

Exploring the Usability of Virtual Robotics Programming Curriculum for Robotics Programming Teaching

Servet KILIÇ¹, Seyfullah GÖKOĞLU²

¹*Computer Technologies Department, Technical Sciences Vocational High School, Ordu University, Ordu, Turkey*

²*Computer Technology and Information Systems, Faculty of Sciences, Bartın University Bartın, Turkey*

e-mail: servetkilig@odu.edu.tr, gokogluseyfullah@gmail.com

Received: May 2021

Abstract. This study aims to explore the usability of the virtual robotics programming curriculum (VRP-C) for robotics programming teaching. Pre-service computer science (CS) teachers were trained for robotics programming teaching by using VRP-C in a scientific education activity. After training, views of the participants were revealed by using a scale and an evaluation form consisting of open-ended questions. Results show that VRP-C is compatible with the curriculum for robotics programming teaching in schools, and pre-service CS teachers tend to use VRP-C in their courses. They think that VRP-C will be beneficial for robotics programming teaching in terms of content, functionality, and cost. Compatibility, visual design, feedback, time management, fiction, gamification, and cost are the characteristics that increase the usability of VRP-C. VRP-C can be used as an online tool for robotics programming training due to the necessity of transition to distance education because of the COVID-19 pandemic.

Keywords: robotics programming teaching, educational virtual robotics, virtual robotics programming curriculum, technology acceptance, computer science, pre-service teachers.

1. Introduction

Learning environments, created with a constructivist approach, contribute to students' taking an active role in problem-solving processes and learning by doing with their peers (Atmatzidou and Demetriadis, 2016; Grover and Pea, 2013). Especially in computer science (CS) education, constructivist applications are actively used and the educators try to create interactive learning environments that are specific to the individual, focusing on active participation and targeting individual construction (Kert, 2018). Seymour Papert, who advocates the idea that individuals structure knowledge in their minds in a produc-

tion process, proposed the constructionism based on the constructivism (Papert & Harel, 1991). In the constructionism, students' process of generating new ideas is substantially seen in the period of creating a product (eg a robot, a poem, a computer program) that they can see their own mistakes and share with their peers (Kafai & Resnick, 1996). While constructionism adopts the idea of "building knowledge structures" of constructivism, it goes further and focuses on the construction of structures created in the mind (Papert & Harel, 1991). Constructivism tends to ignore the importance of individual preferences, media used, and context at different stages of individuals' development (Ackermann, 2001). Constructionism, on the other hand, focuses more on "learning by making" rather than cognitive potentials (Papert & Harel, 1991). Building a mind-designed castle out of sand (Papert & Harel, 1991) or building a mind-designed robot out of Lego pieces can be given as examples. As a result of the studies conducted by Papert and his friends to understand how children think and learn, educational robotics (ERs) have been used in education (Ackermann, 2001). Learning environments where problem-solving and critical thinking skills are developed in cooperation can be created by using ERs (Eguchi, 2016). Activities using ERs play an essential role in increasing students' motivation by drawing their attention (Alimisis, 2013; Eguchi, 2010), developing their problem-solving (Karim *et al.*, 2015; Liu *et al.*, 2010), and computational thinking skills (Amatzidou and Demetriadis, 2016; Chen *et al.*, 2017). ERs also reduce the abstraction and complexity of programming language concepts in computer sciences (Touretzky, 2013). For these reasons, ERs are used in formal and informal education at all levels from kindergarten to university.

ERs are robot sets that learners can program practically (Üçgül, 2018). Learners usually work with ERs in groups in learning environments where they can develop their collaboration, communication, problem-solving, critical thinking, and creativity skills (Eguchi, 2016). In such environments that are called robotics learning environments, various ERs produced by combining parts or produced in a ready-made way are used. Lego Mindstorm EV3, Bee-Bot, Robbo, Cubelets, Ozobots are examples of ERs that are widely used for programming teaching. There are several difficulties in creating robotics learning environments and conducting learning activities in these settings. Schools have difficulties in creating robotics education classes, especially due to the high cost of ERs. Additional time is required before ERs can be designed and used effectively in crowded classrooms. In recent years, educational virtual robotics (EVRs) has been preferred to reduce the limitations specified for the use of ERs. Easy configuration of EVRs, efficient use of time by teachers (Flot *et al.*, 2012), easy creation of learning activities (Liu *et al.*, 2013), access to these environments outside the classroom by teachers and students (Liu *et al.*, 2013), providing faster feedback to problems that occur (Liu *et al.*, 2013) and low cost (Flot *et al.*, 2012; Witherspoon *et al.*, 2017) are the reasons that increase the preferability of these environments. EVRs enable students to focus on higher-level thinking skills and computational principles required for programming by reducing the disappointment and distractibility caused by mechanical errors in using physical ERs (Witherspoon and Schunn, 2019). In this way, the cognitive load that will arise from the mechanical knowledge required to design physical VRs can be reduced (Witherspoon and Schunn, 2019).

