Received: 26 February 2018 | Revised: 24 April 2018 | Accepted: 13 May 2018
DOI: 10.1111/jcal.12281

ORIGINAL ARTICLE

WILEY Journal of Computer Assisted Learning

Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences

Hatice Yildiz Durak

Faculty of Education, Department of Computer Education and Instructional Technology, Bartin University, Turkey

Correspondence
Asst. Prof., Hatice Yıldız Durak, Faculty of Education, Department of Computer Education and Instructional Technology, Bartin University, Bartin, Turkey.
Email: hatyil05@gmail.com

Abstract

The aim of this research is to determine the effects and experiences of the use of digital story design activities in teaching applications of programming on academic achievement, participation, and programming self-efficacy. In the study, which is designed through the mixed method, quasi-experimental design is used in the quantitative dimension. The study group of the research consists of 62 fifth-grade students in a secondary school. During the 10-week application process, the experimental group was engaged in digital story design activities in the programming learning process, and the learning process of the control group was carried out without any extra activity. As a result of the research, it is found that students' level of learning of programming concepts, programming self-efficacy, and level of participation in teaching process change significantly and more positively depending on experimental process. In addition, various suggestions for application and research are presented in the research.

KEYWORDS

digital story design, participation, programming self-efficacy, Scratch, secondary school students, teaching programming concepts

1 | INTRODUCTION

Programming education in K–12 is a topic that gets more and more important day by day. It is possible to see the reflection of this importance in the policies of many countries (Durak, 2016). Because programming education is necessary for the acquisition and development of qualifications of the 21st century (Fessakis, Gouli, & Mavroudi, 2013; Kalelioğlu & Gülbahar, 2014). Programming processes include different thinking skills and knowledge areas (Einhorn, 2011; Fessakis et al., 2013; Grover & Pea, 2013; Ismail, Ngah, & Umar, 2010; Kalelioğlu & Gülbahar, 2014; Lau & Yuen, 2011; Yen, Wu, & Lin, 2012). As a matter of fact, according to the understanding of education framework of the 21st century, it is expected that students should be able to form a specific product, present the product, express themselves creatively, and present their knowledge in a productive manner (Kırmacı & Yıldırım, 2017). Hence, it can be concluded that the programming skill has turned into the basic skill of the 21st century (Durak, 2016).

Sarpong, Arthur, and Amoako (2013) stated that computer programming is an art that involves the ability of one to analyse problems and realize solutions. For this reason, it is important to determine and develop the levels of programming skills of students. The most important variables that need to be addressed in order to examine the levels of programming skill of students are programming achievement and programming self-efficacy, and these variables provide important clues to the development of programming skills (Davidsson, Larzon, & Ljunggren, 2010; Tsai, Wang, & Hsu, 2018). Self-efficacy affects the choice of activities, the level of effort spent, resistance, and performance in coping with difficulties in order to solve a problem and to accomplish a task (Bandura, 1977). It is widely accepted that self-efficacy is related to general academic learning achievements and performance of learning specific skills (Bergey, Ketelhut, Liang, Natarajan, & Karakus, 2015; Girasoli & Hannafin, 2008). On the other hand, strong relationships between self-efficacy and digital learning/programming have also been extensively studied (Chiu & Tsai, 2014; Davidsson et al., 2010; Komarraju & Nadler, 2013; Ramalingam, LaBelle, & Wiedenbeck, 2004; Richardson, Abraham, & Bond, 2012; Tsai, Chuang, Liang, & Tsai, 2011). Moos and Azevedo (2009) stated that students' computer self-efficacy is related to learning performances in computer-based learning environments.

Hongwarittorrn and Krairit (2010) emphasized that a low level of programming self-efficacy is a barrier to programming education. Ramalingam et al. (2004) examined whether experience and self-efficacy have an impact on programming performance. In this study, it was found that the programming performance is related to the experiences of the individuals and that these variables significantly predicted the programming performance.

Considering that there is a strong relationship between self-efficacy and learning performance, in order to evaluate the learning levels of students in the study, computer programming self-efficacy and achievement in the lectures are considered important and useful. On the other hand, it is not enough to know only programming codes when preparing problem solutions in programming processes. Active participation should be provided in order to learn the programming processes (Altun & Mazman, 2012). Because programming is a complex process involving many skills and experiences (Lee & Chen, 2009). Bloom (1984) stated that student participation is the interaction of the learner with the presented items of learning and continuity of this interaction until the desired behaviour is achieved. The level of participation of learners in the learning–teaching process is one of the best indicators of the quality of teaching activity (Fredricks, Blumenfeld, & Paris, 2004). Therefore, there is a need to examine the active participation in the programming process to ensure the individual's continuity to deal with high-level problems, show patience to achieve the goal, and demonstrate the desire to try new solutions.

However, programming is a difficult subject for students to perceive and learn (Askar & Davenport, 2009; Baser, 2013; Bergin & Reilly, 2005; Byrne & Lyons, 2001; Durak, 2016; Esteves & Mendes, 2004; Hongwarittorrn & Krairit, 2010; Kelleher & Pausch, 2007). This perceived difficulty in programming was shown to lead to a decrease in the achievement and self-efficacy of the students and in participation to the programming activities. (Altun & Mazman, 2012). According to Byrne and Lyons (2001), the main reason for the difficulties in programming teaching is traditional programming teaching methods. In programming education with the traditional approach, programming concepts are handled separately and holistic view on the process could not be provided (Esteves & Mendes, 2004). This situation increases the learning difficulties of the students and the teaching difficulties of the teachers (Denner, Werner, & Ortiz, 2012; Robins, Rountree, & Rountree, 2003). In order to find solutions to these problems, it is necessary to examine how programming instruction can be done and what kinds of activities and teaching concepts of programming can be positive for the students.

McDermott, Eccleston, and Brindley (2008) stated that the digital story approach is well suited for solving the challenges of programming teaching. Similarly, there are studies in the literature that emphasized that digital stories facilitate the learning of students (Fields, Kafai, Strommer, Wolf, & Seiner, 2014; Gyabak & Godina, 2011; Karaoglan Yilmaz & Durak, 2018; Kocaman-Karoglu, 2016; Verdugo & Belmonte, 2007; Yang & Wu, 2012). In the present study, programming education was realized with digital story development approach, and it was aimed to overcome the difficulties in programming teaching by examining the methodological effect. It is also thought that this study will give a different perspective to the literature as it observes the effects of digital story development activities through the use of a programming teaching tool (Scratch) during the process of teaching programming concepts.

1.1 | The aim of the research

The aim of this research is to examine the fact that whether the use of digital story design activities in teaching applications of programming for secondary school 5th-grade students has influence on learning of programming concepts, programming self-efficacy, and programming participation.

In line with this objective, answers are needed to find for questions below.

