Ahi Evran Üniversitesi Sosyal Bilimler Enstitüsü Dergisi (AEÜSBED)
2024, Cilt 10, Sayı 2, Sayfa 434-452

Use of the Alice Visual Programming Environment in Programming Education: A Descriptive Case Study

Ceren BAŞTEMUR KAYA
Nevsehir Haci Bektas Veli University
ceren@nevsehir.edu.tr
ORCID ID: 0000-0002-0091-3606

Hasan ÇAKIR
Gazi University
hasanc@gazi.edu.tr
ORCID ID: 0000-0002-4499-9712

Araştırma Makalesi		DOI: 10.31592/aeusbed.1464694
Geliş Tarihi: 03.04.2024	Revize Tarihi: 10.07.2024	Kabul Tarihi: 25.07.2024

Atıf Bilgisi
Baştemur Kaya, C. ve Çakır, H. (2024). Use of the Alice visual programming environment in programming education: A descriptive case study. *Ahi Evran Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 10*(1), 434-452.

ABSTRACT

The process of learning a programming language is seen as complicated, especially for those who first start to learn programming language. For this reason, visual programming tools have begun to be used in educational environments. Visual programming tools purpose to concretize abstract programming language concepts. Alice program is one of the visual programming tools. The aim of this study is to determine the effects of the Alice program on programming learning in the education environment. The study group was 18 freshman students enrolled in the Object Based Programming I course at the vocational school computer technology department. The study lasted 8 weeks, with 4 hours each week. The teaching was realized on Gagne's nine events of instruction model. At the end of the teaching process, two focus group interviews were conducted with the students to determine the obstructing and facilitating factors in using the Alice program. This study was utilized a descriptive case study. The data were analyzed by using the content analysis. When the results were examined; It was seen that the Alice program makes learning the basic code concepts easy and understanding programming logic, and increases the desire to learn programming. However, it was stated that the practicality of writing code could not be improved because code could not be written in the Alice program. It is thought that employing the Alice program is beneficial for enhancing comprehension of programming logic.
Keywords: Alice, programming teaching, visual programming environment.

Alice Görsel Programlama Ortamının Programlama Eğitiminde Kullanımı: Tanımlayıcı Durum Çalışması

ÖZ

Programlama dili öğrenmek, özellikle programlama dilini ilk öğrenmeye başlayanlar için zorlu bir süreç olarak görülmektedir. Bu nedenle eğitim ortamlarında görsel programlama araçları kullanılmaya başlanmıştır. Görsel programlama araçları soyut programlama dili kavramlarını somutlaştırmayı amaçlamaktadır. Alice programı görsel programlama araçlarından biridir. Bu çalışmanın amacı Alice programının eğitim ortamında programlama öğrenimine etkisini incelemektir. Çalışma grubu meslek yüksekokulu bilgisayar teknolojileri bölümü Nesne Tabanlı Programlama I dersine kayıtlı olan18 birinci sınıf öğrencisidir. Öğretim Gagne'nin dokuz aşamalı öğretim modeline göre gerçekleştirilmiştir. Araştırma haftada 4 saat olmak üzere toplam 8 hafta sürmüştür. Öğretim süreci sonunda öğrencilerle Alice programının kullanımını engelleyen ve kolaylaştıran faktörlerin belirlenmesi amacıyla iki odak grup görüşmesi yapılmıştır. Tanımlayıcı durum çalışması kullanılmıştır. İçerik analizi ile toplanan veriler analiz edilmiştir. Sonuçlar incelendiğinde; Alice programının temel kod kavramların öğrenilmesini kolaylaştırdığı ve programlama mantığının anlaşılmasını sağladığı, programlama öğrenme isteğini artırdığı görülmüştür. Ancak Alice programında kod yazılamadığı ve bu nedenle kod yazma pratikliğinin kazanılamadığı belirtilmiştir. Programlama öğretim ortamlarında Alice programı kullanımının programlama mantığının daha iyi anlaşılması açısından faydalı olacağı düşünülmektedir.
Anahtar Kelimeler: Alice, programlama öğretimi, görsel programlama ortamı.

Introduction

The production of new products in the information sector has gained importance with the development of technology. Especially the use of different technologies has come to the fore (Baştemur Kaya and Çakır, 2018). This situation has increased the importance of programming.

It is important to update educational processes in order to adapt to technological developments (Özdemir, 2023). Especially programming education is of great importance. Many metacognitive abilities of individuals such as problem solving, analytical thinking, creativity, and computational thinking skills develop with programming (Ar and Korkmaz, 2023; Denner, Werner and Ortiz, 2012; Feurzeig, Papert and Lawler, 2011; Havenga, Breed and Mentz 2013; Saeli, Perrenet, Jochems and Zwaneveld, 2011). For this reason, programming courses have been added to many levels of education (Fessakis, Gouli and Mavroudi, 2013; Lau and Yuen, 2011; Yıldız Durak and Güyer, 2022). However, especially beginner students are seen learning programming as a complex process (Ford, 2016; Gomes and Mendes, 2007; Topalli and Cagiltay, 2018).

Students can't focus on the general structure of programming and understand abstract concepts (Biju, 2013; Topalli and Cagiltay, 2018; Yıldız Durak and Güyer, 2022). These problems are among the difficulties experienced in teaching programming. In programming languages, although the syntax is different, the basic concepts are similar (Baştemur Kaya and Çakır, 2018; Biju, 2013; Ford, 2016; Gomes and Mendes, 2007; Topalli and Cagiltay, 2018). However, students experience difficulties in learning every new programming language. Although the structure of programming languages has changed over the years, there have not been many changes in teaching programming. Students state that they understand when the teacher explains but in different problems they can not convey what they have learned (Garner, 2003). One of the departments where programming courses are given intensively is the computer technologies department in vocational school. The computer technologies department covers a significant part of the trained manpower of Turkey's information and communication sector (Baştemur Kaya, 2018). However, academic success is seen to be low in vocational schools also. When low academic success' reason is looked at, it has been seen that traditional methods such as demonstration, direct instruction, etc. are used in lessons (Binici and Necdet, 2004; Şahin and Fındık, 2008; Ünal and Çakır, 2016).