EVRs have similar functions as physical ERs. They can be customized and programmed for different activities or challenges. In robotic programming teaching, EVRs can be used by customizing their design for students to understand basic structures such as robot movements, decision structures (if, switch), and loops (for, while) in robotics programming teaching. Redesigning EVRs each time for different activities causes a waste of time for teachers and learners. The virtual robotics programming curriculum (VRP-C), which includes various activity environments and customized EVRs designs for robotics programming teaching, has been used in recent years to eliminate this limitation. VRP-C provides a teaching content comprising progressive activities for robotics programming education and enables students to learn these contents individually (Higashi *et al.*, 2017; Witherspoon *et al.*, 2017; Witherspoon *et al.*, 2018; Witherspoon and Schunn, 2019). VRP-C includes procedural scaffolds (worked examples, guided videos), dynamic mini-challenges, a visual programming language, and Robot Virtual Worlds (RVW) components which support students' robotic programming knowledge and computational thinking skills. Students' computational thinking skills improve and their interest and motivation increase by gaining more knowledge towards robotics programming when they work with VRP-C (Witherspoon *et al.*, 2017; Witherspoon *et al.*, 2018).

Similarly, Witherspoon and Schunn, (2019) emphasized that students' computational thinking skills and attitudes towards programming develop positively depending on the goals of teachers in the classrooms where VRP-C are used. Witherspoon *et al.* (2018) stated that VRP-C, which offers dynamic programming challenges using a visual programming language, can help students to develop the thinking structures that enable them to transfer learned knowledge to different problem situations. For the broad usage of VRP-C, Higashi *et al.* (2017) revealed that VRP-C influences teachers' and students' engagement, depending on motivation factors in an open-access course (MOOC) conducted on the basic subjects of VRP-C.

In recent years, the use of VRP-C as an alternative to physical ERs has become widespread. When the studies are examined, it is seen that the effectiveness of VRP-C is mostly analysed at the K-12 level (Berland and Wilensky, 2015; Witherspoon *et al.*, 2017, Witherspoon *et al.*, 2018; Witherspoon and Schunn, 2019). VRP-C is generally used in programming education within the scope of CS courses in schools. Alimisis and Kynigos (2009) emphasize that improving teachers' knowledge and skills by taking their opinions and thoughts regarding the use of ERs is an important factor for the successful integration of ERs into learning environments. In this context, we consider that it is important to reveal the opinions of the pre-service teachers who will conduct CS courses to evaluate the usability of VRP-C in schools. Davis (1989) emphasized that to determine the usability of technology, individuals' perception of usefulness and ease of use upon that technology have to be revealed. So, Davis (1989) developed the technology acceptance model (TAM) to determine the user acceptance of information technology (see Fig. 1).

TAM is aimed to investigate the individuals' levels of acceptance for the relevant technology based on the perceived ease of use and perceived usefulness. They are the two determinants in the TAM. Davis (1989) defined perceived usefulness as the degree

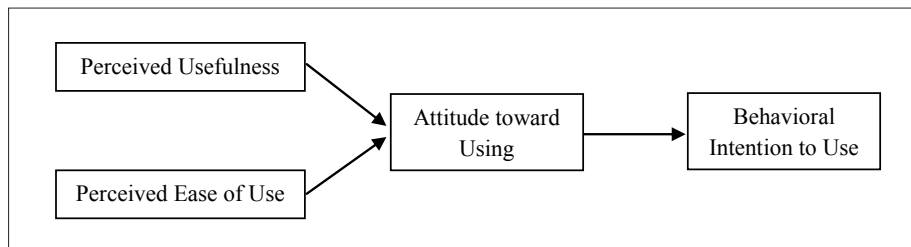


Fig. 1. The Technology Acceptance Model.

to which an individual believes that using a technology would enhance his or her job performance. In contrast, perceived ease of use is the degree to which an individual believes that using a particular technology would be free of effort. Attitude is a response to the use of technology as acceptance or rejection as an impact if someone uses information technology in their work. The response of acceptance or rejection then affects the individual's decision. While the perception of usefulness and ease of use directly affects individuals' attitudes (Dastjerdi, 2016), changes in attitude also determine the behavioral intention to use technology (Lee *et al.*, 2007). In this sense, this study aims to evaluate the usability of VRP-C for robotics programming education from the perspective of pre-service CS teachers. For this purpose, the research questions are presented below:

1. How do pre-service CS teachers accept the VRP-C content for robotics programming teaching?
2. How do pre-service CS teachers evaluate the usability of VRP-C in robotics programming teaching?

2. Method

2.1. Participants

The participants of this study are twenty-eight pre-service CS teachers studying Computer Education and Instructional Technologies (CEIT) at fifteen different universities in Turkey. They were selected among the university students who voluntarily applied to attend a robotics programming training. Pre-service CS teachers learn various programming languages, methods, and how to teach them during their undergraduate education. After graduation, they teach "Information Technologies and Software" courses at the K-8 level and "Computer Science" courses at the high school level. In these courses, CS teachers teach students regarding general programming and robotics programming. University diversity and academic achievement were used as criteria in determining the participants. The demographic information of the participants is presented in Table 1.

