Ilkogretim Online - Elementary Education Online, 2020; 19 (3): pp. 1820-1837
http://ilkogretim-online.org.tr
doi:10.17051/ilkonline.2020.735175

A framework proposal for the design of video-assisted online learning environments for programming teaching*

Mithat Elçiçek, *Siirt University,* Turkey, *mithatelcicek@gmail.com* ORCID: 0000-0003-1845-7271
Hasan Karal, *Trabzon University,* Turkey, *karalhasan@gmail.com* ORCID: 0000-0002-3555-050X

Abstract: It is observed that in the wake of developments in digital technologies, the role of shareholders in the learning-teaching process changed and a transition from the classical face-to-face communication into online learning communication occurred. This transition also led to a change the knowledge and behaviours expected from indivudals. In this context, they are expected to acquire several skills considered as the skills of the 21st century, among which are critical and analytical thinking. Recent studies suggest that programming teaching is influential on these skills. However, programming teaching refers to a process which is structuraly difficult and complex. Thus, we need novel methods and techniques regarding programming teaching. Considering that particulalrly online learning environments have begun to be a crucial part of the learning-teaching process recently, it is believed that assisting onlne learning environments with learning videos in which audiovisual symbol systems are used together may play a crucial role in developing the programming skill that is abstract and unclear intuitively. From this point of view, this study aims to propose a framework related to desgning video-assisted online learning environments for programming teaching. To achieve this goal, the design-based research has been conducted. The research has been conducted through the participation of 48 (F: 27, M: 21) registered undergraduate students and of 4 instructors (F: 1, M: 3) of the Department of Computer Education and Instructional Technology based in the Faculty of Education at a state university located in the east of Turkey. Observation forms, reflective student diaries, the focus group interview, interview, and graded scoring key were used as data collection tools. As a result of the research, a framework has been proposed in the context of "content", "visual design", "interaction" and "practicability" related to the design of video-assisted online learning environments for "cognitive", "affective", "methodological" and "environmental" problems. Suggestions were presented within the framework of the results obtained from the study and the experiences of the researcher.

Keywords: programming teaching, video-assisted online learning, the design-based research

Received: 14.11.2019	Accepted: 10.03.2020	Published: 15.06.2020

INTRODUCTION

With the developments in today's information and communication technology, a transformation is in question where learning-teaching activities are transferred into electronic platforms and digital technologies are used in transferring information and skills. This development called as "Digital Transformation" has increased interest in online learning acvtivities and caused this transformation to gain a fast momentum recently (Hrastinski, 2009; Singh and Hurley, 2017). In this context, online learning environments have become an important actor in the learning-teaching process. In particular, it is observed that higher education students prefer online learning activities, participate in online learning activities and that these activities are among the most benefited applications (Allen & Seaman, 2011). Online learning environments consist of forms such as text, image, sound, video, animation, and simulation. However, video content attracts more than other content (Bayazıt & Akçapınar, 2018 Brame, 2016; Delen, Liew & Willson, 2014). There are many factors for video content to get more attention than other content. Among these factors are the use of audiovisual symbol systems together, motivating them by attracting the attention of learners, presenting real-life examples, facilitating the

* This study was produced from the doctoral thesis prepared by the first author under the supervision of the second author.

teaching of hard-to-explain knowledge and skills (Means, Toyama, Murphy, Bakia & Jones, 2009; Palloff & Pratt, 2007).

Parallel to the developments in information and communication technology, the knowledge and skills expected from individuals also change. In this context, with the critical and analytical thinking skills shown among 21st century skills, individuals are expected to produce effective solutions by approaching the events occurring in their environment from a different perspective (Basten, Evers, Geijsel & Vermeulen, 2018; Sayin & Seferoglu, 2016). When reviewing the literature, it is observed that it is essential to develop programming skills to educate new generations equipped with critical and analytical thinking skills (Psycharis & Kallia, 2017; Sheth, Murphy, Ross & Shasha, 2016). Because, programming skills are not only limited to computer sciences, but they are also important for interdisciplinary interactions (Priest & Gass, 2017). Through programming teaching, what is aimed is to meet the need of qualified people in the field of software apart from training productive generations having the problem-solving skill (Buitrago-Flórez et al., 2017). In this context, it is observed that studies on teaching programming have increased both in Turkey and in the world (Abbasi, Kazi & Khowaja, 2017; Demirer & Nurcan, 2016).

Researches on the development of programming skills show that programming skill is a difficult and complex process, and attempts to teach programming are not effective enough although they facilitate some concepts (Al-Tahat et al., 2016; Gulbahar, Kalelioglu & Kert, 2018). Although there are many reasons behind the failure of teaching programming, it is stated that one of the most important reasons is the difference between the human mind and the mode of operation of the computer (Tall, 1993). For, the human mindset is different from the logical thinking structure that the computer needs to understand and perform some tasks (Bosse & Gerosa, 2017; Kitchin, 2017). So, the gap between intuitional thinking form and algoritmical thinking needs to be filled. Filling this gap is regarded as a significant problem in all education levels including high-education. Therefore, there is a need for novel methods and techniques regarding programming teaching (Bentley, 2016; Erdem, 2018). Novel methods and techniques should be developed in line with the interests and tendencies of learners. In this respect, considering that particularly online learning environments have begun to be a crucial part of the learning-teaching process recently, it is believed that assisting onlne learning environments with learning videos in which audiovisual symbol systems are used together may play a crucial role in developing the programming skill that is abstract and unclear intuitively. Video-assisted online learning environments are considered as a more comprehensive learning approach that encompasses advantages provided by video-based learning environment (Gabriele, Holthaus & Boulet, 2016). The purpose of the video-assisted learning environments is not only to realize learning based on videos but also to make videos an important part of the learning process (Hajhashemi, Caltabiano, Anderson &Tabibzadeh, 2018). In this context, video-assisted learning environments provide real-life examples to learners and provide learning in the context of life (Choi & Johnson, 2005; Dodson et al., 2018). It supports students' participation in the learning process by providing students with the opportunity to speak and discuss on the subject (Shih, 2010; Zhang, Zhou, Briggs & Nunamaker, 2006). It facilitates the teaching of difficult-to-learn topics or abstract concepts by running different learning styles (Franzoni et al., 2008). It increases individuals' ability to control learning speeds by discovering information in the learning process, supports their cooperative learning, and enables them to take an active role in learning activities and increases their motivation to learn (Bannister & Arbaugh, 2018; Reychav & Wu, 2015).

Therefore, it is important to design video-assisted online learning environments for programming teaching in line with the dynamics of learners and teachers. Following a literature review, it has been determined that there exist very few design studies related to the design of video-assisted online learning environments that have focused on the problems encountered during the programming teaching process. In this context, a framework is needed for the design of video-assisted online learning environments for programming teaching. Thus, the purpose of this research is to propose a design framework on how to design video-assisted online learning environments for programming teaching.

METHODS

Research Model

In the research, firstly qualitative and then both qualitative and quantitative data were used depending on the purpose and need. As a result of the research, a framework has been proposed for the design of video-assisted online learning environments for programming teaching. To achieve this goal, a design-based research was conducted. Design-based research refers to the path, tests conducted, and the improvement process of the product depending on the tests carried out with the collaborative design, analysis and redesign of the researchers and participants in order to reveal a product (environment, educational application, model, principle, theory, etc.) in the research process (Çankaya & Egypt, 2011; Wang & Hsu, 2017). The structure designed in design-based researches is constantly being renewed according to the results of the evaluation. Therefore, design-based research differs from classical design methods in this respect. Again, unlike experimental researches, instead of detecting the differences between the two learning situations, it focuses on the improvements in the process to optimize a learning situation (Baltacı, Yıldız, Kıymaz & Aytekin, 2016). The design-based research phases followed in the study are shown in Figure 1 (Kuzu, Çankaya & Misirli, 2011).