It is important to integrate technology into education to increase academic success (Alghamdi and Holland, 2020; Li et al., 2014; Rushkoff, 2010). In programming education, it is useful to use helpful tools to understand the logic of programming, increase programming skills, and understand the meaning of codes and concepts (Bosse and Gerosa, 2017; Fields, Giang and Kafai, 2014; Hill, 2015; Hsu, Chang and Hung, 2018; Resnick et al., 2009; Yildiz Durak, 2020). In programming education, visual programming environments are one of the helpful tools.

Visual programming environments have been developed to facilitate difficult processes in programming and to understand programming logic (Cooper, Dann, Pausch and Pausch, 2000; Hsu et al., 2018; Kölling, 2010; Kölling, Quig, Patterson and Rosenberg, 2003). Alice program, one of the visual programming environments, is developed to facilitate understanding of programming logic (Cooper et al., 2000; Hayat, Al-Shukaili and Sultan, 2017). The Alice program is compatible with object-oriented programming languages. It is three dimensional. It works according to drag and drop logic. Research supports that the Alice program helps understand programming logic and increases interest in programming (Baştemur Kaya and Çakır, 2018; Cliburn, 2008; Cooper et al., 2000; Hayat et al., 2017; Wang, Mei, Lin, Chiu and Lin, 2009). However, research, on how the Alice program should be used in learning environments for more effective, efficient, and interesting learning, is limited. Especially student opinions are the most important data that determine how and when helpful programs should be used, during the learning process.

The aim of this study is to examine the opinions of freshmen students studying in the vocational school computer technologies department in order to ensure more effective, efficient, and interesting use of the Alice program in learning environments in teaching the Java programming language, one of the object-oriented programming languages. The study was sought to address the question, "What are the students' opinions on the use of the Alice program in programming learning?" in the scope of this purpose.

Visual Programming Environments

Visuality can facilitate intelligibility in programming. Two-dimensional representations such as flowcharts have helped to understand programming for years. Visualization plays an important role in preventing misconceptions.

There are many visualization tools in programming language teaching. These visualization tools have different features. Some visualization tools are two-dimensional, some are three-dimensional. While some visualization tools allow writing code, some are aimed at helping you understand what the codes do. Regardless of its features, the purpose of visualization is to provide a better understanding of programming logic. Visual programming environments are employed to address the challenges encountered in learning programming. The Alice program is commonly utilized for instructing Java and analogous object-oriented programming languages (Hayat et al., 2017; Papadakis, Kalogiannakis, Orfanakis and Zaranis, 2014).

The Alice Program

The programmers through the utilization of the Alice program acquire knowledge of general programming structures and also create their virtual worlds (Cooper et al., 2000). The Alice program is viewed as a tool that aids users in enhancing their problem-solving and algorithmic thinking abilities, as it guides users to solve a problem while they create an animation (Cooper, Dann, and Pausch, 2003). The users are able to observe the coordination between code and images, thereby establishing a relationship between the two (Cooper et al., 2000; Wang et al., 2009). The users can readily comprehend the functionalities of structures (such as functions, loops, events, etc.) across various programming languages. They observe how fundamental concepts are utilized and their functionalities. It enables the visualization of what the code accomplishes. Therefore, programming no longer requires users to envision abstract concepts and programming structures. Through active participation in the programming process, they can grasp the functions of structures such as functions, loops, events, etc. across different programming languages easily. This way, the Alice program enhances users' programming readiness level for different programming languages and also boosts their interest and engagement in programming (Cooper et al., 2003; Howard, Evans, Courte and Bishop-Clark, 2006). The Alice program is well-suited for highlighting Java and its related concepts (Papadakis et al., 2014). The program offers both Alice and Java language options. When the Java programming language is selected in the code language opinion, variables and some codes can be inserted and observed in the Java programming language. Additionally, animations created using the Alice program can be compiled using the NetBeans programming language compiler.

There are many studies investigating the effect of the Alice program on programming teaching. The studies have focused on the effect of the Alice program on academic success (Al-Linjawi and Al-Nuaim, 2010; Baştemur Kaya, 2018; Biju, 2013; Moskal, Lurie and Cooper, 2004; Price, 2013; Schultz, 2011; Solmaz, 2014; Sykes, 2007; Wang et al., 2009; Zhang, Liu, Pablos and She, 2014), problem solving skills (Baştemur Kaya, 2018; Solmaz, 2014), motivation (Baştemur Kaya, 2018; Wang et al., 2009; Zhang et al., 2014), and the programming process (Baştemur Kaya, 2018; Baştemur Kaya and Çakır, 2018; Bishop-Clark, Courte, Evans and Howard, 2007; Cliburn, 2008; Daly, 2011; Hayat et al., 2017; Howard et al.,

2006; Liu, Lin, Hasson and Barnett, 2011; Moskal et al., 2004; Solmaz, 2014; Sykes, 2007; Werner et al., 2009) in programming teaching.