Table 1
Demographics of the participants

Characteristics	N	%
Female	20	71
Male	8	29
3rd grade	11	39
4th grade	17	61

2.2. Virtual Robotics Programming Curriculum (VRP-C)

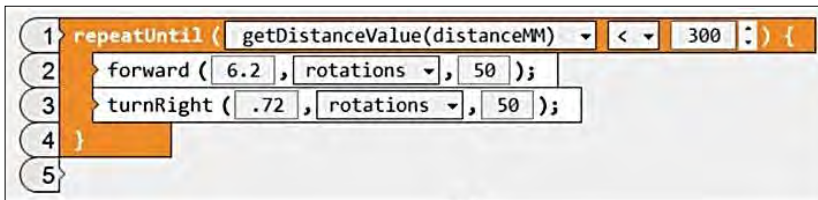
VRP-C, developed by Carnegie Mellon University and Robomatter allowing the use of different programming techniques (see Fig. 2) and structures (if-else, loop, variables, operators etc), and robotics components (eg mechanical parts, motors, sensors) is a teaching curriculum that includes various programming challenges.

```

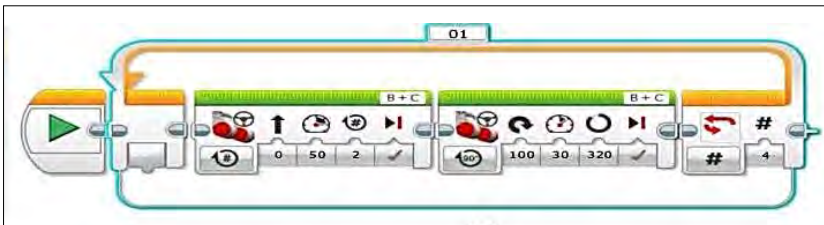
13  task main()
14  {
15      repeatUntil (getDistanceValue(distanceMM) < 300) {
16          forward(6.2, rotations, 50);
17          turnRight(.72, rotations, 50);
18      }
19  }

```

a)



b)



c)

Fig. 2. Text (a), graphical (b) and icon-based (c) programming techniques in VRP-C.

We used the virtual version of the Lego Mindstorm EV3 and the EV3-G icon-based programming language in training given with VRP-C to the participants. Lego Mindstorm EV3 is an educational robotics kit that is widely preferred around the world (Benitti, 2012; Berland and Wilensky, 2015; Üçgül, 2013) and used in international competitions (eg First Lego League, First Robotics Competition). EV3-G is a free-ware popularly used in physical and virtual robots and developed by Lego. The codes developed on the computer can be transferred to the physical EV3 robot via programmable brick, so the robot can perform the programmed task with no cable connection. We used virtual brick for this process in VRP-C. The virtual brick also connects the VRP-C with the EV3-G programming environment. The progress of students who log in to the VRP-C is registered to the CS-STEM network (cs2n.org, 2020). The system gives badges to the students for each completed challenge. Students can monitor their progress and receive a certificate through these badges, which are among the elements of gamification (Bunchball, 2010; Groh, 2012). There are 4 modules entitled as Movement, Sensors, Decisions, and Final Challenge (Search and Rescue) in VRP-C (see Table 2).

The challenges in VRP-C are associated with real life and ensure learning different programming structures (see Fig. 3).

There are short-time introduction videos to provide clues about the relevant challenge and discover the link of its with daily life in modules on VRP-C. These videos play an encouraging role for students to complete the challenges considering attention, interest, and motivation (Witherspoon *et al.*, 2017) by reducing extraneous cognitive processing, managing essential processing, and fostering generative processing (Mayer, 2008). Witherspoon *et al.* (2017) stated that the programming activities in VRP-C that have a rich three-dimensional design and are prepared in relation to daily life contribute to the development of students' computational thinking skills.

Table 2

Modules and challenges in VRP-C

Modules	Challenges	Context
Movement	8	Measurement (eg., centimeters, degrees, rotations) Basic forward, backward and turning movement command sequences Basic arm lifting, lowering and carrying command sequences
Sensors	14	Uses of touch, ultrasonic, light, gyro sensors in robotics Equations with inequalities Boolean logic Decision structures (if-else, wait-until), loops
Decisions	10	Combination of different sensors (touch, ultrasonic, light, gyro, motor rotation sensors) Combination of different programming structures (math blocks, arrays, loop, variable definition, decision structures)
Final Challenge (Search and Rescue)	1	A challenge including 4 different difficulties that require the use of all learned knowledge (passing through the untidy room, firefighting, passing through the empty room, and human rescue)

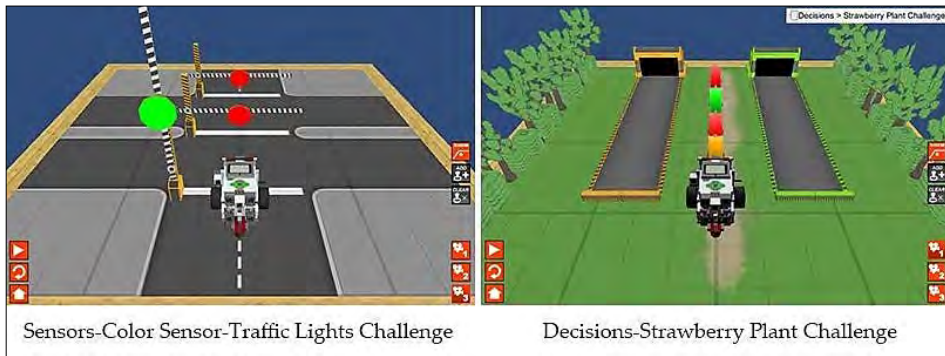


Fig. 3. Various challenges in VRP-C.

2.3. Robotics Programming Training

Five faculty members from CEIT gave thirty-two hours of robotics programming training to the twenty-eight pre-service CS teachers (see Fig. 4). VRP-C environment and physical EV3 robot were used together in the training. Thus, the participants experienced the use of VRP-C environment and physical EV3 robot in robotics programming teaching and discovered the similarities and differences of these two instruments.