- What is the level of students' academic achievement, programming self-efficacy, and participation before and after programming teaching through digital storytelling and direct instruction method?
- Is there a meaningful difference between the contribution level of programming teaching application through digital storytelling and direct instruction method to students' level of participation, academic achievement, and programming self-efficacy?
  - The following hypotheses have been tested for this problem.
    - There is a significant difference between the achievement scores of final test in favour of the experimental group when the self-efficacy pretest scores of the students in the experimental group in which the teaching of programming with the digital story is carried out and the scores of the students of the control group in which the teaching with straight expression is performed are checked.
    - There is a significant difference between the participation scores of final test in favour of the experimental group when the self-efficacy pretest scores of the students in the experimental group in which the teaching of programming with the digital story is carried out and the scores of the students of the control group in which the teaching with straight expression is performed are checked.
    - There is a significant difference between the self-efficacy scores of final test in favour of the experimental group when the self-efficacy pretest scores of the students in the experimental group in which the teaching of programming with

the digital story is carried out and the scores of the students of the control group in which the teaching with straight expression is performed are checked.

- What are the experiences and opinions of the students regarding digital story design and programming teaching process?

2 | CONCEPTUAL FRAMEWORK

2.1 | Why digital story

Today, one of the most important challenges in the learning-teaching process is creating meaningful and engaging learning experiences and providing learners' participation in this process (Sarıtepeci, 2016). Learning through experience during the education process is the most efficient way of ensuring and transferring the permanence of learning. In the literature, digital story seems to be based on active learning theory based on constructivist approach (Dupain & Maguire, 2007; Yang & Wu, 2012). According to the active learning theory, participation is considered as a prerequisite for learning (Fredricks et al., 2004; Guo, Kim & Rubin, 2014).

In the literature it was stated that digital story helps students to understand complex learning content by creating interest, participation, and motivation, facilitating student collaboration and organization of ideas (Robin, 2006, 2008; Van Gils, 2005). Process of digital story creation consists of four stages: preproduction; production; postproduction; and distribution (Robin, 2006). Process of digital story creation seems to be a process based on constructivist learning theory (Yang & Wu, 2012).

In process of digital story creation, a systematic teaching process is followed and this process is integrated with digital technologies by supporting creative thinking and writing skills. Process of digital story creation is an effective teaching approach for creating a creative and open-ended learning environment for teachers and students (Yang & Wu, 2012). In addition, digital storytelling is a useful educational tool that combines innovative teaching and learning practices with digital media (Dakich, 2008) and is an effective method to support active learning processes that can be used by teachers from primary to higher education (Bromberg, Techatassanasoontorn, & Andrade, 2013; Xu, Park, & Baek, 2011).

The student will actively participate in the process of digital story creation. Participation of the students is an inevitable necessity especially in the learning process of the digital story which is created by the student. In this process, learners will have the opportunity to configure and reflect their own learning about content units through the digital story that they have prepared. In addition, the digital story process requires the learner to follow the steps of project creation within the context of a content unit. The use of digital stories in the programming process, which is a complex process and involves many different skills, will allow many different commands and features to be used together. On the other hand, there are many studies that showed that the digital story development approach improves creative thinking ability (Ohler, 2013; Yang & Wu, 2012). Creative thinking is also an important skill for programming. In this context, it is expected that the digital story development approach will contribute to improving the programming skills of the students.

2.2 | Digital story use in programming teaching

In addition to traditional story design, digital story design consists of stages involving multimedia elements such as sound, music, animation, video into a designed story, and the vocalization of the story (Porter, 2005; Robin, 2006; Rule, 2010). Digital story design is a new learning approach for learning that is enriched by multimedia elements used in learning–teaching processes (Robin, 2006).

Providing participation and motivation in programming learning processes is one of the most important problems (Kim, Song, & Lee, 2009); therefore, it is important to present interesting and different activities to provide students' active participation (Jakoš & Verber, 2017). According to Law, Lee, and Yu (2010), active participation and programming learning are related with each other, and it is emphasized that teaching for gaining programming skills will be impossible if learner does not attend the class sufficiently. The main purpose to use digital story is to facilitate the transition of learning into an active and interesting process (Bromberg et al., 2013). Digital stories can be used differently in education (the story scenario can be prepared by the teacher, or the student can create his own digital story; Robin, 2008). If the student creates the digital story in the process of digital story design, the program will provide active participation in the process of developing the program by following the steps of program development according to the content pattern determined during the learning process. In addition, the student creates a multimedia product using the appropriate technological tools and materials in the framework of the story s/he designs and ultimately realizes technology integration. Taking these reasons into consideration, digital storytelling can be an appropriate approach in terms of programming teaching. On the other hand, digital story design activities play an important role in the acquisition and development of some 21st century skills such as creativity, critical thinking, problem solving, and communication for students (Price, Strodtman, Brough, Lonn, & Luo, 2015). It also supports many creative skills that are tried to contribute to students such as research, collecting and organizing information, creating new ideas, designing, writing, expressing themselves, putting out products, solving problems are supported in programming teaching (Csizmar Dalal, 2009; Dupain & Maguire, 2007; Hathorn, 2005; Sarıca & Usluel, 2016; Yang & Wu, 2012). At this point, it can be predicted that digital storytelling will increase the programming achievement (Park, 2014).

Burke and Kafai (2012) investigate how the coding process of computer programs can help children in developing storytelling and their creative writing skills. In the relative study, it is emphasized that the process of the creation of program coding is not only related with the knowledge in computer science, but also it contributes to develop creative storytelling and writing skills. According to Nam (2017), digital story can be defined as an audio-visual approach that engages students in learning motivation and the development of problem-solving skills, to keep their visual and audial senses in such a way that printed textbooks can never realize. Burke (2012) carried out a study that develops story design with secondary school students to improve their coding and writing skills by using Scratch program. In the study, it is found that students effectively use the basic programming concepts (i.e., sequence, iteration, condition, event handling, and parallel execution) to design digital stories. Besides, students develop positive attitude towards programming after teaching. Although there are studies that emphasize that storytelling can facilitate the coding learning in the literature, excluding the studies that search how the use of digital storytelling can lead affective reactions related with academic achievement and programming, there are no examples.

2.3 | Scratch as a tool for digital story design

Scratch, a program aims to teach programming, is a programming environment designed by Lifelong Kindergarten at the Massachusetts Institute of Technology (MIT) Media Laboratory specifically for 8–16 age group of individuals. Scratch (scratch. mit.edu) is a visual programming tool that allows K–12 students to create stories, games, animations, and interactive presentations. Scratch program allows flow of events by dragging–dropping the code blocks and combining them (Grover & Pea, 2013; Resnick et al., 2009). There are various characters, scenes, and sounds in the library of this program, and this program has an editor that allows them to be edited.

Like Scratch program, programming environments that hold the visual foreground have facilitated programming teaching (Shin & Park, 2014). Due to the drag-and-drop structure of the Scratch code blocks and their similarity to the puzzle pieces, it seems to fit the theoretical basis of the constructivist approach (Buckleitner, 2007). On the other hand, Scratch has the ability to combine various media types (pictures, sounds, videos, etc.) to help develop 21st century skills (Romero, 2010). Scratch, which enables to design projects with various multimedia elements, can be used in several lessons, apart from algorithm and programming teaching.

Scratch's prominent visual features and its basic interface are seen as internal motivation for students, and it enables to develop logical thinking (Calder, 2010). In the studies, it is seen that students develop a positive point of view regarding programming while using Scratch, and their experiences are positive (Fesakis & Serafeim, 2009; Smith & Neumann, 2014). Thus, Scratch is thought as an appropriate educational tool for digital story design in programming teaching.