In the studies investigating the effect of the Alice program on the programming process, it was seen that data is generally collected from surveys and interviews (Baştemur Kaya, 2018; Baştemur Kaya and Çakır, 2018; Bishop-Clark et al., 2007; Cliburn, 2008; Daly, 2011; Hayat et al., 2017; Howard et al., 2006; Liu et al., 2011; Solmaz, 2014; Sykes, 2007; Werner et al., 2009). In the studies, most researchers and students found the Alice program easy to use (Baştemur Kaya 2018; Baştemur Kaya and Çakır, 2018; Bishop-Clark et al., 2007; Cooper et al., 2000, 2003; Hayat et al., 2017; Howard et al., 2006; Solmaz, 2014; Sykes, 2007; Werner et al., 2009), fun (Baştemur Kaya 2018; Baştemur Kaya and Çakır, 2018; Cooper et al., 2000, 2003; Hayat et al., 2017; Solmaz, 2014; Werner et al., 2009), and helpful in understanding the logic of programming (Baştemur Kaya 2018; Baştemur Kaya and Çakır, 2018; Bishop-Clark et al., 2007; Cliburn, 2008; Cooper et al., 2000, 2003; Daly, 2011; Howard et al., 2006; Liu et al., 2011; Moskal et al., 2004; Solmaz, 2014; Sykes, 2007; Wang et al., 2009). Students working with the Alice program stated that their self-confidence in programming increased (Bishop-Clark et al., 2007; Cooper et al., 2000, 2003; Daly, 2011; Liu et al., 2011; Moskal et al., 2004). It has been stated that a solid programming foundation can be created with the Alice program (Aktunc, 2013; Kayabaşı, 2016; Powers, Ecott and Hirshfield, 2007). There are also studies indicating that motivation and interest in programming courses (Bishop-Clark et al., 2007; Cooper et al., 2000, 2003; Hayat et al., 2017; Wang et al., 2009).

However, the high system requirements of the Alice program have been criticized (Baştemur Kaya, 2018; Cliburn, 2008; Solmaz, 2014). In addition, the inability to perform advanced programming with the Alice program was stated as a negative situation (Baştemur Kaya, 2018; Powers et al., 2007).

Method

Explanations were made regarding the research model, working group, and data collection tools in this section. The implementation process and data analysis were explained in detail. Additionally, information regarding research ethics was presented.

Research Model

The study utilized one of the qualitative research models, the descriptive case study. In the case study; one or more cases are examined in depth using data collection tools (Creswell and Poth, 2016). According to Creswell and Poth (2016), cases and themes related to cases are determined by collecting data systematically. One type of case study design is the descriptive case study, in which the case is presented in great detail and with realism (Merriam, 2013). The present study employed a descriptive case study methodology with the purpose of thoroughly examining student viewpoints in a realistic manner. This approach was chosen to examine the effective, efficient, and appropriate use of the Alice program in programming language teaching.

Pilot Study Process

A pilot study was conducted before the implementation process in order to identify the deficiencies that may be experienced in the implementation process and to identify the problems that may be experienced in the programming learning environment designed with the Alice program. The pilot study took five weeks. It was applied four hours a week and it was 20 hours in total. The process was designed according to Gagne's nine-stage teaching model as in the implementation process. It covered the topics in the first five weeks of the implementation process. Examples and applications from the implementation process were used. As a result of the pilot study process, it was completed the final version of the implementation process and the

programming learning environment designed with the Alice program (Baştemur Kaya and Çakır, 2018).

Implementation Process

The implementation process was carried out in the spring semester of the 2015-2016 academic year. It took 8 weeks with 4 hours per week. Its scope was the Object Oriented Programming I course. It was studied with 34 freshmen students. The computer technologies department curriculum includes Object Oriented Programming I course as a compulsory course. The course curriculum covers the Java programming language's basic programming concepts, including data types and variables, operators, control structures, loops, arrays, methods, class structures, and inheritance. During the course, students made examples and applications both with the Alice program and the Java programming language. The students did the examples together with the course instructor. The students were anticipated to execute the applications given by the course instructor under the course instructor's guidance as needed. Teaching was carried out using Gagne's nine-stage teaching model. The weekly implementation plan is presented in Table 1. After the implementation process, two focus group interviews were conducted with 18 students selected among the 34 students.

Table 1
Weekly Implementation Plan

Weeks	Topics	Used Programs
Week 1	Week 1 topic (Data types and variables, operators)	Alice + NetBeans
Week2	Week 2 topic (Control structures)	Alice + NetBeans
Week3	Week 3 topic (Loops)	Alice + NetBeans
Week4	Week 4 topic (Series)	Alice + NetBeans
Week5	Week 5 topic (Methods)	Alice + NetBeans
Week6	Week 6 topic (Methods)	Alice + NetBeans
Week7	Week 7 topic (Class structures, inheritance and objectify)	Alice + NetBeans
Week8	Week 8 topic (Class structures, inheritance and objectify)	Alice + NetBeans

In the first half of the course, the topic of each week was first explained and demonstrated by the course instructor with the Alice program, and examples were made. After the examples were done, students were asked to do applications with the Alice program. In the other half of the course, after the course instructor demonstrated examples about using the Java programming language, the students were asked to do applications.

Study Group

The research was conducted during the spring semester of 2015-2016 the academic year. The study group consisted of freshman students enrolled in the Object Oriented Programming I course at the computer technologies department of the vocational school. 34 students participating in the implementation process were selected by using convenience sampling method. The descriptive analysis result, including students' gender information, is given in Table 2.

Table 2
Gender Information of Students

Students	n	%
Female	13	38.24
Male	21	61.76
Total	34	100

Vocational school computer technologies department provides over 90% of the trained manpower in the information and communication sector. However, students can not acquire the

skills they need because it is not used strategies that will make students active in classes in today's vocational schools. For this reason, the selection of students from the vocational school computer technologies department was based on their significance to the software industry in this study.

18 students formed the study group among the 34 students who participated in the implementation process. Maximum diversity sampling, which is included in the purposeful sampling method, was used in the selection of 18 students. Maximum diversity sampling aims to reflect the diversity of the work group to the maximum extent (Yildirim and Şimşek, 2013). It is valuable to work with different groups in order to capture the common dimensions and basic experiences of the factor whose effect is being investigated (Patton, 2014). According to the academic calendar of the vocational school where the research was conducted, a midterm exam of the Object-Oriented Programming I course was held before the focus group interviews. Three performance levels were determined taking into account midterm exam scores (upper level: 100-67, middle level: 66-34, lower level: 33-0). Three students each were selected from each level of performance in the midterm exam for focus group interviews. Interviews were conducted with 9 students in each focus group, a total of 18 students took part in the interviews. The number of students at the upper, middle, and lower levels according to the midterm exam is given in Table 3.