At the beginning of the training, the participants designed physical EV3 robots following a guide prepared by Lego. Then, the components of the EV3 robot (brick, engine, mechanical parts, sensors etc) were introduced comparatively on the physical robot and the virtual robot in the VRP-C. The educators informed the participants about each component and performed various sample applications with them. After this pre-training, complex programming challenges in VRP-C were given to the participants to complete by themselves. At the end of the training, we expected the participants to complete the Search and Rescue task, which is the final challenge and requires the use of all knowledge gained on VRP-C modules. During the training, the participants attempted to complete the final challenge based on what they learned by studying indi-



Fig. 4. Scenes from robotics programming training.

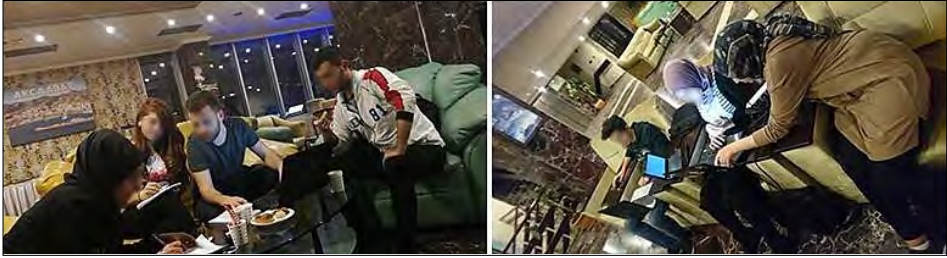


Fig. 5. Scenes from group study of the participants.

vidually or in groups (see Fig. 5). At the end of the training, participants presented their algorithms for the Search and Rescue challenge. The educators rewarded participants who successfully completed the challenge.

2.4. Data Collection and Analysis

We gathered the data from the participants at the end of the training by using The Acceptance Scale of Lego Robotics Instructional Practices (Çukurbaşı *et al.*, 2018) and an evaluation form. The Acceptance Scale of Lego Robotics Instructional Practices consist of four factors: perceived usefulness, perceived ease of use, intention to use, and attitude. The total variance explained by the scale was 78.665% and the Cronbach's Alpha reliability coefficient was .956. The 5-point Likert type scale has been developed to determine the pre-service teachers' acceptance levels of Lego Robotics instructional practices. We used the scale to evaluate the acceptance levels of pre-service CS teachers for using VRP-C in robotics programming teaching.

We collected qualitative data regarding the usability of VRP-C for robotics programming teaching with the evaluation form consisting of open-ended questions. The evaluation form consists of two sections. The participants were requested to evaluate the content of VRP-C (modules and subjects) within the framework of usefulness, ease of use, attitude, and intention to use in the first section of the form. In the second section, we asked the participants to evaluate the usability of VRP-C in robotics programming teaching.

We analyzed the data gathered from the participants according to the research problems. For the first research problem, we used the responses given to the scale and to the first section of the evaluation form. For the second research problem, we used the qualitative data gathered from the second section of the evaluation form. The data gathered from the scale was analyzed using descriptive statistics as well as the data obtained from the evaluation form was examined with content analysis. The separate responses to the evaluation form were combined in a file and transferred to the Nvivo 10, a qualitative analysis software. Researchers independently read the transferred data and created two different code lists. Then, they together compared the codes and reached an agreement on a common code list. The common code list was transferred back to Nvivo, then the researchers independently coded the data according to this list. The Cohen's Kappa reliability coefficient calculated to determine the agreement between

the researchers was found as .80. ($p < .05$). Landis and Koch (1977) expressed that this value is a very good agreement level.

3. Findings

3.1. Acceptance of VRP-C for Robotics Programming Teaching

Table 3 shows the statistics of pre-service CS teachers' responses to the Acceptance Scale of Lego Robotics Instructional Practices.

When Table 3 is examined, the participants think that VRP-C can be used for robotics programming teaching ($\bar{X} = 4.47$). Participants also have positive views in each factor similar to the overall scale. Participants received the highest score from the perceived ease of use factor. These results indicate that the participants think VRP-C will be easy to use in robotics programming training and will be useful. At the end of the training, the participants who have developed a positive attitude towards the use of VRP-C in robotics programming teaching intended to use VRP-C in their future courses.

Besides the scale, the usability of VRP-C's content (modules and subjects) in robotics programming teaching was examined with the qualitative data gathered from the evaluation form. We categorized the participants' responses to the first section of the evaluation form in the context of the factors of the acceptance scale of LEGO robotics instructional practices, perceived usefulness, perceived ease of use, intention to use, and attitude.

3.1.1. Perceived Usefulness

The pre-service CS teachers stated that the content of VRP-C was compatible with the curriculum used in robotics programming teaching, and the modules included were designed considering individual learning differences and establishing a relationship with daily life. The expressions of the pre-service CS teachers about the usefulness of VRP-C are presented below:

“The content of VRP-C was generally successful. Modules and their ordering were suitable for robotics programming teaching (Büşra).

Table 3
Participants' responses to the scale

Factors	Items	Mean	Sd.
Perceived Usefulness	4	4.49	0.57
Perceived Ease of Use	4	4.62	0.57
Attitude toward Using	5	4.55	0.45
Behavioral Intention to Use	3	4.13	0.56
Total	16	4.47	0.42

VRP-C includes a curriculum that is suitable for each student's level in terms of content and is prepared from the basics. A student who has never known about programming can easily learn robotics programming with VRP-C without any challenge in understanding (Öznur). In VRP-C, it was very effective to include challenges consisting of environments that can be used in daily life and related to the subjects learned in the courses (Nuriye)."