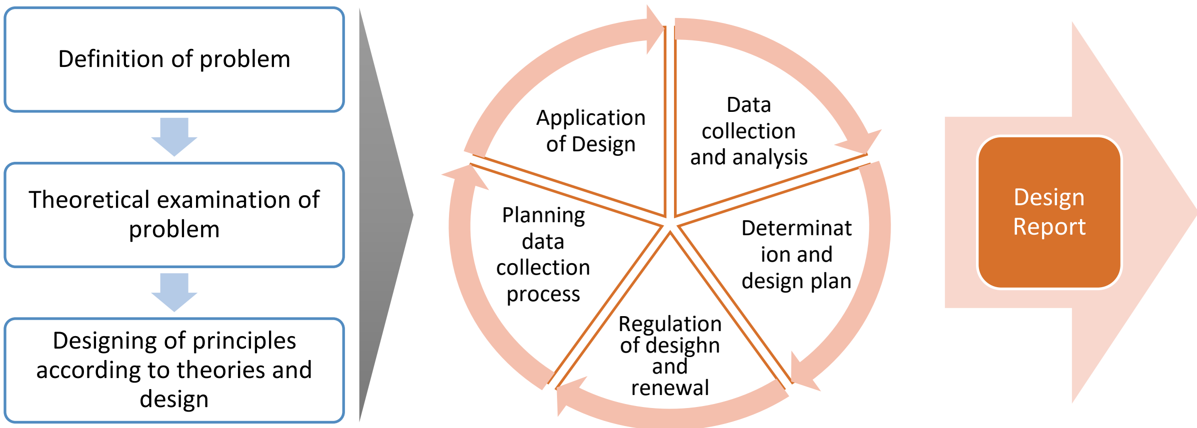

FIGURE 1. *Design-based research phases*

The design-based research followed in the study is comprised of three steps. The preparatory phase includes the definition of problem, theoretical examination of problem, design of principles according to theoris and design; design cycles include the data collection and analysis, determination and design plan, regulation of design, planning data collecttion process and application of design. The final phase includes the the dseign report.

Participants

The research has been conducted through the participation of 48 (F: 27, M: 21) registered undergraduate students and of 4 instructors (F: 1, M: 3) of the Department of Computer Education and Instructional Technology based in the Faculty of Education at a state university located in the east of Turkey. In line with this goal, 6 third-grade students who enrolled programming lessons before, 42 second-grade volunteers who have enrolled programming lessons already and 4 volunteers who have given a lecture about programming have been randomly selected. The aim here is to form a small sample group, allowing the participants to express their learning experiences in the best way and to explain the problem in question through an in-depth analysis (Yıldırım & Şimşek, 2013). Therefore, typical case sampling, which is one of the purposeful sampling methods, was used in the research. The purposeful typical case sampling is to allow for an in-depth study by studying one of the many situations of the research problem along with a typical one (Büyüköztürk et al., 2017).

Research Process

The research was carried out in a classroom environment where a video-assisted online learning environment for programming teaching was used with traditional face-to-face lessons. The research process consists of the preparatory phase, design cyle, and final phase. The design process steps for the preparatory phase are shown in Figure 2.

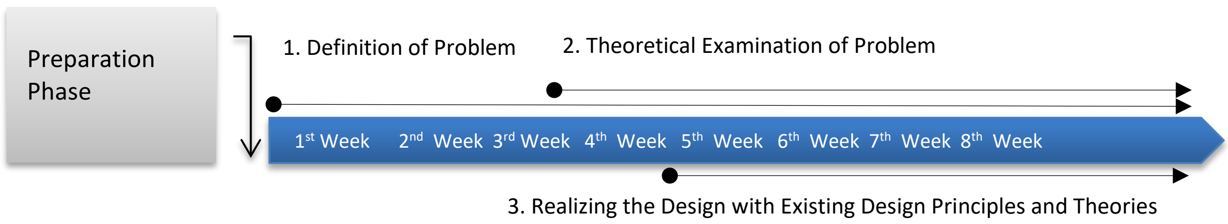

FIGURE 2. *The design process steps in the preparation phase*

The preparation phase consists of the steps of “definition of the problem”, “theoretical examination of the problem”, “realizing the design with existing design principles and theories”. The design cycles of the research consist of "planning the data collection process", "implementation of the design", "data collection and analysis", "decision and design plan", "design correction and renewal". Within the scope of the research, the design cycle was rotated 3 times. The 1st Cycle lasted 6 weeks, the 2nd Cycle 8 weeks, and the 3rd Cycle for 6 weeks. The design process steps for the design cycles of the research are shown in figure 3.

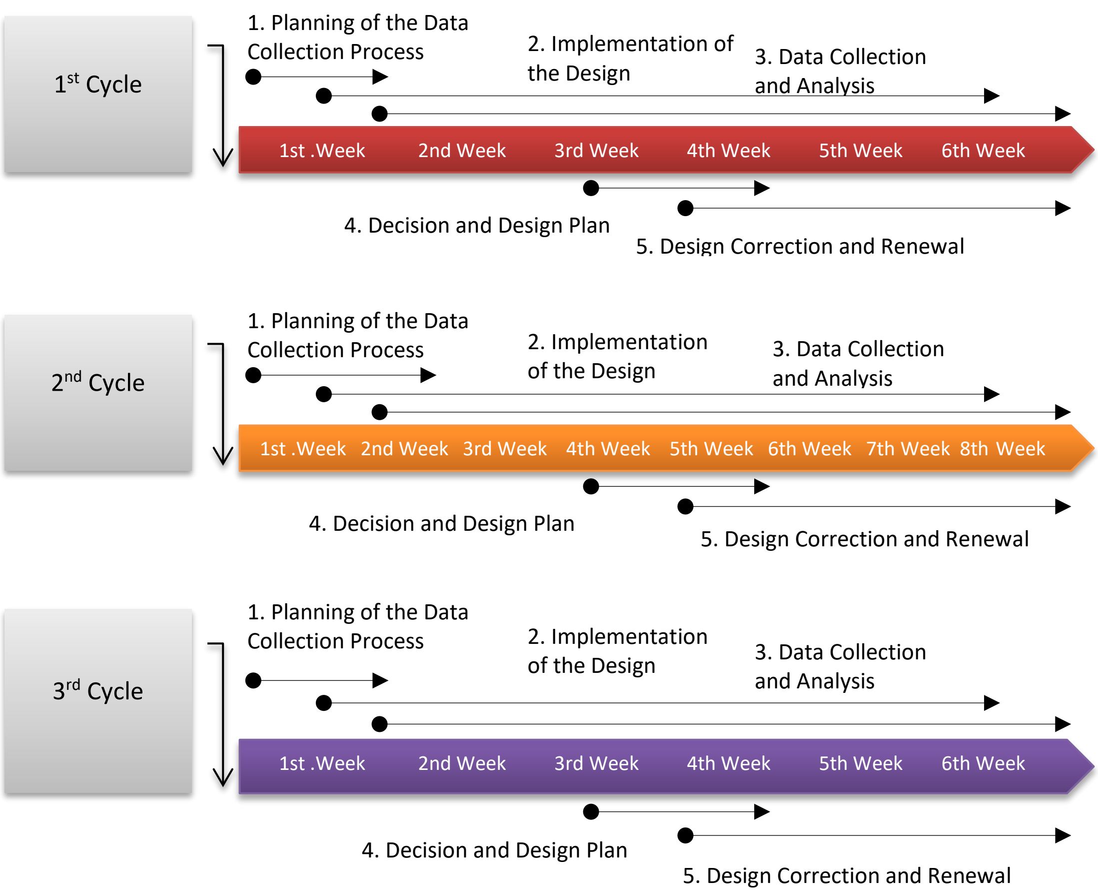

FIGURE 3. *Process steps of design cycles*

The design process of the video-assisted online learning environment for programming teaching was carried out in three cycles. During the preparation phase, the problems encountered during the programming teaching process were revealed. After the problems were examined theoretically, they were turned into problem themes and sub-themes in line with expert opinions. The first design cycle was launched by realizing the first design for problem themes. In the first design cycle, sub-dimensions related to the design of a video-assisted online learning environment for programming teaching were revealed. After examining the sub-dimensions theoretically, they were converted into design variables in line with expert opinions. In this context, design variables and sub-dimensions were converted into a graded scoring key and used as a data collection tool in all of the design cycles (1st Cycle, 2nd Cycle, 3rd Cycle). Revisions were made to the design variables that required correction in each of the design cycles, and the next design cycle was launched. In the third design cycle, no finding that required correction in the axis of design variables was encountered. Therefore, at the end of the third design cycle, the design process of the video-assisted online learning environment for programming teaching was terminated. The design process of the research was conducted regarding Programming Langauges-I and Programming Languages-II, during which the C# programming language is taught. The participants were given access to the video-assisted online learning environment for programming teaching via a web address using the Moodle learning management system. Participants were able to use the video-assisted online learning environment at any time and outside the classroom. Participants were able to use all the components of the video-assisted online learning environment after logging on to the system with their user names and passwords. Sample screenshots belonging to the designed video-assisted online learning environment is given in Figure 4.

