Computer games application within alternative classroom goal structures: cognitive, metacognitive, and affective evaluation. *Educational Technology Research and Development, 56*(5–6), 539–556.

Kelleher, C., Pausch, R., & Kiesler, S. (2007, April). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1455–1464). ACM.

Kolodner, J. L. (1993). *Case-based reasoning*. San Mateo, CA: Kaufmann.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: putting learning by design into practice. *Journal of the Learning Sciences, 12*(4), 495–547, 1532-7809.

Langer, E. J. (1997). *The power of mindful learning*. Reading, MA: Addison-Wesley.

Leng, E. L., Ali, W., & Baki, R. (2010). Computer games development experience and appreciative learning approach for creative process enhancement. *Computers & Education, 55*(3), 1131–1144.

Li, Q. (2010). Digital game building: learning in a participatory culture. *Educational Research, 52*(4), 427–443.

Li, Q. (2012). Understanding enactivism: a study of affordances and constraints of engaging practicing teachers as digital game designers. *Educational Technology Research & Development, 60*, 785–806.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Beverly Hills, CA: Sage Publications.

Marshall, M., & Rossman, G.,B. (2006). *Designing qualitative research*. Thousands Oaks: Sage Publication.

Mason, J., & Davis, B. (2013). The importance of teachers' mathematical awareness for in-the-moment pedagogy. *Canadian Journal of Science, Mathematics and Technology Education, 13*(2), 182–197.

Miles, M., & Hurberman, M. (1994). *Qualitative data analysis: An expanded sourcebook*. London: Beverley Hills.

Mitchell, J., Kelleher, H., & Saundry, C. (2007). A multimedia mathematics project in a teacher education program. In L. Farr Darling, et al. (Eds.), *Collective improvisation in a teacher education community* (pp. 101–118). Springer Science Business Media.

Muller, M. J., & Kuhn, S. (1993). Participatory design. *Communications of the ACM, 36*(6), 24–28.

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

Piaget, J. (1957). *Construction of reality in the child*. London: Routledge & Kegan Paul.

Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology, 82*(1), 33–40.

Powell, A. B., Francisco, J. M., & Maher, C. A. (2003). An analytical model for studying the development of learners' mathematical ideas and reasoning using videotape data. *The Journal of Mathematical Behavior, 22*(4), 405–435.

Robertson, J., & Good, J. (2005). Story creation in virtual game worlds. *Communications of the ACM, 48*(1), 61–65.

Robertson, J., & Howells, C. (2008). Computer game design: opportunities for successful learning. *Computers & Education, 50*(2), 559–578.

Roblyer, M. D., & Edwards, J. (2000). *Integrating educational technology into teaching*. Upper Saddle River, N.J.: Merrill.

Schoenfeld, A. H. (1992). Learning to think mathematically: problem solving, metacognition, and sense-making in mathematics. In D. Grouws (Ed.), *Handbook for research on mathematics teaching and learning* (pp. 334–370). New York: MacMillan.

Shaffer, D. W. (2005). Studio mathematics: the epistemology and practice of design pedagogy as a model for mathematics learning. In *Wisconsin Center for education research working paper*. No. 2005-3.

Shank, R. C. (1999). *Dynamic memory revisited*. New York: Cambridge University Press.

Smeets, E. (2005). Does ICT contribute to powerful learning environments in primary education? *Computers & Education, 44*(3), 343–355.

Sotamaa, O. (2005). Creative user-centred design practices: lessons from game cultures. In Haddon, et al. (Eds.), *Everyday Innovators: Researching the role of users in shaping ICTs* (pp. 104–116). London: Springer Verlag.

Stake, R. (1995). *The art of case research*. Thousand Oaks, CA: Sage Publications.

Sternberg, R. J. (2006). The nature of creativity. *Creativity Research Journal, 18*(1), 87–98.

Tapia, M., & Marsh, G. E. (2004). An instrument to measurement mathematics attitudes. *Academic Exchange Quarterly, 8*(2). Retrieved August 15, 2004, from http://www.rapidintellect.com/AEQweb/cho25344l.htm.

VanLehn, K., Jones, R. M., & Chi, M. T. H. (1992). A model of the self-explanation effect. *Journal of Learning Sciences, 2*(1), 1–59, 1532-7809.

Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education, 56*(1), 127–137.
de Vries, E. (2006). Students' construction of external representations in design-based learning situations. *Learning and Instruction, 16*, 213–227.
Vygotsky, L. (1978). The role of play in development. In T. M. Cole (Ed.), *Mind in society* (pp. 92–104). Cambridge, MA: Harvard University Press.
Wing, J. M. (2011). Computational thinking: what and why. *The Link Newsletter*. Retrieved from http://www.cs.cmu.edu/ourcs/presentations/ct.pdf.
Wong, R. M. F., Lawson, M. J., & Keeves, J. (2002). The effects of self-explanation training on students' problem solving in high-school mathematics. *Learning and Instruction, 12*(2), 233–262.
Yin, R. K. (2008). *Case study research: Design and methods*. Thousand Oaks, CA: Sage Publications.