3.1.2. Perceived Ease of Use

Pre-service CS teachers stated that modules in VRP-C are organized from simple to complex and from part to whole, and they can be easily used by students because of its interface.

"The modules in VRP-C were fragmented and progressing from easy to difficult. These were useful for students to understand programming (Feyza). VRP-C makes it easier for students to understand robotics programming, as it follows a route from part to whole, from simple to complex (Feyzanur). VRP-C has content that we can easily use in robotics programming courses. Its interface is very simple and straightforward (Gisu-Sanem)."

3.1.3. Attitude

Pre-service CS teachers have developed a positive attitude towards VRP-C by stating that VRP-C will increase students' interest and curiosity towards robotics programming.

"Since the coding component of VRP-C is visual, it will be interesting for students. VRP-C teaches robotics programming in a simplified way. It allows us to simulate the codes. This will be very interesting for students (Betül). VRP-C will make students interested in the lesson and increase their curiosity since its remarkable interface (Büşra)."

3.1.4. Intention to Use

Pre-service CS teachers intend to use the VRP-C as an assistive tool in their future courses. Mehmet explained his intention stating:

"I will use VRP-C in my courses. Because students can learn better by seeing which block or code does what through the simulator."

3.2. Usability of VRP-C in Robotics Programming Teaching

In the second section of the evaluation form, pre-service CS teachers were asked to evaluate the usability of VRP-C in robotics programming teaching. As a result of the

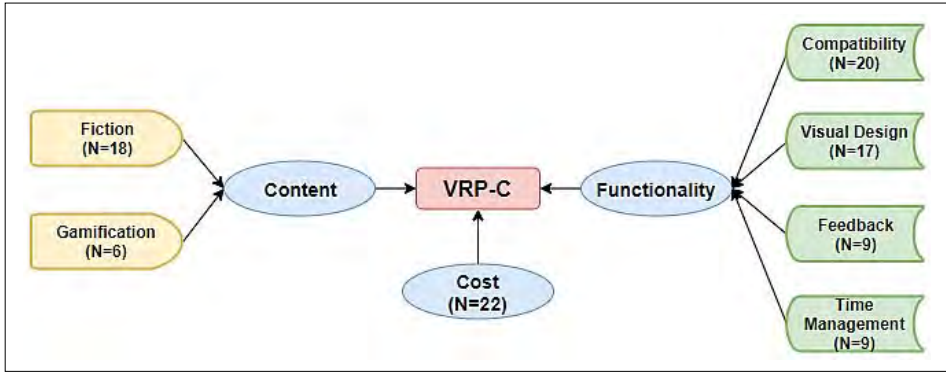


Fig. 6. The contributions of VRP-C to robotics programming teaching

content analysis performed on the data, we determined that VRP-C will contribute to robotics programming teaching within the framework of the themes content, functionality, and cost. Analysis results are visualized in Fig. 6.

3.2.1. Content

The pre-service CS teachers evaluated the features of the modules in VRP-C for robotics programming teaching within the framework of fiction and gamification. The findings and participants' expressions about robotics programming content in VRP-C are presented below.

3.2.1.1. Fiction

- There are many and various activities for robotics programming in VRP-C.

“Since there are many activities and fictions in the virtual environment, it would be more logical to prefer the VRP-C (Eda). It is possible to do many tasks for robotics programming in a virtual environment through the various challenges in VRP-C (Meva).”

- The programming activities in the VRP-C environment are designed realistically.

“According to my opinion, the fictions' closeness to reality and coherence of coordinates are the advantages of VRP-C (Feyza). Fictions in VRP-C allows us to do everything we can do with a real robot using a virtual robot (Hilal).”

3.2.1.2. Gamification

- Gamification elements in VRP-C play a role in encouraging students to learn robotics programming.

“The progression of programming teaching as fiction and the use of reinforcements such as badges in this process are other advantages

of VRP-C. Because when you advance for a purpose, it gains a game atmosphere and computational thinking develops (Hatice). Fictions and badges are encouraging situations. I think this is an important advantage (İrem).”

3.2.2. Functionality

The pre-service CS teachers evaluated the functionality of VRP-C for robotics programming teaching in the categories of compatibility, visual design, feedback, and time management. The findings and participants’ expressions about the functionality of the VRP-C are presented below.

3.2.2.1. Compatibility

- The virtual EV3 robot in VRP-C is compatible with the physical EV3 robot.

“The features in the virtual EV3 robot are generally compatible with the physical robot (Songül).”

- The virtual EV3 robot used in VRP-C works better than the physical robot. Commands and coordinates are detected by the robot more accurately.

“When I compare it with the physical robot, the virtual EV3 robot can adequately detect the given coordinates (Feyza).”

- We can do robotics programming activities and more that can be done with physical EV3 robots by using virtual EV3 robots in VRP-C.

“When I compared the architectural features of the virtual EV3 robot with the physical EV3, I observed that there was not much difference. We can perform all the functions in the physical robot with the virtual EV3 (Gökhan).”

- EV3 robot used in the VRP-C is not affected by the limitations caused by environmental factors such as light, sound, ambient colors. Thus, the results of the created algorithms can be observed without error.

“The physical robot may not work as well as the virtual EV3. Because external factors such as light, reflection, and sound can falsify the sensors (İrem).”

3.2.2.2. Visual design

- VRP-C environment is more colorful, fun, and remarkable for students than robotics programming teaching activities in which physical robots are used.

“The virtual EV3 robot in VRP-C is more colorful and fun in terms of design, but the physical EV3 is not fully met with these expectations (Buse). I think virtual EV3 is more remarkable than the physical one in terms of visual design (Büşra).”