Table 3
Grouping of 34 Students who Took Part in The Implementation Process according to Upper, Middle, and Lower Levels

	Upper Level	**Middle Level**	**Lower Level**	**Total**
Number of Students	9	12	13	34

Data Collection Tools

Two focus group interviews were held within the scope of the research. A semi-structured interview form was prepared to be used in the interviews. The form used in both interviews is the same.

Semi-Structured Interview Form

The researchers developed a semi-structured interview form to detection the students' opinions on the learning environment designed with the Alice program and the Alice program. Necessary arrangements were made by consulting five experts' opinions. The final version of the interview form has 9 items. The interview form includes questions about the most liked and disliked features of the program, how useful the Alice program was in understanding general programming concepts, the usefulness of using the Alice program in programming language learning, how effective it would be to continue using it in the programming course, whether the program would continue to be used, etc. Two groups of 9 students were formed from the students. Two focus group interviews were held involving a total of 18 students.

Data Analysis

The results of interviews with students were analyzed by using the content analysis. A voice recorder was used for recording the interviews. Interview data recorded were transcribed for coding. Coding was done again by a coder other than the researchers to provide reliability. For calculating the consensus between the coders, the formula (consensus/(consensus + disagreement)* 100) (Miles and Huberman, 1994) was used. The reliability between the coders was found to be 91.66%.

Research Ethics

In this study, Nevsehir Hacı Bektas Veli University Ethics Committee dated 12.02.2016, decision number 84902927, decision no: 2016.02.03 ethics committee approval was obtained.

Findings

In this section, the findings of the data analyzed by content analysis are presented. The findings are presented by grouping.

Student Opinions on the Use of the Alice Program in Programming Teaching

The data were analyzed by using the content analysis. Four themes appeared from the result of the analysis. The themes, categories and codes are presented in tables 4, 5, 6 and 7. In the tables, frequency indicates the number of participants expressing the code.

Benefits of the Alice Program in Learning Programming Language Theme

This theme includes the students' opinions about the effects of the features of the Alice program on programming and programming language learning. Three categories and ten codes were obtained in the scope of the theme. Table 4 shows that the theme of Benefits of the Alice Program in Learning Programming Language, its categories, and its codes.

Table 4
Findings on Benefits of the Alice Program in Learning Programming Language Theme

Theme	Category	Code	Frequency
Benefits of the Alice Program in Learning Programming Language	Impact on Programming	Facilitating the learning of basic code concepts	18
		Increasing the desire to learn programming	17
		Facilitating learning programming	16
		Making programming fun	13
		Increasing the desire for research and questioning in programming	11
		Helping to understand algorithm logic	1
	Visuality (Visual display of program outputs)	Increasing permanence	9
		Making coding errors easier to see	7
	Drag-drop	Preventing syntax errors	8
		Reducing the workload in coding	7

Three categories emerged as impact on programming, visuality (visual display of program outputs), drag-drop in this theme. In impact on programming category, facilitating the learning of basic code concepts was the most common code. Increasing permanence was the most common code for visuality (visual display of program outputs) category. Preventing syntax errors was he most common code in drag-drop category. The students interviewed were coded as S1, S2.... Examples of students' opinions about the most repeated codes in each category are as follows:

S3: "I understood the basics of codes and how certain patterns are used in the Alice. I transferred what I learned to Java."

S8: “I understand better what I see. It gets better permanence.”

S5: "The Alice doesn't have code errors. I drop the code on the work screen with the mouse. I don't even bother writing code. I focus on making the app."

Positive Features of the Alice Program Theme

In this theme, the students' opinions about the positive features of the Alice program in programming language learning are contained. Six codes emerged in this theme. Positive features of the Alice program theme and its codes are presented in Table 5.

Table 5
Finding on Positive Features of the Alice Program Theme

Theme	Code	Frequency
Positive Features of the Alice Program	Ease of use	18
	Working with drag-and-drop logic	17
	Visually seeing what the codes do	13
	Fun to use	9
	Ability to move objects in detail	4
	Interesting interface design	2

Ease of use and working with drag-and-drop logic were the most common codes. Examples of students' opinions about the most repeated codes in the theme are as follows:

S5: "It's nice to see the codes visually. That way, I can look at what each one does."

S10: "It was very easy to use. No coding. I loved the drag-and-drop logic."

Negative Features of the Alice Program Theme

This theme includes the students' opinions about the negative features of the Alice program in programming language learning. Two categories and five codes emerged in this theme. Negative features of the Alice program theme and its codes are presented in Table 6.

Table 6
Finding on Negative Features of the Alice Program Theme

Theme	Category	Code	Frequency
Negative Features of the Alice Program	Features of the Alice Program	Limited language support	9
		Lack of variety in objects and scenes	7
		Unable to change codes	3
	Impact on Programming	Lack of practicality to write code because it does not allow to write code	13
		Inadequate for advanced programming	11

Limited language support was the most common code in features of the Alice program category. Lack of practicality to write code because it does not allow to write code was the most common code in impact on programming category. Examples of the students' opinions in the theme are as follows:

S13: "I think it is useful in terms of logic. But when it comes to writing code, I don't think it's very useful. It has had an impact on learning, but not on development. It only helped in the first phase."

Future Use of the Alice Program Theme

This theme includes the students' opinions about the future use of the Alice program. Two codes emerged in the scope of the theme. Future use of the Alice program theme and its codes are presented in Table 7.

Table 7
Finding on Future Use of the Alice Program Theme

Theme	Code	Frequency
Future Use of the Alice Program	If I teach a programming language in the future, I use the Alice program.	17
	I would like to continue using the Alice program	13

It was seen that 17 of the students preferred the Alice program if they need to teach programming language in the future. "I would like to continue to use the Alice program in the future" code included that using the Alice program both to improve the programming and to create different animations. Examples from the students' opinion for future use of the Alice program theme is as follows:

S16: "I would like to use the Alice in the future. I improve my programming while making new stories and applications."