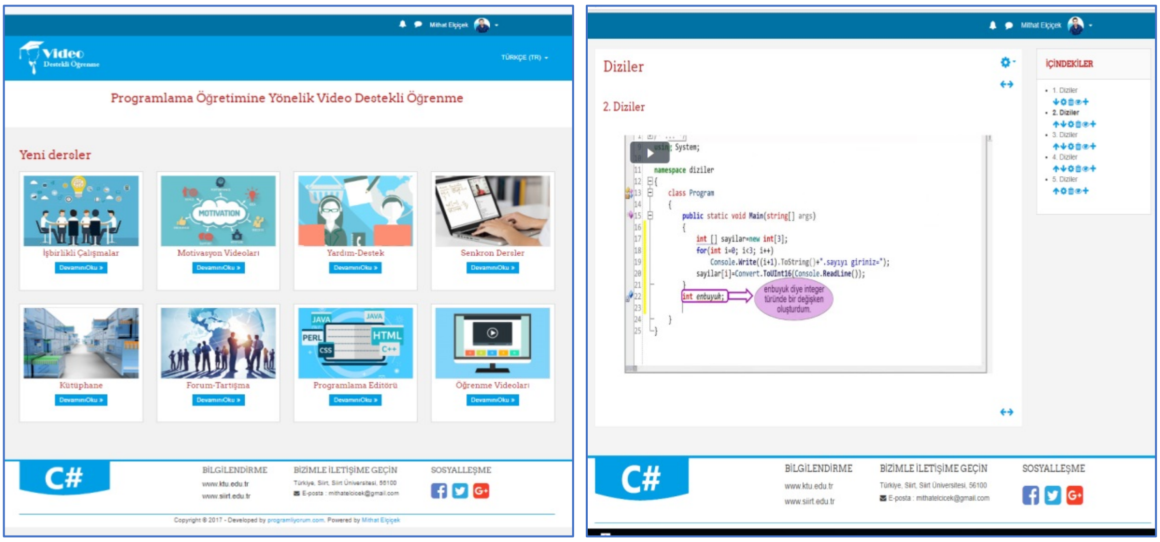

FIGURE 4. *Screenshots belonging to the designed video-assisted online learning environment*

The main page of the video-assisted online learning environment regarding programming teaching includes cooperative works, motivation videos, help-support, syncronized courses, library, forum-discussion, and programming editor and learning videos. 112 videos were used in the video-assisted online learning environment regarding programming teaching. 45 of these videos were generated using Adobe Captivate.

Data Collection Tools

The focus groupinterview, interview, reflective student diaries, graded scoring key, and observation reports were used as data collection tools. Interviews and the focus groupdiscussions were held to identify the participants' experience in designing a video-assisted online learning environment for programming teaching. The interview is a technique of interviewing one or more participants on a specific subject or purpose (Balcı, 2007). A the focus groupinterview is a partially structured flexible group interview technique that is carried out to reveal the information and ideas of the pre-selected participants within the framework of a certain subject (Çokluk, Yılmaz & Oğuz, 2011).

Interviews and the focus groupdiscussions were held face to face in line with pre-determined questions. The order and dimensions of the questions were prepared so that they can be changed during the interview. The level of knowledge that participants had about their own learning and emotions and thoughts they had about a certain topic or situation were collected through reflective student diaries. Within the scope of the research, the design variables and sub-dimensions related to the design of the video-assisted online learning environment for programming teaching were converted into a graded scoring key and used as a data collection tool. It was created by following the phases recommended by Goodrich during the development of the graded scoring key. In the graded scoring key, 3 level ranges, namely "inadequate", "must be developed" and "adequate", were identified, the necessary corrections were made by 4 scoring experts and 2 Turkish language experts and the scoring key consisting of 75 items was reduced to 55 items and its final form was created.

Observations were made to identify the behaviors of the participants in both the classroom and video-assisted online learning environment and to conduct an in-depth study. The responses of the participants to the topics covered in the course, their likes, their participation and the changes in the process were recorded with the observer's notes. The method of the research was explained in detail to ensure the validity of the research. The selection of the studygroup and the participants was explained along with their reasons. Attention was paid to ensure that the entire study group included volunteers. The collection and analysis of the data were described in detail. Data collection tools were examined by field experts before being implemented. The participant observer method was used to make observations during the research. It was noted that the questions posed in the data collection tools included a single judgment. It was examined by Turkish language specialists to clear the questions posed in the data collection tools from language errors. In order to ensure the reliability of the research, the questions in the data collection tools were prepared for the literature and research questions. The research questions were associated with the data collection process. Pilot implementation was carried out before the data collection tools were implemented. The data analysis process was carried out for research questions. Interviews were recorded and data loss was prevented. The distribution of the data collection tools used in the study according to the participants is shown in Table 1.

Table 1. *Distribution of the data collection tools according to participants*

Participants		Interview		The focus Groupinterview		Observation		Reflective Diaries		Graded Scoring Key	
		M	F	M	F	M	F	M	F	M	F
Preparation Phase	S	-	-	5+3*	3+3*	18	24	18	24	-	-
	SM	3	1	-	-	-	-	-	-	-	-
1st Cycle	S	-	-	4	4	18	24	18	24	18	24
	SM	2	1	-	-	-	-	-	-	2	1
2nd Cycle	S	-	-	-	-	-	-	-	-	18	24
	SM	-	-	-	-	-	-	-	-	3	1
3rd Cycle	S	-	-	-	-	-	-	-	-	18	24
	SM	-	-	-	-	-	-	-	-	3	1

S: Student, SM: Staff Member, M: Male, F: Female, *Have Taken Programming Course before

Table 1 demonstrates that 4 instructors were selected for interviews during the preparatory phase and 3 in the first cycle. In the first cycle, 1 instructor did not participate in the interview due to a busy work schedule. In the preparatory phase, 6 and 8 students (14 in total) who took programming lessons previously were selected for the the focus groupinterview while 14 students who took programming lessons in the first cycle were selected for the the focus groupinterview. While students' mid-term examination scores were taken as a basis in the preparatory phase, students' responses to the topics taught, their likes, participation and changes in the process were taken as basis in the first cycle. Participants who could express themselves best were selected from among students with "low", "medium" and "high" scores. A total of 46 participants were selected, including 42 students who took programming lessons for observation and reflective student diaries, and 42 students and 4 lecturers who took programming lessons for the graded scoring key.

Data Analysis

Data obtained from the the focus groupinterviews, interviews, reflective student diaries, and observation reports were analyzed in line with descriptive and content analysis techniques. In this context, reflective diaries, observation reports, and voices recorded during the interviews were then converted into written language and analyzed. While analyzing the content, the reliability coefficient, which is called consensus by Miles and Huberman (1994), was calculated by the formula of $\Delta = C \div (C + \partial) \times 100$ to determine the similarity rate for the same data set by different coders (3 experts). (Δ: Reliability coefficient, C: Number of terms with which consensus is provided, ∂: Number of terms with which there is no consensus). Miles and Huberman (1994) state that consensus should be at least 90%. In all qualitative data analysis, due attention was paid to have a reliability coefficient of 90% and above. Descriptive statistics techniques were used in the analysis of the data obtained from the graded scoring key.