- Environments that are difficult to create in actual life for robotics programming teaching can be designed with the tools in VRP-C.

“We may not have all materials in terms of visual design. We can design and apply whatever we want in the Builder interface of VRP-C (Nuriye). If we compare the virtual EV3 robot in VRP-C with the physical EV3, I think the virtual environment is more advantageous. Because the environments we will create in real life are more limited than the virtual environment (Sibel).”

3.2.2.3. Feedback

- The results of the codes and algorithms created in VRP-C can be observed quickly.

“I think the virtual robot is more functional than the physical EV3 robot because it can perform faster. Also, we can see the results of the codes we write immediately (Oğuz).”

- Codes written in VRP-C can be viewed step by step, and errors can be detected quickly.

“It is an advantage that we see our commands step by step through the simulation. This allows us to easily detect if there is an error in the program or algorithm (İrem).”

3.2.2.4. Time management

- Many students can connect to the VRP-C environment and do programming at the same time through the internet.

“Having only one robot will be a waste of time. Waiting for all students in a class to practice with just one robot will be a waste of time. With the virtual robot, it is possible to avoid all these concerns. For example, in the computer lab, all students can connect to the VRP-C, so they have a chance to try (Betül).”

3.2.3. Cost

Pre-service CS teachers stated that using VRP-C in robotics programming teaching will be more cost-effective than using physical robots.

“If we are financially insufficient, we can eliminate the disadvantages by using the VRP-C (Gisu-Sanem). We may not always have a physical EV3 robot because it is an expensive device. For this reason, simulators like VRP-C will be more economic for us (Metin). The most important advantage of VRP-C is its low cost. In this way, its use

for robotics programming in courses further increased. Since getting a physical robot with all its sensors or creating a project environment will be time-consuming and costly, VRP-C offers us many advantages in robotics programming teaching (Nuriye)."

4. Discussion

This study evaluated the usability of VRP-C for robotics programming teaching in schools via robotics programming training given to the pre-service CS teachers. Results indicated that VRP-C is compatible with the curriculum used for programming education in schools in terms of content (modules and subjects). Therefore, pre-service CS teachers stated they can use VRP-C in their robotics programming courses. Research indicated that VRP-C can be used for robotics programming teaching likewise the views of pre-service CS teachers (Çakıroğlu and Kiliç 2020; Liu *et al.*, 2013).

Teachers spend most of their time helping students with robot communication, mechanics, and classroom organization when they use physical robots for robotics programming teaching (Liu *et al.*, 2013). It takes time to design robots for a specific purpose in training conducted with physical robots. Also, the mechanical and electronic problems encountered during the process make it difficult for students to focus on programming (Alimisis, 2012). Since EVRs are easily modified, they ensure efficient use of time (Flot *et al.*, 2012), creating learning activities easily (Liu *et al.*, 2013), and receiving quick feedback from students about problems they encountered (Liu *et al.*, 2013). EVRs, like in VRP-C, provide students with quick feedback for errors they encounter while coding. So, they can stop the application and fix the errors without delay (Liu *et al.*, 2013). Thus, we can say that using VRP-C in robotics programming training will be useful for the effective use of course time.

Alimisis (2019) emphasized that it is essential to offer a curriculum that they can use in their future courses for robotics programming teaching to the pre-service teachers. In this study, pre-service CS teachers stated that designing the activities in VRP-C considering the individual characteristics of the students and relation to daily life will contribute to the concretization of robotics programming subjects. Activities designed for daily life problems using ERs also improve students' critical thinking and computational thinking skills (Amatzidou and Demetriadis, 2016; Chen *et al.*, 2017). In the robotic programming course included in the CS curriculum in Turkey, mechanical and electromechanical components (action components such as robot arm, motor and wheel), electronic components (sensors such as ultrasonic, color, touch and gyro), block-based and text-based programming concepts are included (Ministry of National Education [MoNE], 2021). In VRP-C, there are activities categorized under the above-mentioned topics. Activities are structured as basic activities to teach the use of components and more complex activities to teach the use of different components and programming structures (decision structures, loops, operators, etc.) together from easy to difficult.

VRP-C gives badges to the students for each challenge they complete successfully with the gamification elements it includes. Students can see the badges they won at the activities over the CS-STEM network. In addition, teachers can follow the badges earned by the students and their progress by registering the students in the groups they have created over this network. Badges show the progress level of the students and encourage them to learn (Zichermann and Cunningham, 2011). In this study, pre-service CS teachers stated the badges are an incentive element to complete the challenges. By the way, they pointed out the usefulness of badges as auto-reinforces.

Teachers face various difficulties in using appropriate media, planning the course, and preparing learning activities according to student level in courses that are newly added to curricula such as robotics programming (Karacaoğlu and Acar, 2010). VRP-C is designed based on constructionism in the manner students can progress individually and the teacher can guide them (Witherson *et al.*, 2018). Graphical and icon-based coding interfaces used in VRP-C can help students to focus on programming by keeping them from syntax errors (Kelleher and Pausch, 2005). As pre-service CS teachers stated, the ordering of challenges in VRP-C from simple to complex and using a graphical and icon-based coding interface can provide teachers with a curriculum suitable for individual progress of students and enable coding easy for students.