3 | METHOD

3.1 | Research design

This research is designed with mixed methods. It was designed according to Creswell's (2012) mixed research strategies, "explanatory sequential mixed methods design." Creswell demonstrated that using quantitative and qualitative methods together would provide a better understanding of research problems than using two approaches separately. For the purpose of the research, quantitative research methods were used first. The quantitative data collection tools were applied to 62 students. Qualitative research methods were used for a more detailed examination in the direction of the analysis of the quantitative data obtained. Within this scope, 32 students in the experimental group were interviewed. In the quantitative dimension, quasi-experimental research design was employed with pretest–posttest control groups. The experimental design used in the research is presented in Table 1.

3.2 | Study group

The research is conducted with 62 participants studying at 5th grade in Ankara. Within the scope of the research, secondary school 5th-grade students who have no experience in programming and take the course named as Information Technologies and Software are preferred. The students are assigned to experimental and control groups in a neutral manner.

Thirty-two students who formed the study group are in the experimental group, and 30 students are in the control group. Approximately 41.9% of the study group are female, and 58.1% are male; 34.4% of the experimental group comprised of women, and 65.6% of them are male. Fifty per cent of control group is female, and 50% is male.

3.3 | Process

The implementation of the research has lasted for 10 weeks. During the implementation process, digital storytelling and related materials were used for learning process in the experimental group, whereas direct instruction was used in the control group. At the end of the process, volunteer students were interviewed.

Application was made in the course of "programming technology and software" where programming is taught. The scope of the course aims to teach the programming concepts effectively. In this context, digital story designing activities were carried out in the experimental group whereas straight expression was used in control group for the teaching of programming concepts.

In the experimental application, stages of digital story designing determined by Robin (2008) were used. First, students began to research to determine the topics that are based on their experiences, lives, and interests. Most of the students have chosen subjects related to science, social and language, and, stories, tales that have social messages.

After the preliminary work in order to produce an effective story, students were asked to write their stories. After writing the story, the

TABLE 1 Experimental design

Group	Assignment	Used tools/operations	Pretest	Posttest
G_1	U	X_A	$O_{1.1}$ $O_{1.2}$ $O_{1.3}$	$O_{2.1}$ $O_{2.2}$ $O_{2.3}$
G_2	U	X_S	$O_{1.1}$ $O_{1.2}$ $O_{1.3}$	$O_{2.1}$ $O_{2.2}$ $O_{2.3}$

G_1 = first experimental group; G_2 = first control group; U = convenience sampling; X_A = teaching programming with digital storytelling; X_S = teaching programming with direct instruction; $O_{1.1}$ = Programming Self-Efficacy Scale for Secondary School Students—Pretest; $O_{1.2}$ = Participation Scale—Pretest; $O_{1.3}$ = academic achievement test for programming—pretest; $O_{2.1:}$ = Programming Self-Efficacy Scale for Secondary School Students—Posttest; $O_{2.2}$ = Participation Scale—Posttest; $O_{2.3}$ = academic achievement test for programming—posttest.

stage of storyboarding was started. Storyboards were built on paper with various desktop applications or online tools (see Figures 1 and 2).

Using the storyboard as a guide, images, sounds, and animations were created. At this stage, the students explored the characters on the internet or drew their own. At this stage, the written story texts and the dialogues in this story were vocalized by the students. At this stage, the students used the sound recorder and the Scratch software. At this stage, students have combined multimedia elements such as text, sound, and visuals they have prepared in the Scratch program according to the storyboards. Scratch commands were used to control the events in the scenario, after the multimedia elements that would be included in each scene were added. Many events were realized with programming commands, such as adding visuals or video transition effects, adjusting transition and sound timings, and adding background music, audio narration, and background music volume.

Finally, students shared their digital stories (see Figures 1 and 2) that were prepared with Scratch with classmates in class and others in an online environment. It has been seen that the generated stories were mostly directed towards the subjects related to lectures.

3.4 | Data collection tools

3.4.1 | Personal Information Form

Personal Information Form are developed by researchers. Through this four-item data collection tool, data related with personal information of participants are collected. Survey items differ according to the questions and are generally Likert type. During the development of this form, two field experts were consulted and the number of questions was reduced from 7 to 4 by taking into account the recommendation of experts to reduce the number of questions.

3.4.2 | Participation Scale

The Participation Scale used in the research developed by Sarıtepeci (2012) within the scope of master's thesis. In this scale, the items of the online scale, National Survey of Student Participation, were adjusted to the secondary school level, and the previous studies on student participation were examined to form items. Validity and reliability studies were carried out during the development process of the scale. Expert opinions were taken in this context. In the explanatory factor analysis made for examining the validity of the scale, three dimensions emerged. In the confirmatory factor analysis, it was determined that the conformity of the model based on the relationships between the three structures was at a good level. The Cronbach's alpha coefficient of Participation Scale was 0.93 in the original scale, and the subscales ranged from 0.80 to 0.89. When the process of this scale was examined, it was seen that validity and reliability were provided. There are many scales related to the participation in the field. The reason for choosing this scale was that it is both a validated and tested scale, and it was developed with the aim of determining participation in a course where technology integration at secondary level was done. So the purpose of the scale and the content was very

Title: Story of the truth cannot be hidden

Scenario:

Main character: King

Supporting Characters: Loam, bad guys

Story (introduction): (external sound)

Today I will tell you the story of the saying, "The truth cannot be hidden" I know you all are curious. I'm starting my story right now.

One day, there is a hard-working and well-behaved king. He always works for his country. Those who dislike the king are constantly slandering him. The king on this situation

King: I work for my country. Those who slander me will regret one day.

The men who slander: We must blame him constantly to cover up what the king does.

Story (Conclusion):(external sound)

As the time goes by, the king has made the effort and the development of the country increased. Hence, the king ...

King: The truth cannot be hidden, so the good works are like sunshine. It could not be hidden with evil

(external sound)

This story tells us how the expression "The truth cannot be hidden" emerges. This story also tells us that the good things can not be hidden and one day they will surely emerge.

Project Screenshot

Storyboard

FIGURE 1 Examples from digital stories developed by students—1 [Colour figure can be viewed at wileyonlinelibrary.com]

Title: Elements are discussing

Scenario:

Main character: Oxygen

Supporting characters: Hydrogen, Carbon, Nitrogen

Story (introduction): (external sound) One day hydrogen, oxygen, carbon, and nitrogen meet. The elements asks after each other and suddenly start to argue. First oxygen begins to speak in a tone of arrogance

Oxygen: I am the most found element in nature. I am the most precious among you.

(external sound) The carbon that gets annoyed and talks

Carbon: Oxygen what are you saying? I am building the whole life in the world. I am in the structure of all living beings. The diamond is my allotrope.

(external sound) Nitrogen talks.

Nitrogen: I am the gas that covers 78% of the air. The plants can not do without me.

Oxygen: How do you forget that we form many compounds together?

Hydrogen: I think you've forgotten me. I am the most abundant element in the environment, even if i am the lightest one. I constitute about 90% of the universe in terms of weight.

(external sound) The discussion gets bigger and bigger. Oxygen has noticed that all of them are important for life. Now it has no arrogance. It understands its fault and tries to calm the environment.