RESULTS

Preparation Phase

As a result of the analysis of the data obtained from the interview, the focus groupinterview, reflective student diaries and observations, 4 themes and a total of 22 sub-themes related to the problems encountered during the programming teaching process were defined. The ratios of frequency and percentage of the participants (student + instructor) regarding the themes and sub-themes are shown below.

Table 2. *The ratios of frequency and percentage of the participants' opinions on cognitive problems*

Theme and sub-themes	Student		Staff Member		Total	
	f	%	f	%	f	%
Cognitive Problem Theme					84	36,5
Syntactic Error	6	12,5	1	25	7	13,5
Conceptual Misconceptions	7	14,6	1	25	8	15,4
Mathematical Thinking	8	16,7	4	100	12	23,1
AlgorithmicThinking	11	22,9	4	100	15	28,8
Successive and Cyclical Thinking	5	10,4	3	75	8	15,4
Pattern Recognition and Pattern Building	3	6,3	1	25	4	7,7
Logical Reasoning	9	18,8	4	100	13	25
Abstract Thinking and Generalisation	8	16,7	4	100	12	23,1
Previous False Learnings	4	8,3	1	25	5	9,6

Table 2 highlights that students and staff members expressed fewer views about cognitive problems in sub-themes of "algorithmic thinking" (f: 15,%: 28.8), "logical reasoning" (f: 13,%: 25), "mathematical thinking" (f: 12) , "abstract thinking and generalization" (f: 12,%: 23.1) than "pattern recognition and building" (f: 4,%: 7,7), "previous learnings" (% 23.1), f: 5,%: 9.6), "syntactic errors" (f: 7,%: 13.5), "conceptual misconceptions" (f: 8, 15.4%), and "sequential and cyclical thinking" (f: 8,%: 15,4). The ratios of frequency and percentage of the participants' opinions on affective problems are shown in Table 3.

Table 3. *The ratios of frequency and percentage of participants ' opinions on affective problems*

Theme and sub-themes	Student		Staff Member		Total	
	f	%	f	%	f	%
The Affective Problem Theme					69	30
Self-Sufficiency	15	15	31,3	3	75	15
Motivation	11	11	22,9	4	100	11
Attitude	13	13	27,1	2	50	13
Anxiety	9	9	18,8	3	75	9
Self-Regulation	8	8	16,7	1	25	8

The sub-themes that the participants expressed opinions most about the affective problems were “self-efficacy” (f: 18, 34.6%), “motivation” (f: 15, 28.8%), “attitude” (f: 15,%: 28.8); whereas, the sub-themes that the participants expressed opinions least about the affective problems were “self-regulation” (f: 9, 17.3%) and “anxiety” (f: 12, 23.1%). The ratios of frequency and percentage of the participants' opinions on methodological problems are shown in Table 4.

Table 4. *The ratios of frequency and percentage of participants' opinions on methodological problems*

Theme and sub-themes	Student		Staff Member		Total	
	f	%	f	%	f	%
Methodological Problem Theme					39	17
Practice and Repetition	7	14,6	4	100	7	14,6
Real-Life Context	9	18,8	3	75	9	18,8
Effective Environment	5	10,4	2	50	5	10,4
Rote Learning	6	12,5	3	75	6	12,5

It is seen that the participants express opinions more on the sub-themes of “real-life context” (f:12, %:23,1), “practice and repetition” (f:11, %:21,2) and less on the sub-themes of effective media” (f:9, %:17,3) and “rote learning” (f:9, %:17,3). The ratios frequency and percentage of participants' opinions on environmental problems are shown in Table 5.

Table 5. *The ratios of frequency and percentage of participants ' opinions on environmental problems*

Theme and sub-themes	Student		Staff Member		Total	
	f	%	f	%	f	%
Environmental Problem Theme					38	16,5
1. Place and Time	8	16,7	2	50	10	19,2
2. Crowded Classes	6	12,5	1	25	7	13,5
3. Course Duration	9	18,8	2	50	11	21,2
4. Lack of Turkish Source	8	16,7	2	50	10	19,2

It is seen that participants express opinions more on the sub-themes of “course duration”, “place and time” (f:10, %:19,2), “lack of Turkish source” (f:10, %:19,2) than on problem of “crowded classes” (f:7, %:13,5). The relationship between the distribution of the problems experienced by the participants in the programming teaching and the data collection tools is shown In Figure 5.

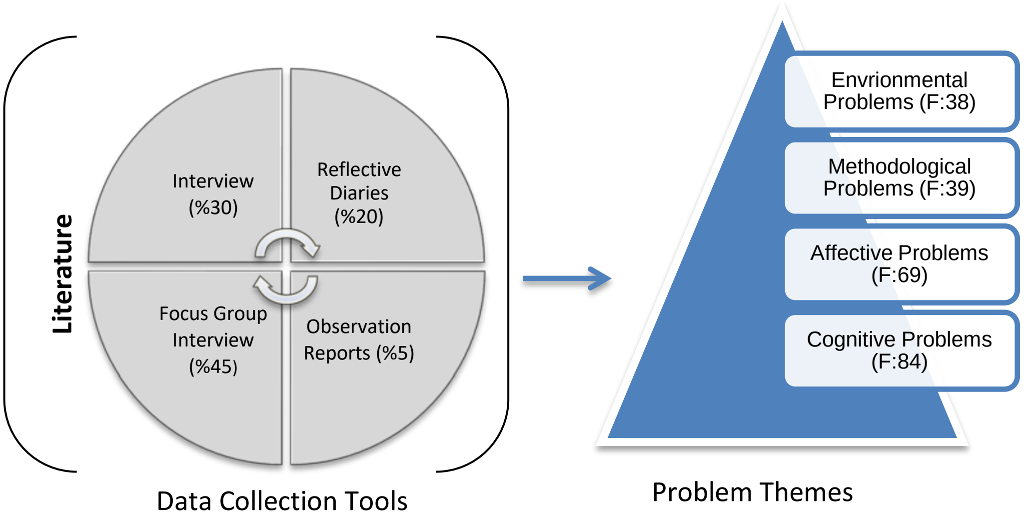

FIGURE 5. *The relationship between the distributions of the problems experienced by the participants in the programming teaching and the data collection tools*

Figure 5. demonstrates that 45% of the collected data was collected from the focus groupinterviews, 30% from interviews, 20% from reflective student diaries and 5% from observation reports. As a result of the analysis of the data, problem themes were determined as "cognitive" (f: 84), "affective" (f: 69), "methodological" (F: 39) and "environmental" (F: 38).

Design Cycles

Findings obtained by applying the graded scoring key to students and faculty members in all design cycles are shown below.