The perceived usefulness and ease of use towards technology also affect the individual's attitude towards using it (Dastjerdi, 2016). At the end of the training, we determined that pre-service CS teachers developed a positive attitude towards the use of VRP-C for robotics programming teaching. They think that students' interests and curiosity will increase with the features of VRP-C. They stated that the visual design of the virtual robot in VRP-C is impressive, and students will also develop a positive attitude due to it is designed compatible with the physical robot. The students stated that the components in the Lego Mindstorms EV3 simulation integrated into VRP-C have similar visual properties with the physical EV3, the properties of the components (speed, rounds, seconds, angle, etc.) are used similarly and the virtual robot provides appropriate outputs for coding as it is not affected by environmental factors. In addition, they stated that the activities in VRP-C have rich visual designs and that the activity problems are fictionalized in relation to daily life. Videos in VRP-C that link the challenges with daily life and offer clues for solutions provide additional resources for the teacher to explain the challenges to students at different levels (Witherson *et al.*, 2018).

Individuals' intention to use technology is determined by their attitudes and perceived usefulness (Dastjerdi, 2016; Lee *et al.*, 2007). Witherson *et al.* (2018) observed that students who do robotics coding with VRP-C and work on complex activities by spending more time on VRP-C have higher interest and motivation. The modules in VRP-C can contribute to the development of positive attitudes for students by emphasizing the rich three-dimensional visual design and its relationship with daily life (Witherspoon *et al.*, 2017). The low cost of VRP-C increases the intention of teachers to use this environment in their courses. The extensive use of robotics teaching materials in schools depends on their cost and easy procurement (Sipitakiat and Blikstein, 2010; Witherspoon *et al.*, 2018). EVRs are preferred more due to reasons such as the cost of

physical robotics kits and their inadequacy to apply to crowded groups. (Berland and Wilensky 2015; Flot *et al.*, 2012; Shoop *et al.*, 2016). Due to the large number of students in classrooms and the lack of sufficient resources in schools, we think it would be preferable to use VRP-C for robotics programming teaching in the courses.

5. Conclusion

In this study, we trained twenty-eight pre-service CS teachers studying at fifteen different universities in Turkey for robotics programming using VRP-C. Due to the programming challenges are arranged from simple to complex and it has a graphical coding interface, participants stated that VRP-C is easy to use. We determined that VRP-C can facilitate robotics programming teaching due to the compatibility of its modules and subjects with robotics programming curriculums in schools, the enabling teachers to use time more efficiently and providing instant feedback on coding outputs, the encouraging students to learn through gamification elements, and the editing its challenges concerning daily life. Pre-service CS teachers stated that the rich visual design of the VRP-C environment and the compatibility of the virtual robot in the VRP-C with the physical EV3 will increase the interest and curiosity of the students. They further emphasized that the use of VRP-C will contribute to the development of a positive attitude towards robotics programming for students. They intend to use VRP-C in their courses because the cost of VRP-C is lower than physical educational robots and students can practice individually by using VRP-C.

When the usability of VRP-C was evaluated in the context of TAM, pre-service CS teachers perceived VRP-C would be useful and easy for robotics programming teaching. Therefore, they developed a positive attitude towards VRP-C and intended to use it in their future courses. Also, it was seen that the Acceptance Scale of Lego Robotics Instructional Practices, which was developed to evaluate the acceptability of the physical Lego Mindstorm EV3 robots, can also be used to evaluate the acceptability of the VRP-C, which includes a virtual version of the EV3. In future research, VRP-C can be presented to CS teachers to test the acceptance results revealed by this research, and their perspectives on using VRP-C for robotics programming teaching can be evaluated with the Acceptance Scale of Lego Robotics Instructional Practices. So, the future response of the current attitude and intention expressed by TAM can be compared.

The COVID-19 pandemic, which has emerged since the beginning of 2020 and has been effective worldwide, made it necessary to carry out educational processes with distance education. The results of this research revealed that VRP-C is an alternative learning media that can be used for robotics programming teaching. In future research, the effectiveness of VRP-C can be examined in robotics programming teaching that will be conducted through distance education.