Oxygen: Let's calm down, my companion elements. I apologize for my arrogant behavior. After listening to the conversation I understand that we are all valuable. For example, how will water (H2O) carbon dioxide gas (CO2) be formed if we do not exist? For this reason, we are all necessary for life, we are very important for nature.

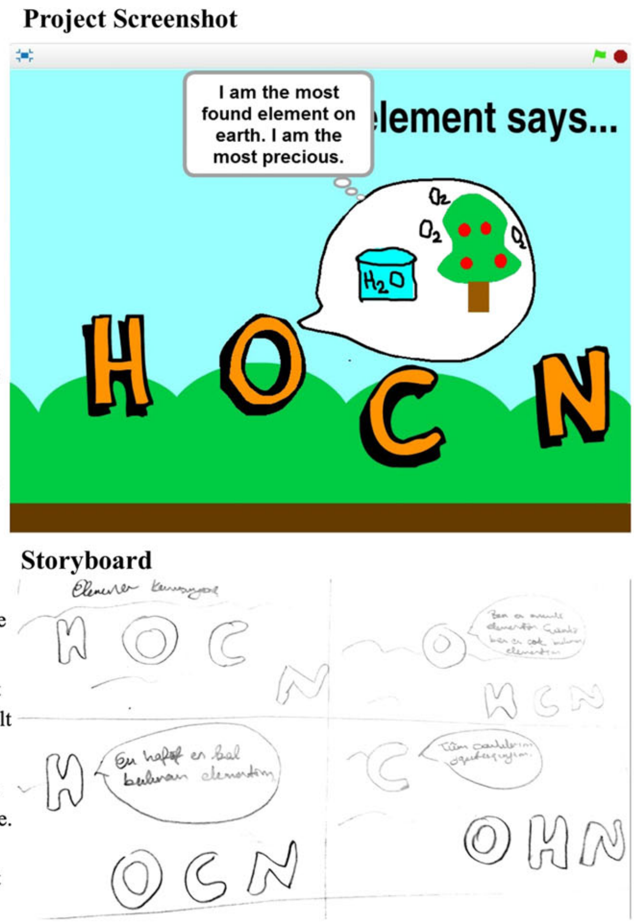

FIGURE 2 Examples from digital stories developed by students—2 [Colour figure can be viewed at wileyonlinelibrary.com]

suitable for the purpose of the present study. This scale consisting of 34 items and three factors is of 5-scale Likert type. The grading is between *Never* (1) and *Always* (5). In this scale consisting of three factors, "active and assisted learning" subdimension has 17 items, "student's level of difficulty" has 7 items, and "feedback level and teacher-student interaction" subdimension has 10 items. The lowest score that can be taken from this scale is 34, and the highest score is 170. The high score on the scale is interpreted as the fact that the active participation of the student is high in the class.

In the research, Cronbach alpha reliability coefficient for the scale is 0.980. Cronbach alpha reliability coefficients for subdimensions are 0.968; 0.921, 0.945, respectively. The values obtained exceed the acceptable level of 0.70 (Nunnally, 1978). On the other hand, validity and reliability were tested with 351 students before the scale was used in the study, and the reliability coefficient was found to be 0.93. For all the items in the scale, item-total correlations change between 0.54 and 0.68, and t values are significant (p < .001). These results were interpreted as the fact that the validity of the items was high and items measure the same behaviour. Reliability and validity of the scale were checked by the researchers to be used in the scope of the study.

3.4.3 | Programming Self-Efficacy Scale

This scale was developed by Kukul, Gökçearslan, and Günbatar (2017). This scale consists of 31 items and one factor. This scale aims to programming self-efficacy level of students during teaching programs such as Scratch, Logo, and Alice, which are used to help students gain programming skill by teachers and researchers. The scale was selected because it was appropriate for the context of the study and provided validity and reliability and also as the Scratch software was used for programming. On the other hand, this scale was the only Turkish one developed to determine the programming self-efficacy at the secondary school level. When the development process of this scale is examined, it is seen that validity and reliability studies have been done. The item loads of the items for measurement vary between 0.47 and 0.71. Cronbach's alpha coefficient is 0.95, and the two half method results were 0.96, indicating that the internal consistency of the scale is high. A one-dimensional structure was obtained by exploratory and confirmatory factor analyses that were made to determine the validity of the scale structure.

This scale is of 5-point Likert scale type, and there is no reverse item in the scale. Taking a high score on the scale indicates that the programming self-efficacy is high. According to 5-point Likert scale type, the grading is between *Strongly Disagree* (1) and *Strongly Agree*. In this research, the evaluated Cronbach alpha reliability coefficient is at high level (0.975).

On the other hand, validity and reliability were tested with 351 students before the scale was used in the study, and the reliability coefficient was found to be 0.93. For all the items in the scale, item-total correlations change between 0.46 and 0.70, and t values are significant ($p < .001$). These results were interpreted as the fact that the validity of the items was high and items measure the same behaviour. Reliability and validity of the scale were checked by the researchers to be used in the scope of the study.

3.4.4 | Academic achievement test

Academic achievement test is developed by researchers in order to determine the level of participants' learning programming concepts. While developing achievement test, table of specifications are prepared according to the content to be given in the implementation process first. Headings partaking in the content are “the basic concepts of programming (variables, constraints, flow control blocks/conditional elements, operators), data representation, Parallelism, Synchronization, User interactivity, Flow control, interface elements of Scratch.” In the process of constituting achievement exam, a question pool that consists of 50 questions measuring attitude at every level in the table of specification is formed. In addition, while the question pool is created, questions at three different levels of difficulty for analysis, synthesis, and evaluation that state complex behaviours are constituted. The prepared achievement test was applied to 351 middle school students who had taken the programming course in the pilot study. Item difficulty and item discrimination values, and test questions out of the limit values have been removed. The reliability of the Kuder–Richardson 20 of achievement test was calculated, and the reliability coefficient was found to be 0.79. This value exceeds the recommended threshold (Nunnally, 1978). The difficulty index of the items in the test range from .26 to .85. The discrimination index of the items in the tests are between .24 and .54.By choosing 10 questions from question pool, achievement test was formed and the duration of exam that was evaluated out of 100 was determined. According to the feedbacks from the specialists and students' scores, item complication levels were evaluated, and necessary changes were made.

3.4.5 | Semistructured interview form

In this data collection tool developed by researchers, participants were asked six questions under the theme of “the effect of process on skill development, interaction, the view on programming process, liked and disliked sides of the process, students' views on their experience regarding the process.” To ensure the validity and reliability, the data collection tool was formed by consulting two domain experts in the process of developing the data collection tool.

3.5 | Data analysis

One-way analysis of covariance (ANCOVA) and descriptive statistics were used in the analysis of quantitative data. In the analysis of qualitative data, content analysis method was used. Data collected through interview forms were examined by themes and codes. To ensure reliability among coders, 10% of the data were coded by two coders. The coding is made by two coders to ensure reliability of the research, and consistency was calculated as 90.2% among coders. In calculating this ratio, Miles and Huberman's (1994) “reliability = (number of consensus)/(total consensus+total divergence) formula was used. Two coders reached a compromise for differential codes.

Academic achievement, participation, and programming self-efficacy scores of the experimental and control groups were examined through quantitative methods.