Conclusion, Discussion and Recommendations

In this study, the opinions of freshmen students studying in the vocational school computer technologies department were examined in order to ensure more effective, efficient, and interesting use of the Alice program in learning environments in teaching the Java programming language, one of the object-oriented programming languages. When the results of the focus group interview with 18 students were analyzed, it was seen that the students expressed on the opinion which are the benefits of the Alice program in learning programming language, the positive and negative features of the Alice program, and opinions about use of the Alice program in the future.

When the research results were examined, the students mostly stated that The Alice program facilitated the learning of basic code concepts, increased permanence, and prevented syntax errors relating to the benefits of the Alice program in learning programming language opinion. In the literature, Al-Tahat (2021), Biju (2013), Cooper et al. (2000), Cooper et al. (2003), Fasy et al. (2020), Hayat et al. (2017), Johnsgard and McDonald (2008), Rojat (2022), Schultz (2011) and Solmaz (2014) specified in their study that the Alice program helps learn basic code concepts. Al-Tahat (2021), Courte, Howard and Bishop-Clark (2006), Fasy et al. (2020) and Hayat et al. (2017) emphasized in their studies that students stated that the Alice program encouraged them to programming. Al-Linjawi and Al-Nuaim (2010), Cooper et al. (2000), Howard et al. (2006), McKenzie (2009), Powers et al. (2007), Rojat (2022), Solmaz (2014) and Zhang et al. (2014) pointed out in their study that the Alice program makes it easier to learn the complex structure of programming languages and understand the programming logic. Solmaz (2014) reported in their study that students stated that the Alice program increased permanence. Brown (2008) and Solmaz (2014) emphasized that the correct syntax was achieved by working with the drag-and-drop logic of the Alice program. Al-Tahat (2021) stated in his study that students developed positive attitudes towards programming. Al-Tahat (2021) and Fasy et al. (2020) indicated that students' interest in computers had increased. Fasy et al. (2020) emphasized in their study that students wanted to reuse the Alice program in different scenarios. Fauzan (2021) stated that computational thinking, decomposition, and abstraction programming abilities can be developed with the Alice program.

In the Alice program, codes are added using drag-and-drop logic. It can see which code gives what result visually in the animation. The effect of each code can be seen in the animation. The working logic of the codes can be better understood. Thus, users' interest in the programming course can be increased. Especially the permanence of the learned information may increase. The increase in students' interest in programming courses, which they generally describe as boring, is committing to producing original technologies through programming.

The students stated that the most positive features of the Alice program were that it was easy to use and worked with the drag-and-drop feature. In the literatüre, Cliburn (2008), Cooper et al. (2003), Fasy et al. (2020), Hayat et al. (2017), Howard et al. (2006) and Sykes (2007) stated in their studies that students found the Alice program easy and simple to use. Cooper et al. (2003) emphasized that the drag-and-drop editor was useful for beginner users. Solmaz (2014) stated that students liked the feature of adding ready-made codes.

While working with the Alice program, instead of writing code, users drag and drop the codes onto the screen. Thus, instead of dealing with syntax errors, they only care about what the codes do. With a full understanding of what the codes do, it can be developed new and different applications and programs. Thus, it can move forward with more confident steps in the information technology sector.

The students stated that the most negative features of the Alice program were that The Alice program does not allow to write code so there is lack of practicality to write code and it is inadequate for advanced programming. Brown (2008), Cliburn (2008), Garlick and Cankaya (2010), McKenzie (2009), Powers et al. (2007) and Sykes (2007) emphasized in their studies that there is a big difference between programming with the Alice program and programming with a programming language, and that the program is not suitable for solving real-world problems. Howard et al. (2006) and Werner et al. (2009) stated in their study that students did not like the software restrictions in the Alice program. Schultz (2011) pointed out that the Alice program is insufficient for programming.

Although the Alice program makes learning the programming language easier, it may not be sufficient for advanced programming. It may be more suitable for users who are new to programming. Most of the students stated that they would use the Alice program in the future. This statement can be interpreted as the Alice program benefits students in learning programming language. Considering the whole of the study, it is thought that the Alice program will be useful for users who are new to programming. Visual programs can be used in educational environments to like programming and understand the logic of programming.

Author Contribution Statement

Researchers contributed equally to this study.

Conflicts of Interest

There is no conflict of interest in this study.

Acknowledgments and/or Explanation: This study is a part of the first author's PhD thesis under the supervision of the second author.

References

Aktunc, O. (2013). A teaching methodology for introductory programming courses using Alice. *International Journal of Modern Engineering Research (IJMER), 3*(1), 350-353.

Al-Linjawi, A. A. and Al-Nuaim, H. A. (2010). Using Alice to teach novice programmers OOP concepts. *Journal of King Abdulaziz University: Science*, *148*(632), 1-20. DOI:10.4197/Sci.22-1.4.

Al-Tahat, K. (2021, December). *Alice adventures in computingland: A review*. 22nd International Arab Conference on Information Technology (ACIT), IEEE. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9677413 in 10.07.2024.

Alghamdi, J. and Holland, C. (2020). A comparative analysis of policies, strategies and programmes for information and communication technology integration in education in the Kingdom of Saudi Arabia and the republic of Ireland. *Education and Information Technologies*, *25*(6), 4721-4745. https://doi.org/10.1007/s10639-020-10169-5.

Ar, Ü. and Korkmaz, Ö. (2023). Mesleki ve teknik anadolu lisesi öğrencilerinin kodlama eğitimine dönük tutumlarının ve bilgisayarca düşünme beceri düzeyleri. *Ahi Evran Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *9*(3), 873-892. https://doi.org/10.31592/aeusbed.1350287.

Baştemur Kaya, C. (2018). *Programlama dili öğretiminde Alice programının öğrencilerin akademik başarıları, problem çözme becerisi algıları, motivasyonları ve programlamaya hazırbulunuşluk düzeyleri üzerisine etkisi*. Doktora Tezi, Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.