Table 6. *Findings on the content variable*

Design Variables	1st Cycle (%)			2nd Cycle (%)			3rd Cycle (%)		
	1*	2*	3*	1*	2*	3*	1*	2*	3*
Content									
1. The content of the system overlaps with the gains	31,1	60	8,9	0	2,2	97,8	0	0	100
2. The language used for the content in the system is plain and understandable	42,2	51,1	6,7	8,7	45,7	45,7	2,2	4,3	93,5
3. Information in the system is up to date	0	2,2	97,8	0	0	100	0	0	100
4. Giving the topics in the system from simple to complex	44,4	40	15,6	2,2	2,2	95,7	0	2,2	97,8
5. Visualization of algorithm topics in videos	40	42,2	17,8	13	50	37	2,2	6,5	91,3
6. Presenting content for reducing conceptual misconceptions in the system	46,7	46,7	6,7	4,3	60,9	34,8	0	8,7	91,3
7. Presenting content for supporting mathematical thinking skills in videos	42,2	42,2	15,6	2,2	2,2	95,7	2,2	2,2	95,7
8. Presenting content for supporting sequential and cyclical thinking skills in videos	51,1	37,8	11,1	0	2,2	97,8	0	0	100
9. Presenting content for reducing syntactic errors in videos	35,6	55,6	8,9	6,5	63	30,4	2,2	2,2	95,7

Table 6. *Continued*

10. Providing content to support the ability to pattern in videos	37,8	44,4	17,8	4,3	2,2	93,5	0	2,2	97,8
11. Presenting content to support the ability to make logical inferences in videos	40	46,7	13,3	2,2	2,2	95,7	2,2	2,2	95,7
12. Presenting content in videos to support abstract thinking and generalization skills	31,1	42,2	26,7	30,4	41,3	28,3	2,2	6,5	91,3
13. Video titles overlap with content	0	2,2	97,8	0	0	100	0	0	100
14. Including real-life examples in videos	28,9	55,6	15,6	4,3	2,2	93,5	0	2,2	97,8
15. Subject summaries at the end of the video	37,8	51,1	11,1	0	0	100	0	0	100
16. Specifying the purpose of the subject in the videos	46,7	46,7	6,7	0	0	100	0	0	100
17. Video times are of appropriate length	46,7	46,7	6,7	8,7	58,7	32,6	4,3	4,3	91,3
18. Online coding editor on the system	15,6	73,3	11,1	0	4,3	95,7	0	0	100
Average	34,3	43,7	22,0	4,8	18,9	76,3	1,0	2,4	96,6

*1: Insufficient, 2: Must-Be-Developed, 3: Sufficient

While the ratio of "insufficient" responses according to the content variable of the participants in the first cycle was 34.3%, it was seen that this ratio decreased by 8.8% and 1.0% in the second and third cycles. While the ratio of the participants to the "must-be-developed" level was 43.7% in the first cycle, this ratio decreased to 18.9% and 2.4% in the second and third cycles. While the ratio of the participants regarding the "sufficient" level was 22.0% in the first cycle, it was observed that this rate increased by 76.3% and 96.6% in the second and third cycles. The findings regarding the visual design variable are shown in Table 7.

Table 7. *Findings on the visual design variable*

Design Variables	**1st Cycle (%)**			**2nd Cycle (%)**			**3rd Cycle (%)**		
	1*	**2***	**3***	**1***	**2***	**3***	**1***	**2***	**3***
Visual Design									
19. The visuality of menus and buttons in a way that they reflect their functions in the system	28,9	53,3	17,8	0	4,3	95,7	0	0	100
20. System interface Colour consistency	37,8	40	22,2	0	0	100	0	0	100
21. Consistency of visual elements in the system	2,2	2,2	95,6	0	0	100	0	0	100
22. The high visual quality of videos	28,9	60	11,1	6,5	50	43,5	0	2,2	97,8
23. Readable size of texts in the system	0	2,2	97,8	0	0	100	0	4,3	95,7
24. Highlight tones used in videos being eye-catching	42,2	42,2	15,6	4,3	4,3	91,3	0	2,2	97,8
25. Completeness of text and visual elements in the system	2,2	4,4	93,3	0	4,3	95,7	2,2	4,3	93,5
26. Presence of plain and simple visual elements in the system	46,7	44,4	8,9	0	8,7	91,3	0	2,2	97,8
27. Balanced in-page placement of visual elements in the system	26,7	48,9	24,4	6,5	41,3	52,2	0	0	100
28. Being Related of Visual Elements with Subject in the system	2,2	2,2	95,6	0	0	100	2,2	2,2	95,7
29. Supporting flow of Narrator's screenshot in videos	31,1	37,8	31,1	0	8,7	91,3	0	0	100
30. Highlighting of keyboard and Mouse moves in videos	51,1	42,2	6,7	2,2	32,6	65,2	0	2,2	97,8
Average	25,0	31,7	43,3	1,6	12,9	85,5	0,4	1,6	98,0

*1: Insufficient, 2: Must-Be-Developed, 3: Sufficient

While the ratio of "insufficient" responses was 25.0% in the first cycle according to the visual design variable of the participants, it was seen that this ratio decreased by 1.6% and 0.4% in the second and third cycles. While the ratio of the participants to the "must-be-developed" level was 31.7% in the first cycle, this ratio decreased to 12.9% and 1.6% in the second and third cycles. While the ratio of the participants for the "sufficient" level was 43.3% in the first cycle, this ratio increased to 85.5% and 98.0% in the second and third cycles. The findings regarding the interaction variable are shown in Table 8.

Table 8. *Findings on the interaction variable*

Design Variables	1st Cycle (%)			2nd Cycle (%)			3rd Cycle (%)		
	1*	2*	3*	1*	2*	3*	1*	2*	3*
Interaction									
31. The high audio quality of videos	24,4	44,4	31,1	2,2	0	97,8	0	0	100
32. Videos supporting learning passion	42,2	48,9	8,9	6,5	43,5	50	2,2	6,5	91,3
33. Forum or discussion component supporting interaction in the system	0	2,2	97,8	0	0	100	0	0	100
34. Facilitating process tracking of the mouse pointer in videos	33,3	31,1	35,6	4,3	4,3	91,3	2,2	0	97,8
35. Mouse or keyboard moves in videos overlapping with the narrator's speech	22,2	55,6	22,2	8,7	0	91,3	0	4,3	95,7
36. Using fluent and simple language in videos	46,7	40	13,3	0	6,5	93,5	2,2	2,2	95,7
37. The visual elements' used in the system supporting student motivation	28,9	40	31,1	2,2	2,2	95,7	0	2,2	97,8
38. Effective use of gestures in narrator videos	48,9	37,8	13,3	6,5	2,2	91,3	0	2,2	97,8
39. The system providing synchronous course support	0	0	100	0	0	100	2,2	0	97,8
40. The system facilitates collaborative learning	20	51,1	28,9	2,2	28,3	69,6	0	6,5	93,5
41. The system supporting effective student-teacher communication	2,2	6,7	91,1	0	6,5	93,5	0	0	100
42. Correct and complete links between pages in the system	0	2,2	97,8	0	0	100	0	2,2	97,8
43. Help and support items' being functional in the system	24,4	40	35,6	10,9	45,7	43,5	2,2	2,2	95,7
44. The system providing notification support to users	31,1	46,7	22,2	2,2	17,4	80,4	2,2	0	97,8
45. Accurate and complete operation of videos such as forward, backward, pause	0	0	100	0	0	100	0	0	100
Average	21,6	29,8	48,6	3,0	10,4	86,5	0,9	1,9	97,2

*1: Insufficient, 2: Must-Be-Developed, 3: Sufficient

While the ratio of "insufficient" responses was 21.6% in the first cycle according to the interaction variable of the participants, it was observed that this ratio decreased by 3.0% and 1.9% in the second and third cycles. While the ratio of the participants to the "must-be-developed" level was 29.8% in the first cycle, this ratio decreased by 10.4% and 1.9% in the second and third cycles. It is also seen that the ratio of the participants for the "sufficient" level increased from 48.6% in the first cycle to 86.5% and 97.2% in the second and third cycles. The findings regarding the usefulness variable are shown in Table 9.