ANCOVA was used to compare academic achievement pretest–posttest scores of the groups. Hypotheses of this test were controlled before analysis. Findings related to pretest scores were examined and found no significant difference between pretest-related groups. ANCOVA was used to reduce the error variance and to remove the differences between the groups at the beginning of the practice (Mertler & Vannatta, 2002), although there is no difference between the groups in terms of pretest scores.

In order to apply ANCOVA, it has been examined whether the assumptions of this analysis were met.

In this context, firstly average scores were compared and samples were found to be unrelated. The skewness and kurtosis values of the dependent variable were examined, and the distribution was found to be normal. For each of the other assumptions of the ANCOVA, the homogeneity of the variance scores of the dependent variable was checked by the Levene test. In this technique, variance was considered to be homogeneous if p value is greater than .05 ($p > .05$; Field, 2005). Following this assumption, the Group × pretest interaction in the Custom Model for the ANCOVA was checked under the assumption of equality of regression coefficients (slopes) within the groups, and this assumption was also found to be ensured. Finally, the linear relationship between the dependent variable and the covariate was tested using the Pearson correlation analysis technique. It is effective when the relationship between the dependent variable and the covariate is greater than 0.30 (Field, 2005), and there is a relationship of $r = .68$ between the variables.

4 | FINDINGS

Findings related with pretest–posttest scores of experimental and control groups are presented in Table 2.

When Table 2 is examined, it is found that the scores of students' academic achievement is found as 40.33 in the group that is implemented direct instruction before the experiment, then 73.17 is found after the experiment. However, in the experimental group, academic achievement scores are found as 40.16 before experiment and 81.56 after experiment.

When Table 3 is examined, the mean scores of the students in the experimental group that is equal to 2.49 are 3.24. The participation score of the control group is 2.81, whereas the posttest score is

TABLE 2 Descriptive findings on students' academic achievement pretest–posttest scores

		Pretest		Posttest	
Groups	*N*	Mean	*SD*	Mean	*SD*
G1	32	40.16	17.34	81.56	15.68
G2	30	40.33	21.65	73.17	13.42

TABLE 3 Levels of students lecture participation and programming self-efficacy

		Pretest		Posttest	
Scale	Groups	Mean	*SD*	Mean	*SD*
Participation	G1	2.49	1.22	3.24	.91
	G2	2.81	1.27	3.07	1.17
Programming self-efficacy	G1	2.65	1.21	3.42	.91
	G2	2.86	1.32	3.09	1.09

3.07. The programming self-efficacy is 2.65 before the experiment and 3.24 after the experiment in the experiment group. The programming self-efficacy is 2.86 before the experiment in the control group and 3.09 after the experiment.

Thus, when digital story design applications with Scratch in programming teaching are compared with direct instruction method, it can be said that group that designs digital story affects the level of learning programming concepts, participation level, and programming self-efficacy more positively than the control group.

The results of the ANCOVA on whether there is a significant difference between the posttest scores when the groups' academic achievement test pretest scores are controlled are presented in Table 4.

As shown in Table 4, there is a statistically significant difference between the posttest academic achievement scores corrected according to pretest scores of the groups, $F(1–59) = 4.97$, $p < .05$. In this context, it can be said that the academic achievement of the students who participated in the study changes significantly according to the experimental process.

As shown in Table 5, there is a statistically significant difference between posttest participation scores corrected according to pretest scores of the groups, $F(1–59) = 43.74$, $p < .05$. In this context, it can be said that the participation level of the students participating in the study changed significantly according to the experimental process.

As shown in Table 6, there is a statistically significant difference between posttest programming self-efficacy scores corrected according to pretest scores of the groups, $F(1–59) = 29.19$, $p < .05$. In this context, it can be said that the programming self-efficacy levels of the students who participated in the study did not change significantly depending on the experimental process.

4.1 | Qualitative findings regarding students' thoughts

Theme that views about digital story design activities in programing teaching for students, codes placed under themes, and frequency of coding of these codes are presented in Tables 7–10.

When student opinions in Table 7 are examined, it is understood that students view digital story design activities in programming teaching as a process that improves many skills. When the codes are examined, it is seen that students mention about research (f = 8) and questioning (f = 8) skills while explaining the contributions of digital story design activities. Afterwards, technical skills (f = 6), problem-solving (f = 5), reflective thinking (f = 5), algorithmic thinking (f = 5), designing skills (f = 3), and creativity skills (f = 3) are used in this explanation, respectively. Here are some examples of students' views:

> *I have done constant research to design digital stories in the class. I scanned my books to find a subject. I looked*

TABLE 4 Analysis of covariance results of students' academic achievement posttest scores adjusted according to students' pretest scores by methods

Groups	Mean	Adj. mean	Variance source	Sum of squares	*df*	Mean Square	*F*	Sig.	Partial Eta sq.
G1	81.56	80.5	Pretest	999.14	1	999.14	4.97	.03	.078
G2	73.17	72.2	Method	1,101.07	1	1,101.07	5.48	.02	.085
			Error	11,846.90	59	200.79			
			Total	386,325.00	62				

TABLE 5 Analysis of covariance results of students' participation posttest scores adjusted according to students' pretest scores by methods

Groups	Mean	Adj. mean	Variance source	Sum of squares	*df*	Mean square	*F*	Sig.	Partial Eta sq.
G1	3.24	3.10	Pretest	27.56	1	27.56	43.74	.00	.43
G2	3.07	3.00	Method	1.75	1	1.75	2.78	.10	.05
			Error	37.17	59	.63			
			Total	686.08	62				

TABLE 6 Analysis of covariance results of students' programming self-efficacy posttest scores adjusted according to students' pretest scores by methods

Groups	Mean	Adj. mean	Variance source	Sum of squares	*df*	Mean square	*F*	Sig.	Partial Eta sq.
G1	3.42	3.35	Pretest	29.19	1	29.19	54.88	.00	.48
			Method	3.08	1	3.08		.02	.09
G2	3.09	3.01	Error	31.31	59	.53	5.81		
			Total	722.55	62				

TABLE 7 Learners' opinions on the contribution of the digital story designing activities used in programming education to the development of their skills

Theme	Code	f
The skills that digital story designing activities contribute	research	8
	questioning	8
	technical skills/technology literacy	6
	problem-solving	5
	reflective thinking	5
	algorithmic thinking	5
	designing skills	3
	creativity skills (individualization, writing, designing)	3

TABLE 8 Learners' opinions about their interactions in the process of digital story design

Theme	Code	f
Direction of the learners' interactions in digital story design process	student-student	15
	student-teacher	10
	student-content	5

TABLE 9 The learners' opinions on the process of designing a digital story in programming education (liked and disliked aspects)

Theme	Code	f
Positive	A Fun/Enjoyable/Striking Method	12
	Multimedia item content (character, scene, sound, music, script, animation/motion, etc.)	11
	Scratch program	6
	Facilitating the learning of programming concepts	5
	Motivation to learn programming	5
	Suitability to personal preferences/ Individualization possibility	5
	Use of digital media	5
	Maintaining exercise and activity together	3
	Developing positive attitude towards programming	3
	Increasing the satisfaction level	3
	Providing active cognitive participation and encouragement	3
	Providing permanent learning	3
	Producing a product and transforming what is imagined into a product	3
Negative	Inability to find requested images	6
	Difficulty of digitizing the designed stories	5
	Difficulties due to Scratch's features	5
	Unable to switch scenes in the encoding process	5
	The lack of time in school	5
	No computer and internet access outside of the school	5
	Preparing an activity is time consuming	5
	Idea theft	1

at my books to write a scenario. I used internet to find characters. [S2]