Baştemur Kaya, C. and Çakır, H. (2018). Programlama dili öğretiminde Alice programının kullanım sürecinin incelenmesi. *Eğitimde Nitel Araştırmalar Dergisi*, *6*(2), 187-206. DOI:10.14689/issn.2148 - 2624.1.6c2s9m.

Biju, S. M. (2013). Taking advantage of Alice to teach programming concepts. *E-Learning and Digital Media*, *10*(1), 22-29. https://doi.org/10.2304/elea.2013.10.1.22.

Binici, H. and Necdet, A. (2004). Mesleki ve teknik eğitimde arayışlar. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, *24*(3), 383-396.

Bishop-Clark, C., Courte, J., Evans, D. and Howard, E. V. (2007). A quantitative and qualitative investigation of using Alice programming to improve confidence, enjoyment and achievement among non-majors. *Journal of Educational Computing Research, 37*(2), 193-207. https://doi.org/10.2190/J8W3-74U6-Q064-12J5.

Bosse, Y. and 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. http://doi.acm.org/10.1145/3011286.3011301.

Brown, P. H. (2008). Some field experience with Alice. *Journal of Computing Sciences in Colleges, 24*(2), 213-219.

Cliburn, D. C. (2008, October). *Student opinions of Alice in CS1*. 2008 38th Annual Frontiers in Education Conference. Retrieved from https://ieeexplore.ieee.org/abstract/document/4720254/?casa_token=YdLbVr4xU6AAAAAA:HwIViHzqX5J-V83tSbSh2ndJhURsiiOSYqWeKXDN1fCWOshgT4wk2tGzOJL c5l-x-LmcQ3CjaY in 02.04.2024.

Cooper, S., Dann, W. and Pausch, R. (2000, November). *Developing algorithmic thinking with Alice.* The Proceedings of ISECON. Retrieved from https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=15c0016285c067f2f8dbbf2d7d59f9ffceb9237f in 02.04.2024.

Cooper, S., Dann, W. and Pausch, R. (2003). Using animated 3d graphics to prepare novices for CS1. *Computer Science Education*, *13*(1), 3-30. https://doi.org/10.1076/csed.13.1.3.13540.

Cooper, S., Dann, W., Pausch, R. and Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, *15*(5), 107-116.

Courte, J., Howard, E. V. and Bishop-Clark, C. (2006). Using Alice in a computer science survey course. *Information Systems Education Journal, 4*(87).

Creswell, J. W. and Poth, C. N. (2016). *Qualitative inquiry and research design: Choosing among five approaches*. Sage Publications.

Daly, T. (2011). Minimizing to maximize: An initial attempt at teaching introductory programming using Alice. *Journal of Computing Sciences in Colleges, 26*(5), 23-30.

Denner, J., Werner, L. and Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers and Education, 58*(1), 240-249. https://doi.org/10.1016/j.compedu.2011.08.006.

Fasy, B. T., Hancock, S. A., Komlos, B. Z., Kristiansen, B., Micka, S. and Theobold, A. S. (2020, June). *Bring the page to life: Engaging rural students in computer science using Alice.* The 2020 ACM Conference on Innovation and Technology in Computer Science Education. Retrieved from https://dl.acm.org/doi/abs/10.1145/3341525.3387367?casa_token=3gwoo8vi9KcAAAAA:yKxsyeJZJV66TPjYAi5rMJBe7axZJTxD07ty7fYjNAyTHk0aRTBwYmChCMPvIeXhjKMwOTsBvVS2 in 10.07.2024.

Fauzan, M. N. (2021). Computational thinking dan Alice 3: Prodi/Jurusan D4 teknik informatika politeknik pos Indonesia. *Jurnal Teknik Informatika, 13*(1), 1-4.

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*, *63*, 87-97. https://doi.org/10.1016/j.compedu.2012.11.01.

Feurzeig, W., Papert, S. A. and Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, *19*(5), 487-501. https://doi.org/10.1080/10494820903520040.

Fields, D., Giang, M. and Kafai, Y. (2014). *Programming in the wild: Patterns of computational participation in the Scratch online social networking forum*. The 9th Workshop in Primary and Secondary Computing Education. New York,NY: ACM.

Ford, J. L. (2016). *Programming for the absolute beginner*. Nelson Education.

Garlick, R. and Cankaya, E. C. (2010, June). *Using Alice in CS1: A quantitative experiment.* The Fifteenth Annual Conference on Innovation and Technology in Computer Science Education. Retrieved from https://dl.acm.org/doi/pdf/10.1145/1822090.1822138 in 02.04.2024.

Garner, S. (2003, June). *Learning resources and tools to aid novices learn programming*. Informing Science and Information Technology Education Joint conference (INSITE). Retrieved from https://proceedings.informingscience.org/IS2003Proceedings/docs/036Garne.pdf in 02.04.2024.

Gomes, A. and Mendes, A. J. (2007, June). *An environment to improve programming education*. The 2007 International Conference on Computer Systems and Technologies. Retrieved from https://dl.acm.org/doi/abs/10.1145/1330598.1330691 in 02.04.2024.

Havenga, M., Breed, B. and Mentz, E. (2013). Metacognitive and problem-solving skills to promote self-directed learning in computer programming: Teachers' experiences. *SA-eDUC*, *10*(2).

Hayat, K., Al-Shukaili, N. A. and Sultan, K. (2017). Alice in oman: A study on object-first approaches in computer science education. *Education and Information Technologies*, *22*(4), 1553-1569. DOI 10.1007/s10639-016-9499-4

Hill, C. (2015). *Programming environments for children: Creating a language that grows with you.* Master's Thesis, University of California, Santa Barbara.

Howard, E. V., Evans, D., Courte, J. and Bishop-Clark, C. (2006). *A qualitative look at Alice and pair-programming*. ISECON 2006. Retrieved from https://isedj.org/7/80/ISEDJ.7(80).Howard.pdf in 02.04.2024.