Table 9. *Findings on the practicability variable*

Design Variables	1st Cycle (%)			2nd Cycle (%)			3rd Cycle (%)		
	1*	2*	3*	1*	2*	3*	1*	2*	3*
Practicability									
46. Screen sizes of videos being modifiable	33,3	37,8	28,9	0	8,7	91,3	0	2,2	97,8
47. The system having simple and easy usage	24,4	51,1	24,4	0	4,3	95,7	0	2,2	97,8
48. Quick resolution of system errors	46,7	48,9	4,4	2,2	39,1	58,7	0	4,3	95,7
49. First-time users the system being able to easily perform basic tasks	35,6	40	24,4	0	6,5	93,5	0	6,5	93,5
50. The system's providing session support for users	2,2	4,4	93,3	0	0	100	2,2	2,2	95,7
51. Correct and complete operation of menus and buttons in the system	0	0	100	0	0	100	0	0	100
52. Operation of the system without the need for an additional program	0	0	100	0	0	100	0	2,2	97,8
53. Instant printing of codes written to the editor in the system	26,7	42,2	31,1	2,2	6,5	91,3	0	0	100
54. Customization of a system interface (item locations, colors, etc.) according to the user.	51,1	44,4	4,4	4,3	28,3	67,4	2,2	2,2	95,7
55. Users' adding content (video, text, image, editor) to the system	26,7	33,3	40	0	8,7	91,3	0	4,3	95,7
Average	24,7	30,2	45,1	0,9	10,2	88,9	0,4	2,6	97,0
General Average	27,1	34,8	38,1	2,9	13,7	83,4	0,7	2,1	97,2

*1: Insufficient, 2: Must-Be-Developed, 3: Sufficient

While the ratio of "insufficient" responses was 24.7% in the first cycle according to the practicability variable of the participants, it was observed that this ratio decreased by 0.9% and 0.4% in the second and third cycles. While the ratio of the participants to the "must-be-developed" level was 30.2% in the first cycle, this ratio decreased to 10.2% and 2.6% in the second and third cycles. While the ratio of the participants for the "sufficient" level was 45.1% in the first cycle, this ratio increased to 88.9% and 97.0% in the second and third cycles.

The percentages of the participants' responses to the graded scoring key on the basis of all variables are as follows: In the first cycle, the ratio of the participants at the level of "insufficient" compared to the graded scoring key was 27.1%, whereas, in the second and third cycles, this ratio decreased to 2.9% and 0.7%. While the ratio of the participants to the "must-be-developed" level was 34.8% in the first cycle, it was observed that this ratio decreased to 13.7% and 2.1% in the second and third cycles. While the ratio of the participants regarding the "sufficient" level was 38.1% in the first cycle, this ratio increased to 83.4% and 97.2% in the second and third cycles. The change graph of the graded scoring key data according to the cycles is shown in Figure 6.

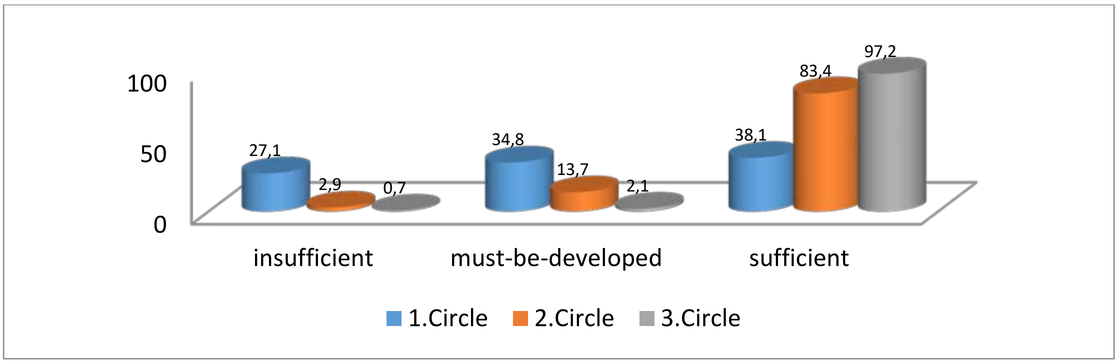

FIGURE 6. *The change graph of the graded scoring key data according to cycles*

When Figure 6 is analyzed, it is seen that the responses of the participants at the level of "insufficient" or "must-be-developed" in the direction of "sufficient" level after improvement and corrections.

DISCUSSION, CONCLUSIONS and RECOMMENDATIONS

As a result of the analysis of the data obtained, it was seen that four design variables emerged, namely "content", "visual design", "interaction" and "practicability" related to the design of video-assisted online learning environments for programming teaching. Content (Dai, Daloukas, Rigou & Sirmakessis, 2011; Cevahir & Özdemir, 2017), visual design (Fessakis, Gouli & Mavroudi, 2013; Gezgin & Adnan 2017; Gülbahar & Karal, 2018), interaction (Daniels & Walker, 2001; Williams, McCrickard, Layman, & Hussein, 2008; Kert, 2018) and practicability (Arabacıoğlu, Bülbül & Filiz, 2007; Karaca & Ocak, 2017) are supported by similar research results in various contexts in the literature.

In the research conducted by Dai et al. (2011), it is stated that the theory of "constructionism" based on the "learning by doing and living" approach supports the programming teaching process and that the content prepared especially in accordance with the acquisitions plays an important role in the learning process. Constructionism theory put forward by Seymor Papert suggests that learners gain deeper learning experiences through simple, clear and understandable content (Crichton & Carter, 2015). When evaluated in the context of the content variable, this result coincides with the research findings (the content overlaps with the acquisitions, the language used for the content is plain and understandable, the information is up-to-date, the topics are given from simple to complex).

In the study of Ben-Ari (2001), which includes suggestions for the studies to be carried out for programming teaching, it is pointed out that not only content transfer, but also the content should be organized according to the degree of difficulty, revealing deficiencies and reducing the conceptual misconceptions. In the research conducted by Guzdial (2016), 5 important principles are mentioned in programming teaching: "prior knowledge", "cognitive load", "honesty", "productivity" and "testing". The prior knowledge principle stands out in the context of content design. The prior knowledge principle states that the existing knowledge of the learners provides a basis for the new content to be learned and that when it comes to programming teaching, the content should be analyzed and increased. This principle provides a design framework on how content features should be designed. In this respect, it shows parallelism with research findings (reducing the conceptual misconceptions, supporting the content of mathematical thinking skills, supporting sequential and cyclical thinking skills, reducing syntactic errors, building patterns, logical reasoning, supporting abstract thinking and generalization skills).

On the other hand, it is stated in the literature that some difficulties experienced in the programming teaching process may be related to the visual design features of the learning environment used (Gülbahar & Karal, 2018). In this context, in the study conducted by Fessakis et al. (2013), it was stated that the visual design variables used in the process of programming teaching are supportive for the learners to analyze symbols, create familiarity and perform some tasks. In the study conducted by Sadchenko and Kushnirenko (2015), it was stated that due to the abstract and complex nature of the programming teaching process, the learners could not define the tasks related to the programming concepts which should be visual and consistent to reflect the functions of the items to be used in the learning environment. Calao et al. (2015) stated that students had difficulties in reading and following during the programming process. Gezgin et al. (2017) stated that the learners had difficulties due to the complex visual elements in the learning environment, while Arabacıoğlu, Bülbül and Filiz (2007) stated that the students had problems when the pointers could not lead to the correct result. The researchers point out that the texts used in the learning environment should be readable, the visual elements should be consistent, the color harmony should be highlighted and the pointers should be emphasized for these difficulties and problems. In this respect, the findings obtained from the literature overlap with the results obtained from this research in the context of the visual design variable.

In the study carried out by Demir (2015), different uses of educational programming language were investigated. It turns out that the different usage types of programming language are indirectly related to visual design features. In the research conducted by Ouahbi et al. (2015), the method based on the design of simple games was used by using a scratch environment. As a result of the research, it was concluded that visual design features are important for learners. The integrity of the text and visual elements, the use of plain and simple elements, and the well-balanced in-page placement of the visual elements indirectly influenced this result. It can be said that the findings obtained in this context are in line with the results of the research.

The literature review reveals that the efficiency of programming teaching should be supported with learner-centered applications (Kert, 2018). In this context, Daniels and Walker (2001) point out the importance of the interaction element in programming teaching, stating that the pair programming method contributes to the efficiency of the programming teaching process in conjunction with cooperative learning. The pair programming method is defined as a method in which the two people take part in the programming process, contribute to the decision-making processes for the programming process in all aspects of the people and provide critical feedback support (Kamthan, 2009). It may also be observed that the pair programming application coincides with the interaction features regarding the design of the video-assisted online learning environment for programming teaching. It is observed that Williams et al. (2008) emphasized especially the learner-teacher and learner-learner interaction in the guideline including 11 items that they put forward regarding the efficiency of the programming teaching process. The results of the research conducted by Williams et al. (2008) coincide with the characteristics of the interaction variable included in this research (supporting cooperative learning, functional helpful and supporting elements, and providing notification support to users).