While developing a digital story, both scenario and application had to be implemented. On the other hand, I had to search the Internet and find out the character that I wanted and organize the ones that I couldn't find. I learned the computer better. [S4]

When the student opinions in Table 8 are examined, it is seen that the activities performed in the process of digital story designing will mostly contribute to student-student interaction. Later, it is thought to be beneficial for student-teacher interaction (f = 10) and student-content interaction (f = 5), respectively. Here are some examples from student views:

I did not know how to use a computer or write code. I applied constantly to create a digital story. [S1]

When I was designing a digital story, I mostly got the help from my friends. I've learned better from them. [S7]

When the student opinions in Table 9 are examined, it is stated that the use of digital story in programming lesson is the most remarkable, fun, and different way of learning. Later on, it is thought that they favoured and contributed positively to the accommodation of multimedia media (f = 11) and the features of Scratch program (f = 6), respectively. When examining the other subthemes, the main benefits of the student's favourite aspects are thought to be facilitating the learning of programming concepts (f = 5), increasing motivation for programming learning (f = 5), personalization possibilities (f = 5), using digital environment (f = 5), coexistence of practice and activity (f = 3), developing positive attitudes for programming and increasing satisfaction (f = 3), positive cognitive participation and encouraging effort (f = 3), providing permanent learning (f = 3), and product development and turning imagination to production (f = 3). When the views of students on the process are examined, it is stated that not using the appropriate pictures and scenes in digital story is not welcomed by the students. Here are some examples from students' views:

The biggest challenge I had was finding the pictures I wanted while composing the story I imagined. [S9]

The part I did not like was that I could not make a scene in Scratch while creating a digital story. Scratch would have been better if it was in a structure that made the stage pass. [S20]

When the student opinions in Table 10 are examined, it is found that the most frequently used textbooks were used to create digital stories (f = 11), scripts were inspired from cartoons (f = 8), Scratch library is mostly used for character selecting (f = 8), the process of creating digital stories allows many commands in Scratch to be applied (f = 8), and it is observed that the fact that Scratch has many visualizations is liked by the students. Here are some examples from student views:

I first used the Turkish textbook to determine the subject. I searched the internet to find the characters. But mostly I used characters and scenes in Scratch. [S18].

I really liked Scratch. Especially the visual. There are many beautiful characters and scenes. [S22].

TABLE 10 The learners' highlights and experiences of digital story designing process

Theme	Code	f
Subject identification	Identifying the subject by using the textbooks	11
	Identifying the subject using the Internet	9
	Identifying the subject with help of friends	8
	Identifying the subject by using other books	2
Scenario writing	Writing a scenario based on cartoons	8
	Writing a scenario to teach the subject	5
	Creating a scenario that I dream about	2
	Writing scenario according to Scratch features	2
Selecting and specifying characters/ scene design	Using and customizing the characters in the Scratch library	10
	Favorite cartoon characters	5
	Top characters in novel/fairy tale/stories	4
	Any character that is easy to find in search engines	3
	Creating your own character	2
Creation of digital stories	Use/application of various commands	8
	Arrangement of algorithms (arranging code blocks in an algorithmic order to carry out the designed script correctly)	5
	Switching between scenes and providing parallelism for multiple events (e.g., “do the announcement … make, when the green flag is clicked, when … is pressed)	5
	Use of repeat (continuous, “repeat …”) commands for repeated events	5
	Providing data entry with speech bubble, volume adjustment, coordinates and keyboard for dynamic interaction	3
	Performing error checks	3
	Using Variables/Constants	
	Use of conditional constructs to control events in the scenario	3
	Recording/Adding audio	3
	Using Boolean logic (… and …, … or …, … not)	1
	Using Scratch editing options for user interface design	1
Experiences in the Scratch program while preparing digital story	Visuality is in the foreground	8
	Immediate display of code results	6
	Codes are already available (in blocks)	6
	Characters have different costumes	4
	The examples used are in different languages.	2

5 | RESULTS, DISCUSSION, AND SUGGESTIONS

5.1 | Results

In this study, the effects of using digital story design activities in programming instructional practices on academic achievement, participation, programming self-efficacy, and student experience are examined. The level of students' learning of programming concepts in programming instruction process, programming self-efficacy, and participation levels are found to be significant and more positive. When the literature is examined (Price et al., 2015), it is seen that the digital story plays an important role in the acquisition and development of some 21st century skills such as creativity, critical thinking, problem solving, and communication, which will support this finding. Csizmar Dalal (2009) reported that many creative skills such as researching, gathering information, organizing information, creating new ideas, designing, writing, expressing themselves, putting out products, and solving problems are being supported by digital storytelling activities. Therefore, the finding that programming activities will be effective with digital story activities (Park, 2014) coincides with the findings obtained in this study.

In the study, it is seen that students mostly describe the contributions of digital story designing activities as research and inquiry skills in general. This is followed by technical skills, problem solving, and reflective thinking skills. In general, when student opinions are examined, it appears that digital story development activities are related to the development of other thinking skills. When the literature is examined (Price et al., 2015), it is emphasized that digital story design activities seem to be important for students to acquire and develop some 21st century skills such as creativity, critical thinking, problem solving, and communication. Again, the description made by International Society for Technology in Education (2015) defines information processing thinking as being handled with creative thinking, algorithmic thinking, critical thinking, and problem-solving skills. These skills are also very important skills for teaching programming. From here, it can be said that the digital story development process and its influences provide good opportunities for students to learn to program and make programming teaching enjoyable.

The students claim that the student-student interaction mostly takes place in the process. It is remarkable that the student-content interaction is at the last place. This finding suggests that students prefer most of their peers in unrealized topics and prefer to learn from their peers. The existence of a positive relationship between learning and interactivity in the literature is often expressed (Swan, 2001; Valasek, 2001). According to Wagner (1998), interaction is important in the realization of active learning. Interaction, on the other hand, allows learners to share ideas, to assess their performance with their own observations, new knowledge to be more meaningful, and increased interest and motivation to learn and reflect on their learning. In the work done by Durak (2016), students are first contacted with friends and teachers in challenging tasks related to scheduling, and they prefer course contents at the very last. In the research conducted by Kim et al. (2009), it is observed that positive experiences about programming, positive attitude towards the course, process contribution and expectations of appreciation and motivation levels are affected in primary school students' interaction of programming education. It can be said that many factors such as internal and external motivational attitudes are influential in the process of student interaction.

Students have emphasized that the process is mostly enjoyable and entertaining; however, the most unpopular is the process of finding objects to use in the digital story. From this point of view, the emphasis is placed on the need for different, visual, multimedia learning environments in the students' opinions. In support of these findings, Durak (2016) notes that Scratch's visual orientation is a good motivating factor for students in programming instruction. What students do not like in the process seems to be a reflection of the difficulties that students have had in learning programming and transferring it.