Hsu, T. C., Chang, S. C. and Hung, Y.T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, *126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004.

Johnsgard, K. and McDonald, J. (2008, April). *Using Alice in overview courses to improve success rates in programming I*. 2008 21st Conference on Software Engineering Education and Training. Retrieved from https://ieeexplore.ieee.org/abstract/document/4556958?casa_token=RQWOYm5jO8QAAAAA:ojgOkfjB5ZoX0TnpqbL2LoyIszaJK0tlO7nFHWt2Rdcg4eGh_qWX4d_jYDyk8TLfIui1vde7a8 in 02.04.2024.

Kayabaşı, E. (2016). *Öğretmen adaylarının Alice deneyimi: 3B ortamda programlama.* Yüksek Lisans Tezi, Uludağ Üniversitesi, Eğitim Bilimleri Enstitüsü, Bursa.

Kölling, M. (2010). The greenfoot programming environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 1-21. https://doi.org/10.1145/1868358.1868361.

Kölling, M., Quig, B., Patterson, A. and Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Computer Science Education*, *13*(4), 249-268. https://doi.org/10.1076/csed.13.4.249.17496.

Lau, W. W. and Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers and Education*, *57*(1), 1202-1213. https://doi.org/10.1016/j.compedu.2011.01.002.

Li, N., Verma, H., Skevi, A., Zufferey, G., Blom, J. and Dillenbourg, P. (2014). Watching MOOCs together: Investigating co-located MOOC study groups. *Distance Education*, *35*(2), 217-233. https://doi.org/10.1080/01587919.2014.917708.

Liu, J., Lin, C., Hasson, E. P. and Barnett, Z. D. (2011, March). *Introducing computer science to K-12 through a summer computing workshop for teachers.* The 42nd ACM Technical Symposium on Computer Science Education, Dallas. Retrieved from https://dl.acm.org/doi/abs/10.1145/1953163.1953277?casa_token=8CxZVgvEbAwAAAAA:9eGPr2Xc6dF8LpbBoYTSpMxAE-Edfn6_CvTCKzyXty3t7t3WgS7bsC3hhglmPHd_coi2z3lNJZTt. https://doi.org/10.1145/1953163.1953277 in 10.07.2024.

McKenzie, B. (2009). Introductory programming with Alice as a gateway to the computing profession. *Information Systems Education Journal*, *7*(38), 3-8.

Merriam, S. (2013). *Nitel araştırma: Desen ve uygulama için bir rehber*. Ankara: Nobel Yayıncılık.

Miles, M. B. and Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Sage Publications.

Moskal, B., Lurie, D. and Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. *ACM Sigcse Bulletin*, *36*(1), 75-79. https://doi.org/10.1145/971300.971328.

Özdemir, E. Ç. (2023). Okuma becerisinin geliştirilmesinde artırılmış gerçeklik teknolojisinin kullanımına yönelik çalışmaların sistematik derlemesi. *Ahi Evran Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *9*(1), 1-19. https://doi.org/10.31592/aeusbed.1226068.

Papadakis, S., Kalogiannakis, M., Orfanakis, V. and Zaranis, N. (2014, June). *Novice programming environments. Scratch and app inventor: A first comparison*. The 2014 Workshop on Interaction Design in Educational Environments. Retrieved from https://dl.acm.org/doi/abs/10.1145/2643604.2643613 in 02.04.2024.

Patton, M. Q. (2014). *Nitel araştırma ve değerlendirme yöntemleri*. Ankara: Pegem Akademi Yayınları.

Powers, K., Ecott, S. and Hirshfield, L. M. (2007). *Through the looking glass: Teaching CS0 with Alice*. The 38th SIGCSE Technical Symposium on Computer Science Education. Retrieved from https://dl.acm.org/doi/abs/10.1145/1227310.1227386 in 02.04.2024.

Resnick, M., Flanagan, M., Kelleher, C., MacLaurin, M., Ohshima, Y., Perlin, K. and Torres, R. (2009). *Growing up programming: Democratizing the creation of dynamic, interactive media*. The CHI'09 Extended Abstracts on Human Factors in Computing Systems Conference, Boston, Massachusetts. Retrieved from http://web.media.mit.edu/~mres/papers/CHI-programming-panel.pdf in 02.04.2024.

Rojat, M. R. (2022). Pembelajaran pemrograman berorientasi objek (object oriented programming) berbasis project based learning. *Jurnal Portal Data*, *2*(7).

Rushkoff, D. (2010). *Program or be programmed: Ten commands for a digital age*. Or Books.

Saeli, M., Perrenet, J., Jochems, W. M. and Zwaneveld, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, *10*(1), 73-88.

Schultz, L. A. (2011). Student perceptions of instructional tools in programming logic: A comparison of traditional versus Alice teaching environments. *Information Systems Education Journal*, *9*(1), 60.

Solmaz, E. (2014). *Programlama dili öğretiminde Alice yazılımının ders başarısı, eleştirel düşünme ve problem çözme becerileri ile üstbilişsel farkındalık düzeyine etkisi*. Doktora Tezi, Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.

Sykes, E. R. (2007). Determining the effectiveness of the 3D Alice programming environment at the computer science I level. *Journal of Educational Computing Research, 36*(2), 223-244. https://doi.org/10.2190/J175-Q735-1345-270.

Şahin, İ. and Fındık, T. (2008). Türkiye'de mesleki ve teknik eğitim: Mevcut durum, sorunlar ve çözüm önerileri. *TSA, 12* (3), 65-86.

Topalli, D. and Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers and Education, 120*, 64-74. https://doi.org/10.1016/j.compedu.2018.01.011.

Ünal, E. and Çakır, H. (2016). İşbirlikli teknolojilerle desteklenen yapılandırmacı öğrenme ortamının akademik uğraşıya etkisi. *Journal of Instructional Technologies and Teacher Education*, *5*(1), 13-18.