Motivation is defined as one of the most important elements of the programming teaching process (Gülbahar & Karal, 2018). It is stated that the programming teaching process, in which the passive student role comes to the fore, should be supported to discover, manage the content and sustain the motivation of the lesson. This is in line with the features of the interaction (visual elements that support student motivation and learning desire and the interaction of the forum and discussion component) regarding the design of a video-assisted online learning environment for programming teaching. In the research conducted by Ünal and Bay (2010), educational software was designed to teach Java programming language. In the research, in which student development is also monitored, it is stated that the designed multimedia software improves the sense of sharing among learners and supports social communication and interaction skills. It is stated that learner-teacher interaction emerged as an important element especially in the teaching of abstract and complex structures such as programming. In this context, it is stated that interaction features will be effective in the design of applications for programming teaching. These results overlap with the results obtained in the research.

It is not possible to develop metacognitive gains such as problem-solving, critical thinking and algorithmic thinking with superficial teaching activities (Kert, 2018). The abstract and complex process of programming is a form of approach that is required for learners to do programming studies in a flexible environment. Thus, it is in parallel with the practicability of the video-assisted online learning environment for programming teaching regarding the design (an adaptation of the system interface according to the user, the operation of the system without the need for an additional program, the output of the codes written to the editor instantly). In the research conducted by Karaca and Ocak (2017), the effect of the reverse learning method for the algorithm and programming course on students' academic success was investigated. It has been stated that besides the play, stop, and rewind features of the videos used in the research, the ability to add a voice note to the videos, adding a table of contents to the videos and giving feedback may support the practicability features of the learners. This result coincides with the practicability features of the video-assisted online learning environment for programming teaching.

Conclusions (Design Report)

As a result of the analysis of the data collected from the participants based on the literature, It was determined that there was a problem in four themes of the programming teaching process: "cognitive" (f: 84), "affective" (f: 69), "methodological" (f: 39) and "environmental" (f: 38). It was observed that 16% of the problems were caused by environmental, 27% by methodological, 30% by affective and 37% by cognitive factors. As a result of the analysis of the data collected from the participants, the categories of "content" by 33%, "visual design" by 22%, "interaction" by 27% and "practicability" by 18% were reached in the design of a video-assisted online learning environment. It was found that the video-assisted online learning environment for programming teaching reached a 97.2% of "sufficient" level when 3 cycles (1st cycle, 2nd cycle, 3rd cycle) were rotated through design, analysis and redesign according to the design variables determined. This shows that the proposed framework can be used in the design of these environments. The relationship between the problem and design variables regarding the proposed framework within the scope of the research is shown in the figure below.

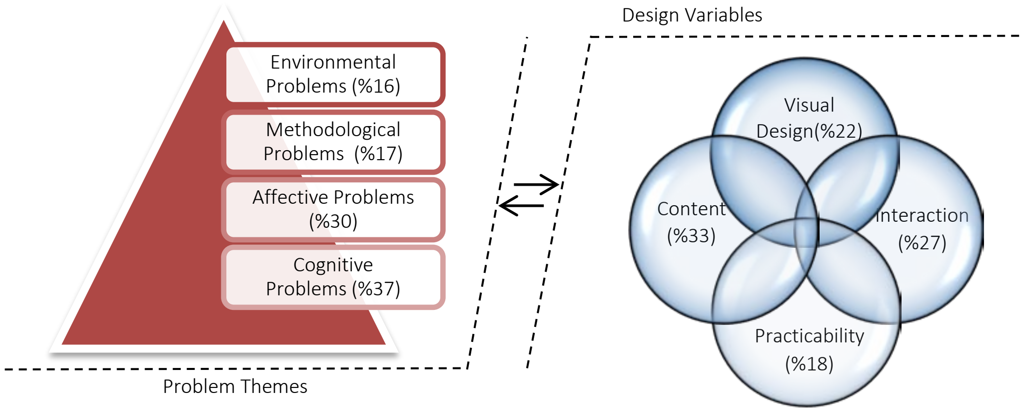

FIGURE 7. *The relationship between the problem and design variables*

Recommendations

Studies on the effects of each of the problem themes that arise during the determination of the problems experienced during the programming teaching process on the development of programming skills can be conducted. Studies can be conducted to develop or monitor programming skills for blended learning environments, taking into account the design framework for the video-assisted online learning environment for programming teaching. The framework for the design of a video-assisted online learning environment for programming teaching can be used in the development of videos to be used in flipped classroom model applications. The design framework for video-assisted online learning environments for programming teaching can be used in the process of transforming face-to-face learning into blended learning environments by applying it in similar courses or different research groups.

REFERENCES

Abbasi, S., Kazi, H., & Khowaja, K. (2017). *A systematic review of learning object-oriented programming through serious games and programming approaches*. In 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-6). IEEE.

Allen, I. E., & Seaman, J. (2011). *Going the Distance: Online Education in the United States, 2011*. Sloan Consortium. Newburyport.

Al-Tahat, K., Taha, N., Hasan, B., & Shawar, B. A. (2016, July). *The impact of a 3d visual tool on female students' attitude and performance in computer programming*. In 2016 SAI Computing Conference (SAI) (pp. 864-867). IEEE.

Arabacıoğlu, C., Bülbül, H. ve Filiz, A. (2007). *Bilgisayar programlama öğretiminde yeni bir yaklaşım*. Akademik Bilişim 2007 Konferansı, Dumlupinar Üniversitesi, Kütahya.

Balcı, A. (2007). *Sosyal Bilimlerde Araştırma: Yöntem, Teknik ve İlkeler*. Ankara: Pegem A Yayıncılık.

Baltacı, S., Yıldız, A., Kıymaz, Y., & Aytekin, C. (2016). Üstün yetenekli öğrencilere yönelik geogebra destekli etkinlik hazırlamak için yürütülen tasarım tabanlı araştırma sürecinden yansımalar. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi*, *1*(39), 70-90.

Bannister, N., & Arbaugh, F. (2018). In the spotlight: Supporting focus teachers in video-based collaborative learning settings. *Advanced Learning and Teaching Environments: Innovation, Contents and Methods*, 73.

Basten, F., Evers, A., Geijsel, F., & Vermeulen, M. (2018, January). 2*1st Century leadership skills for learning networks at school*. In International Conference for School Effectiveness and Improvement: Deepening School Change for Scaling: Principles, Pathways & Partnerships.

Bayazıt, A. ve Akçapınar, G. (2018). Çevrimiçi dersler için video analitik aracının tasarlanması ve geliştirilmesi. *İlköğretim Online*, *17*(1), 14-25.

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching, 20*(1), 45-73.

Bentley, J. (2016). *Programming pearls*. Addison-Wesley Professional.

Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? Patterns of Difficulties Related to Programming Learning Mid-Stage. *ACM SIGSOFT Software Engineering Notes*, *41*(6), 1-6.

Brame, C. J. (2016). Effective educational videos: Principles and guidelines for maximizing student learning from video content. *CBE—Life Sciences Education*, *15*(4), 1-6.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834-860.

Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş. ve Demirel, F. (2017). *Bilimsel Araştırma Yöntemleri*. Ankara: Pegem Yayınları.

Calao, L. A., Moreno-León, J., Correa, H. E. and Robles, G. (2015). Developing mathematical thinking with scratch. *In Design for Teaching and Learning in A Networked World, 2*(2) 17-27.

Cevahir, H., & Özdemir, M. (2017). *Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri.* Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu içinde, İnönü Üniversitesi, Malatya.