According to the student experience of the digital story development process, it was noticed that the subjects are mostly drawn by using textbooks, the scenarios are generally inspired from cartoons, the Scratch library is mostly used in the selection of characters, the process of creating digital stories allows the execution of many scripts in Scratch, and that Scratch is attracted by the students due to its visualization feature. From these findings, it can be said that the digital story development process gives opportunity to apply the basic elements of programming and to use it in all aspects of the Scratch program. In the research conducted by Sarıtepeci (2016) and Kim et al. (2009), it was observed that primary school students' positive experiences about environment; their interaction with the environment; the positive attitude of the environment; and the contribution, appreciation, and motivation levels were increased in programming education, and this increased the programming achievement. Similarly, Wang, Huang, and Hwang (2014) investigated the effects of the learning environment on the students' learning achievements in the 7th-grade students and found that they increased the attitude and motivation towards the curriculum based on the game and animation preparation in the direction of the students' personal preferences.

5.2 | Suggestions and limitations

Within the scope of this study, the effects of digital story development activities in programming teaching are examined. In the study, it is found that students are challenged in the digital story development process. The findings of this work provide conclusions in order to develop a specific instructional design for instructional material designers or trainers and to learn digital storytelling programming by presenting an application example from the digital story development and teaching material design process.

The study has certain limitations regarding the study of the teaching process, digital story designing skills, and products of the students. In digital storytelling-based programming learning environments, it is recommended to develop online teaching materials such as checklists or rubrics to examine the teaching activities of teachers and the learning activities of students. Additionally, it is important for teachers to develop tutorial strategies for different topics of computer science by effectively using digital storytelling and to develop instructional resources to facilitate the social presence and interactions of students. During the application phase of this study, students had some difficulties while creating the digital story. In order to overcome these difficulties and facilitate the process, teachers can give students instructions to assist them in fulfilling their role as screenwriter, technician, character picker, and narrator in a faithful manner.

One of the limitations of this study was the small sample size, which limits the statistical power of the findings of this study. Further research should be done to explore the deep relationship between digital storytelling backed student achievement, and programming self-efficacy and experience.

ORCID

Hatice Yildiz Durak http://orcid.org/0000-0002-5689-1805

REFERENCES

Altun, A., & Mazman, S. G. (2012). Developing computer programming self-efficacy scale. *Journal of Measurement and Evaluation in Education and Psychology*, *3*(2), 297–308.

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *TOJET: The Turkish Online Journal of Educational Technology*, *8*(1), 26–32.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191–215. https://doi.org/10.1037/0033-295X.84.2.191

Baser, N. (2013). Asian Research Consortium. *Asian Journal of Research in Banking and Finance*, *3*(10), 28–40.

Bergey, B. W., Ketelhut, D. J., Liang, S., Natarajan, U., & Karakus, M. (2015). Scientific inquiry self-efficacy and computer game self-efficacy as predictors and outcomes of middle school boys' and girls' performance in a science assessment in a virtual environment. *Journal of Science Education and Technology*, *24*(5), 696–708. https://doi.org/10.1007/s10956-015-9558-4

Bergin, S., & Reilly, R. (2005). Programming: Factors that influence success. *In ACM SIGCSE Bulletin*, *37*(1), 411–415. ACM

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4–16.

Bromberg, N. R., Techatassanasoontorn, A. A., & Andrade, A. D. (2013). Engaging students: Digital storytelling in information systems learning. *Pacific Asia Journal of the Association for Information Systems*, *5*(1), 1–22.

Buckleitner, W. (2007). Why "Scratch" is significant. *Children's Technology Review*, *15*(6), 17.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, *4*(2), 121–135.

Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers: Digital storytelling with scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 433–438). ACM.

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *In ACM SIGCSE Bulletin ACM*, *33*(3), 49–52. ACM

Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *APMC*, *15*(4), 9–14.

Chiu, Y. L., & Tsai, C. C. (2014). The roles of social factor and internet self-efficacy in nurses' web-based continuing learning. *Nurse Education Today*, *34*(3), 446–450. https://doi.org/10.1016/j.nedt.2013.04.013

Creswell, J. W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (4th ed.). Boston: Pearson.

Csizmar Dalal, A. (2009). Digital Storytelling as a Gateway to Computer Science. *Journal of the Research Center for Educational Technology*, *4*(2), 124–137.

Dakich, E. (2008). Towards the social practice of digital pedagogies. In N. Yelland, G. Neal, & E. Dakich (Eds.), *Rethinking education with ICT: New directions for effective practices* (pp. 13–30). Rotterdam and Taipei: Sense Publishers.

Davidsson, K., Larzon, L. & Ljunggren, K. (2010). *Self-efficacy in programming among STS students*. Retrieved from http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, *58*(1), 240–249.

Dupain, M., & Maguire, L. L. (2007). Health digital storytelling projects. *American Journal of Health Education*, *38*(1), 41–43.

Durak, H. (2016). Design and development of an instructional program for teaching programming process to gifted students. *Unpublished doctoral dissertation*, Gazi University, Ankara.

Einhorn, S. (2011). Microworlds, computational Thinking, and 21st century Learning Logo Computer System Inc, White Paper. Retrieved from http://www.microworlds.com/

Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics. In *Frontiers in Education, 2004. FIE 2004. 34th Annual*, F4C-7. IEEE.

Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. In *ACM SIGCSE Bulletin* 41, 3, 258–262. ACM.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97.

Field, A. (2005). *Discovering statistics using SPSS* (2nd ed.). London: Sage.

Fields, D. A., Kafai, Y. B., Strommer, A., Wolf, E., & Seiner, B. (2014). Interactive storytelling for promoting creative expression in media and coding in youth online collabortives in Scratch. Proceedings of Constructionism, pp. 19–23.

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School participation: Potential of the concept, state of the evidence. *Review of Educational Research*, *74*(1), 59–109.

Girasoli, A. J., & Hannafin, R. D. (2008). Using asynchronous AV communication tools to increase academic self-efficacy. *Computers & Education*, *51*(4), 1676–1682. https://doi.org/10.1016/j.compedu.2008.04.005

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Gyabak, K., & Godina, H. (2011). Digital storytelling in Bhutan: A qualitative examination of new media tools used to bridge the digital divide in a rural community school. *Computers & Education*, *57*(4), 2236–2243.

Hathorn, P. P. (2005). Using digital storytelling as a literacy tool for the inner city middle school youth. *The Charter Schools Resource Journal*, *1*(1), 32–38.

Hongwarittorrn, N., & Krairit, D. (2010). Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming. In *The spring 8th International conference on Computing, Communication and Control Technologies*, pp. 6-9.

International Society for Technology in Education (2015). *CT Leadership toolkit*. Retrieved from http://www.iste.org/docs/ct-documents/ctleadershipt-toolkit.pdf?sfvrsn=4

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. *TOJET: The Turkish Online Journal of Educational Technology*, *9*(2), 125–131.

Jakoš, F., & Verber, D. (2017). Learning basic programing skills with educational games: A case of primary schools in Slovenia. *Journal of Educational Computing Research*, *55*(5), 673–698.

Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, *13*(1), 33–50.

Karaoglan Yilmaz, F. G., & Durak, H. (2018). Examining pre-service teachers' opinions about digital story design. *Education and Information Technologies*, *23*(3), 1277–1295.

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, *50*(7), 58–64.