References

- Ackermann, E. (2001). Piaget's constructivism, What's the difference. *Future of Learning Group Publication*, 5(3), 438.
- Alimisis, D. (2012). Robotics in education and education in robotics: Shifting focus from technology to pedagogy. In: D. Obdržálek (Ed.), *Proceedings of the 3rd International Conference on Robotics in Education*, RiE2012 (pp. 7–14). Charles University in Prague, Faculty of Mathematics and Physics.
- Alimisis, D. (2013). Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, 6(1), 63–71.
- Alimisis, D. (2019). Teacher training in educational robotics: The ROBOESL Project paradigm. *Technology, Knowledge and Learning*, 24(2), 279–290.
- Alimisis, D., Kynigos, C. (2009). Constructionism and robotics in education. In: D. Alimisis (Ed.), *Teacher Education on robotic-Enhanced Constructivist Pedagogical Methods* (pp. 11–26). Athens, School of Pedagogical and Technological Education (ASPETE).
- Atmatzidou, S., Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670.
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education*, 58(3), 978–988.
- Berland, M., Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, 24(5), 628–647.
- Bunchball (2010). Gamification 101: An introduction to the use of game dynamics to influence behavior. <http://www.bunchball.com/sites/default/files/downloads/gamification101.pdf>
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162–175.
- cs2n.org (2020). CS-STEM network. <https://www.cs2n.org>
- Çakıroğlu, Ü., Kiliç, S. (2020). Assessing teachers' PCK to teach computational thinking via robotic programming. *Interactive Learning Environments*, 1–18.
- Çukurbaşı, B., Konokman, G.Y., Güler, B., Kartal, S.E. (2018). Developing the acceptance scale of lego robotics instructional practices: Validity and reliability studies. *Bartın University Journal of Faculty of Education*, 7(1), 191–214.
- Dastjerdi, N. B. (2016). Factors affecting ICT adoption among distance education students based on the technology acceptance model – A case study at a distance education university in Iran. *International Education Studies*, 9(2), 73–80.
- Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 318–339.
- Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson (Ed.), In: *Society for Information Technology and Teacher Education International Conference* (pp. 4006–4014). Waynesville, Association for the Advancement of Computing in Education.
- Eguchi, A. (2016). Educational robotics as a learning tool for promoting rich environments for active learning (REALs). In: *Human-computer Interaction: Concepts, Methodologies, Tools, and Applications* (pp. 740–767). IGI Global.
- Flot, J., Schunn, C., Lui, A., Shoop, R. (2012). Learning how to program via robot simulation. *Robot-Congers*, 37, 68–70.
- Groh, F. (2012). Gamification: State of the art definition and utilization. In: *Proceedings of the 4th Seminar on Research Trends in Media Informatics* (pp. 39–46), Institute of Media Informatics, Ulm University.
- Grover, S., Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Higashi, R. M., Schunn, C. D., Flot, J. B. (2017). Different underlying motivations and abilities predict student versus teacher persistence in an online course. *Educational Technology Research and Development*, 65(6), 1471–1493.
- Kafai, Y.B., Resnick, M. (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a digital World*. Routledge.
- Karacaoğlu, Y., Acar, Y. (2010). The issues that teachers encounter during application of new curricula. *Van Yuzuncu Yil University Journal of Education*, 7(1), 45–58.
- Karim, M.E., Lemaignan, S., Mondada, F. (2015). A review: Can robots reshape K-12 STEM education?, In: *Advanced robotics and its social impacts: 2015 international workshop* (pp. 1–8). Lyon, France, Institute of Electrical and Electronics Engineers.

- Kelleher, C., Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83–137.
- Kert, S.B. (2018). Introduction to computer science education. In: Y. Gulbahar (Ed.), *From computational thinking to programming* (pp. 1–20). Ankara, Pegem Akademi.
- Landis, J.R., Koch, G.G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33, 159–174.
- Lee, H.Y., Qu, H., Kim, Y.S. (2007). A study of the impact of personal innovativeness on online travel shopping behavior – A case study of Korean travelers. *Tourism Management*, 28(3), 886–897.
- Liu, A.S., Schunn, C.D., Flot, J., Shoop, R. (2013). The role of physicality in rich programming environments. *Computer Science Education*, 23(4), 315–331.
- Liu, E.Z.F., Lin, C.H., Chang, C.S. (2010). Student satisfaction and self-efficacy in a cooperative robotics course. *Social Behavior and Personality: An international Journal*, 38(8), 1135–1146.
- Mayer, R.E. (2008). Applying the science of learning: Evidence-based principles for the design of multimedia instruction. *The American Psychologist*, 63(8), 760–769.
- MoNE (2021). Computer science course curriculum. <https://mufredat.meb.gov.tr/Dosyalar/2018120203611364-BILGISAYAR%20BILIMI%20DERSI%20OGRETIM%20PROGRAMI.pdf>
- Papert, S., Harel, I. (1991). Situating constructionism. *Constructionism*, 36(2), 1–11.
- Shoop, R., Flot, J., Friez, T., Schunn, C., Witherspoon, E. (2016). Can computational thinking practices be taught in robotics classrooms? In: *International Technology and Engineering Education Conference* (pp.1–15). Carnegie Mellon Robotics Academy, University of Pittsburgh.
- Sipitakiat, A., Blikstein, P. (2010). Robotics and environmental sensing for low-income populations: Design principles, impact, technology, and results. In: *Proceedings of the 9th International Conference of the Learning Sciences* (pp. 447–448). Chicago, International Society of the Learning Sciences.
- Touretzky, D.S., Marghitu, D., Ludi, S., Bernstein, D., Ni, L. (2013). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 609–614). ACM.
- Üçgül M (2013). History and educational potential of Lego Mindstorms NXT. *Mersin University Journal of the Faculty of Education*, 9(2), 127–137
- Üçgül, M. (2018). Educational robots and computational thinking. In: Y. Gulbahar (Ed.), *From Computational Thinking to Programming* (pp. 295–317). Ankara, Pegem Akademi.
- Witherspoon, E.B., Schunn, C.D. (2019). Teachers' goals predict computational thinking gains in robotics. *Information and Learning Sciences*, 120(5/6), 308–326.
- Witherspoon, E.B., Higashi, R.M., Schunn, C.D., Baehr, E.C., Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education*, 18(1), 1–20.
- Witherspoon, E.B., Schunn, C.D., Higashi, R.M., Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning*, 34(2), 115–128.
- Zichermann, G., Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media Inc., Sebastopol, CA.

S. Kılıç is a Ph.D and an instructor at Ordu University. His academic specialty is instructional technologies, distance learning, teacher training, programming and robotics education, and problem-solving strategies.

S. Gökoğlu is an assistant professor at the Department of Computer Technology and Information Systems at Bartın University, Turkey. He received a Ph.D. in computer education and instructional technologies from the Karadeniz Technical University, with a study of behavioral skills training with virtual reality. His research interests include programming education, student engagement, teacher training, and virtual reality.