Choi, H. J., & Johnson, S. D. (2005). The effect of context-based video instruction on learning and motivation in online courses. *The American Journal of Distance Education*, *19*(4), 215-227. *Classroom*. San Francisco, CA: Jossey-Bass.

Çokluk, Ö., Yılmaz, K., & Oğuz, E. (2011). Nitel bir görüşme yöntemi: Odak grup görüşmesi. *Kuramsal Eğitimbilim Dergisi*, *4*(1), 95-107.

Crichton, S. E. and Carter, D. (2015). Taking making into the schools: An immersive professional development approach. *Handbook of Research on Teacher Education in the Digital Age, 1*(1), 412-438.

Dai, V., Daloukas, V., Rigou, M., & Sirmakessis, S. (2011). Context-free educational games: open-source and flexible. In *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches* (pp. 1064-1085). IGI Global.

Daniels, S. E. and Walker, G. B. (2001). *Working through Environmental Conflict: The Collaborative Learning Approach*. Westport, CT: Praeger.

Delen, E., Liew, J., & Willson, V. (2014). Effects of interactivity and instructional scaffolding on learning: Self-regulation in online video-based environments. *Computers & Education*, *78*, 312-320.

Demir, F. (2015). *Programlama Öğretiminde Eğitsel Programlama Dilinin Farklı Kullanımlarının Programlama Başarısı ve Kaygısına Etkisi* (Yayınlanmamış doktora tezi). Atatürk Üniversitesi/Eğitim Bilimleri Enstitüsü, Erzurum.

Demirer, V., & Nurcan, S. A. K. (2016). Programming education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, *12*(3), 521-546.

Dodson, S., Roll, I., Fong, M., Yoon, D., Harandi, N. M., & Fels, S. (2018, June). *An active viewing framework for video-based learning*. In Proceedings of the Fifth Annual ACM Conference on Learning at Scale (pp. 1-4).

Erdem, E. (2018). *Blok Tabanlı Ortamlarda Programlama Öğretimi Sürecinde Farklı Öğretim Stratejilerinin Çeşitli Değişkenler Açısından İncelenmesi* (Yayınlanmamış yüksek lisans tezi), Başkent Üniversitesi/Eğitim Bilimleri Enstitüsü.

Fessakis, G., Gouli, E. and Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education, 1*(63), 87-97.

Franzoni, A. L., Assar, S., Defude, B., & Rojas, J. (2008, July). *Student learning styles adaptation method based on teaching strategies and electronic media*. In 2008 Eighth IEEE International Conference on Advanced Learning Technologies (pp. 778-782). IEEE.

Gabriele, K. M., Holthaus, R. M., & Boulet, J. R. (2016). Usefulness of video-assisted peer mentor feedback in undergraduate nursing education. *Clinical Simulation in Nursing*, *12*(8), 337-345.

Gezgin, D. M. ve Adnan, M. (2016). Makine mühendisliği ve ekonometri öğrencilerinin programlamaya ilişkin öz yeterlik algılarının incelenmesi. *Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi*, *17*(2), 509-525.

Gulbahar, Y., Kalelioglu, F., & Kert, S. B. (2018, October). *Teaching computational thinking to ın-service computer science teachers through a massive open online course*. In E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (pp. 922-928). Association for the Advancement of Computing in Education (AACE).

Gülbahar, Y., Karal, H. (2018). *Kuramdan Uygulamaya Programlama Öğretimi*. Ankara: Pegem Akademi Yayıncılık.

Guzdial, M. (2016). Five Principles for Programming Languages for Learners. Retrieved from http://cacm.acm.

Hajhashemi, K., Caltabiano, NJ, Anderson, N. ve Tabibzadeh, SA (2018). Video destekli öğrenme ortamlarında öğrencilerin çoklu zekaları. *Eğitimde Bilgisayar Dergisi*, *5* (3), 329-348.

Hrastinski, S. (2009). A theory of online learning as online participation. *Computers & Education*, *52*(1), 78-82.

Kamthan, P. (2009). A Methodology for Integrating Information Technology in Software Engineering Education. En R. Donnelly & F. McSweeney (Eds.), *Applied E-learning and E-teaching in Higher Education* (pp. 201-219).Hershey: Information Science Reference

Karaca, C., & Ocak, M. A. (2017). Effects of Flipped Learning on University Students' Academic Achievement in Algorithms and Programming Education. *International Online Journal of Educational Sciences*, *9*(2), 527-543.

Kert, S. B. (2018). Programlama öğretimi için pedagojik yaklaşımlar. Gülbahar, Y. ve Karal, H. (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi* içinde (s. 93-130), Ankara: Pegem Akademi Yayıncılık.

Kitchin, R. (2017). Thinking critically about and researching algorithms. *Information, Communication & Society*, *20*(1), 14-29.

Kuzu, A., Çankaya, S. and Mısırlı, Z. A. (2011). Tasarım tabanlı araştırma ve öğrenme ortamlarının tasarımı ve geliştirilmesinde kullanımı. *Anadolu Journal of Educational Sciences International*, *1*(1), 19-35.

Means, B., Toyama, Y., Murphy, R., & Baki, M. (2013). The effectiveness of online and blended learning: A meta-analysis of the empirical literature. *Teachers College Record*, *115*(3), 1-47.

Miles, M, B. and Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook.* Washington DC: Sage.org/blogs/blog-cacm/203554-five-principles-for-programming-languages-for-learners/fulltext

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A. and Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, *1*(191), 1479-1482.

Palloff, R. M. , & Pratt, K. (2007). *Building Online Learning Communities: Effective Strategies For The Virtual*

Priest, S., & Gass, M. (2017). *Effective Leadership in Adventure Programming, 3E*. Human Kinetics.

Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, *45*(5), 583-602.

Reychav, I., & Wu, D. (2015). Mobile collaborative learning: The role of individual learning in groups through text and video content delivery in tablets. *Computers in Human Behavior*, *50*, 520-534.

Sadchenko, A. V., Kushnirenko, O. A., & Troyansky, A. V. (2015). The algorithm of random length sequences synthesis for frame synchronization of digital television systems. *Odes' kyi Politechnichnyi Universytet. Pratsi*, *2015*(3), 97-104.

Sayın, Z., & Seferoğlu, S. S. (2016). Yeni bir 21. Yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı*, 3-5.

Sheth, S., Murphy, C., Ross, K. A., & Shasha, D. (2016, February). *A course on programming and problem solving*. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (pp. 323-328).

Shih, R. C. (2010). Blended learning using video-based blogs: Public speaking for English as a second language students. *Australasian Journal of Educational Technology*, *26*(6).

Singh, R. N., & Hurley, D. (2017). The effectiveness of teaching and learning process in online education as perceived by university faculty and instructional technology professionals. *Journal of Teaching and Learning with Technology*, *6*(1), 65-75.

Tall, D. (1993). Zihin ve bilgisayar arasındaki ilişkiler: süreçler, imgeler, semboller. In Matematik ve Fen Gelişmiş Eğitim teknolojileri (s. 385-413). Springer, Berlin, Heidelberg.

Ünal, C. and Bay, Ö. F. (2010). Java programlama dili'nin bilgisayar destekli öğretimi. *Bilişim Teknolojileri Dergisi, 2*(1), 1-14.

Wang, S. K., & Hsu, H. Y. (2017). A Design-Based Research Capturing Science Teachers' Practices of Information and Communication Technologies (ICTs) Integration Using the New Literacy Framework. *Journal of Computers in Mathematics and Science Teaching*, *36*(4), 387-396.

Williams, L., McCrickard, D. S., Layman, L., & Hussein, K. (2008). *Eleven guidelines for implementing pair programming in the classroom*. In Agile 2008 Conference (pp. 445-452). IEEE.

Yıldırım, A ve Şimşek, H. (2005). *Sosyal Bilimlerde Nitel AraşTırma Teknikleri*. Ankara: Seçkin Yayınları.

Zhang, D., Zhou, L., Briggs, R. O., & Nunamaker Jr, J. F. (2006). Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management*, *43*(1), 15-27.