Kim, K. Y., Song, J. B., & Lee, T. W. (2009). Effect of digital storytelling based programming education on motivation and achievement of students in elementary school. *Journal of the Korea Society of Computer and Information*, *14*(1), 47–55.

Kırmacı, Ö., & Yıldırım, E. (2017). Digital storytelling with using Scracth. *11th International Computer & Instructional Technologies Symposium (ICITS-2017)*. May 24–26, 2017, İnönü university, Malatya, Turkey.

Kocaman-Karoglu, A. (2016). Personal voices in higher education: A digital storytelling experience for pre-service teachers. *Education and Information Technologies*, *21*(5), 1153–1168.

Komarraju, M., & Nadler, D. (2013). Self-efficacy and academic achievement: Why do implicit beliefs, goals, and effort regulation matter? *Learning and Individual Differences*, *25*, 67–72. https://doi.org/10.1016/j.lindif.2013.01.005

Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Journal of Educational Technology Theory and Practice*, *7*(1), 158–179.

Lau, W. W., & Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, *57*(1), 1202–1213.

Law, K. M., Lee, V. C., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, *55*(1), 218–228.

Lee, C. Y., & Chen, M. P. (2009). A computer game as a context for non-routine mathematical problem solving: The effects of type of question prompt and level of prior knowledge. *Computers & Education*, *52*(3), 530–542. https://doi.org/10.1016/j.compedu. 2008.10.008

McDermott, R., Eccleston, G., & Brindley, G. (2008). More than a good story—Can you really teach programming through storytelling? *Innovation in Teaching and Learning in Information and Computer Sciences*, *7*(1), 34–43.

Mertler, C. A., & Vannatta, R. A. (2002). Advanced and multivariate statistical methods: Practical application and interpretation. Los Angeles: Pyrczak. In *Predictors of accommodations use for deaf students*.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis*. Thousand Oak: CA: Sage.

Moos, D. C., & Azevedo, R. (2009). Learning with computer-based learning environments: A literature review of computer self-efficacy. *Review of Educational Research*, *79*(2), 576–600. https://doi.org/10.3102/0034654308326083

Nam, C. W. (2017). The effects of digital storytelling on student achievement, social presence, and attitude in online collaborative learning environments. *Interactive Learning Environments*, *25*(3), 412–427.

Nunnally, J. C. (1978). *Psychometric theory*. New York: McGraw Hill.

Ohler, J. B. (2013). *Digital storytelling in the classroom: New media pathways to literacy, learning, and creativity*. Corwin Press.

Park, J. H. (2014). A study on digital storytelling based programming education. *Journal of the Korea Society of Computer and Information*, *19*(5), 119–128.

Porter, B. (2005). "The art of digital storytelling". Retrieved from http://digitalstorytelling.wiki.usfca.edu/file/view/ArtOfStorytelling.pdf/523804710/ArtOfStorytelling.pdf on 06.02.2017.

Price, D. M., Strodtman, L., Brough, E., Lonn, S., & Luo, A. (2015). Digital storytelling: An innovative technological approach to nursing education. *Nurse Educator*, *40*(2), 66–70.

Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *In ACM SIGCSE Bulletin*, *36*(3), 171–175.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60–67.

Richardson, M., Abraham, C., & Bond, R. (2012). Psychological correlates of university students' academic performance: A systematic review and meta-analysis. *Psychological Bulletin*, *138*(2), 353–387.

Robin, B. (2006). The educational uses of digital storytelling. *In Society for Information Technology & teacher education international conference*, *2006*(1), 709–716.

Robin, B. R. (2008). Digital storytelling: A powerful technology tool for the 21st century classroom. *Theory Into Practice*, *47*(3), 220–228.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137–172.

Romero, J. S. (2010). Library Programming with LEGO MINDSTORMS, Scratch, and PicoCricket: Analysis of best practices for public libraries. *Computers in Libraries*, *30*(1), 16–19.

Rule, L. (2010). Digital storytelling: Never has storytelling been so easy or so powerful. *Knowledge Quest*, *38*(4), 56–57.

Sarıca, H. Ç., & Usluel, Y. K. (2016). The effect of digital storytelling on visual memory and writing skills. *Computers & Education*, *94*, 298–309.

Sarıtepeci, M. (2012). *The effect of blended learning environment on students' participation, academic achivement, attitude and motivation to the course of social studies in seventh grade in primary education. Unpublished MA thesis, Gazi University*, *Ankara*.

Sarıtepeci, M. (2016). *Investigation of the effectiveness of digital storytelling method in social studies course*, PhD Thesis, Gazi University, Ankara.

Sarpong, K. A. M., Arthur, J. K., & Amoako, P. Y. O. (2013). Causes of failure of students in computer programming courses: The teacher-learner perspective. *International Journal of Computer Applications*, *77*(12), 27–32.

Shin, S. I., & Park, K. (2014). A physical interactive game for learning English of children. In *Future information technology* (pp. 219–224). Berlin, Heidelberg: Springer.

Smith, C. P., & Neumann, M. D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, *21*(3), 185–188.

Swan, K. (2001). Virtual interaction: Design factors affecting student satisfaction and perceived learning in asynchronous online courses. *Distance Education*, *22*(2), 306–331.

Tsai, C. C., Chuang, S. C., Liang, J. C., & Tsai, M. J. (2011). Self-efficacy in Internet-based learning environments: A literature review. *Journal of Educational Technology & Society*, *14*(4), 222–240.

Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2018). Developing the Computer Programming Self-Efficacy Scale for Computer Literacy Education. *Journal of Educational Computing Research*. https://doi.org/10.1177/0735633117746747

Valasek, T. (2001). *Student persistence in Web-based courses: Identifying a profile for success*. Branchburg, NJ: Raritan Valley Community College, Center for the Advancement of Innovative Teaching and Learning. (ERIC Document Reproduction Service No. ED466276)

Van Gils, F. (2005). Potential applications of digital storytelling in education. In *3rd twente student conference on IT* (Vol. 7). University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science Enschede.

Verdugo, D. R., & Belmonte, I. A. (2007). Using digital stories to improve listening comprehension with Spanish young learners of English. Retrieved from http://scholarspace.manoa.hawaii.edu/bitstream/10125/44090/1/11_01_ramirez.pdf

Wagner, E. D. (1998). Interaction strategies for online training designs. In Distance Learning '98. Proceedings of the Annual Conference on Distance Teaching & Learning (14th), Madison.

Wang, H. Y., Huang, I., & Hwang, G. J. (2014). Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses. In *Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on*, pp. 382–387. IEEE.

Xu, Y., Park, H., & Baek, Y. (2011). A new approach toward digital storytelling: An activity focused on writing self-efficacy in a virtual learning environment. *Educational Technology & Society*, *14*(4), 181–191.

Yang, Y. T. C., & Wu, W. C. I. (2012). Digital storytelling for enhancing student academic achievement, critical thinking, and learning motivation: A year-long experimental study. *Computers & Education*, *59*(2), 339–352.

Yen, C. Z., Wu, P.-H., & Lin, C.-F. (2012). Analysis of expert's and novice's thinking process. *Engaging Learners through Emerging Technologies*, *Communication in Computer and Information Science*, *302*, 122–134.

How to cite this article: Yildiz Durak H. Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *J Comput Assist Learn*. 2018;1–13. https://doi.org/10.1111/jcal.12281