Wang, T.C., Mei, W.H., Lin, S.L., Chiu, S.K. and Lin, J. M.C. (2009). *Teaching programming concepts to high school students with Alice*. 2009 39th İEEE Frontiers in Education Conference. Retrieved from https://ieeexplore.ieee.org/abstract/document/5350486?casa_token=3UWHXQkArjwAAAAA:JOWauWC1An1hsO28jeA2_ZzLP1s1eEQIr0Kfd_a7wolq3NSrlK0NSzOb__dQxTPH3uVOMC0sSZ4 in 02.04.2024

Werner, L., Denner, J., Bliesner, M. and Rex, P. (2009). *Can middle-schoolers use Storytelling Alice to make games? Results of a pilot study*. The 4th International Conference on Foundations of Digital Games. Retrieved from https://dl.acm.org/doi/abs/10.1145/1536513.1536552 in 02.04.2024.

Yildirim, A., and Şimşek, H. (2013). *Sosyal bilimlerde nitel araştırma yöntemleri.* Ankara: Seçkin Yayıncılık.

Yildiz Durak, H. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology, Knowledge and Learning*, *25*(1), 179-195. https://doi.org/10.1007/s10758-018-9391-y.

Yıldız Durak, H. and Güyer, T. (2022). Design and development of an instructional program for teaching programming processes to gifted students using scratch. In *Research Anthology on Computational Thinking, Programming, and Robotics in the Classroom* (127-155). IGI Global.

Zhang, J. X., Liu, L., de Pablos, P. O. and She, J. (2014). The auxiliary role of information technology in teaching: Enhancing programming course using Alice. *The International Journal of Engineering Education*, *30*(3), 560-565.

Genişletilmiş Özet

Giriş

Teknolojinin gelişmesiyle birlikte bilişim sektöründe yeni ürünlerin geliştirilmesi önem kazanmıştır. Özellikle farklı teknolojilerin kullanımı ön plana çıkmıştır. Bu durum programlamanın önemini artırmıştır. Bireylerin problem çözme, analitik düşünme, yaratıcılık gibi birçok üstbilişsel yeteneği programlamayla birlikte gelişmektedir. Bu nedenle eğitimin birçok kademesine programlama dersleri eklenmiştir. Ancak özellikle başlangıç düzeyindeki öğrenciler programlama öğrenmeyi karmaşık bir süreç olarak görmektedir. Öğrenciler programlamanın genel yapısına odaklanamamakta ve soyut kavramları anlayamamaktadırlar. Bu sorunlar programlama öğretiminde yaşanan zorluklar arasındadır. Programlama dillerinde söz dizimi farklı olsa da temel kavramlar benzerdir. Yıllar geçtikçe programlama dillerinin yapısı değişmiş ancak programlama öğretiminde programlamayı kolaylaştıracak değişiklikler çok fazla olmamıştır. Öğrenciler öğretmen açıkladığında anladıklarını ancak öğrendiklerini farklı problemlere aktaramadıklarını belirtmektedirler. Meslek yüksekokulu bilgisayar teknolojileri bölümü müfredatında farklı nitelikte birçok programlama dersi bulunmaktadır. Zira bu bölüm bilişim teknolojileri alanında Türkiye'nin önemli ölçüde nitelikli insan gücünü karşılamaktadır. Ancak meslek yüksekokullarında da akademik başarının düşük olduğu görülmektedir. Akademik başarının düşük olmasının sebeplerine bakıldığında, derslerde sıklıkla geleneksel yöntemlerin kullanıldığı, öğrencinin dikkatini çekebilecek öğretim ortamlarına yer verilmediği veya verilse de yeterli olmadığı gözlemlenmektedir.

Akademik başarıyı artırmak için teknolojinin eğitime entegre edilmesi önemlidir. Programlama eğitiminde programlamanın mantığını anlamak, programlama becerilerini artırmak, kod ve kavramların anlamlarını anlamak için yardımcı araçların kullanılması faydalıdır. Programlama eğitiminde görsel programlama ortamları yardımcı araçlardan biridir. Görsel programlama ortamlarından biri olan Alice programı, programlama mantığının anlaşılmasını kolaylaştırmak amacıyla geliştirilmiştir. Ancak Alice programı kullanılarak daha nitelikli eğitim yapılabilmesi için öğrenme ortamlarında programın kullanımına yönelik araştırmalara ihtiyaç duyulmaktadır. Özellikle öğrenmeye yardımcı programların veya öğrenme ortamlarının hangi durumlarda kullanımının uygun olduğu, ne kadar sürede kullanılması gerektiği ve nasıl öğrenme ortamına entegre edilebileceğinin belirlenmesinde öğrenci görüşleri önem arz etmektedir.

Bu çalışmanın amacı, Alice programının programlama öğrenme ortalarında verimli bir şekilde kullanımının sağlanması amacıyla meslek yüksekokulu bilgisayar teknolojileri bölümünde öğrenim gören birinci sınıf öğrencilerinin görüşlerini incelemektir. Bu amaç doğrultusunda çalışmada "Öğrencilerin Alice programının programlama öğretiminde kullanımına ilişkin görüşleri nelerdir?" sorusuna yanıt aranmıştır.

Yöntem

Tanımlayıcı durum çalışması araştırma kapsamında kullanılmıştır. Durum çalışması desenlerinden biri olan tanımlayıcı durum çalışmalarında, araştırılan durum detaylı bir şekilde incelenerek realist bir şekilde ortaya çıkarılır. Bu çalışmada Alice programının programlama dili öğretiminde etkili, verimli ve amaca uygun kullanılabilmesi için öğrenci görüşlerinin ayrıntılı ve realist bir şekilde incelenmesi amaçlandığından tanımlayıcı durum çalışması tercih edilmiştir.

Uygulama süreci 2015-2016 eğitim-öğretim yılının bahar döneminde gerçekleştirilmiştir. Nesne Tabanlı Programlama I dersi kapsamında yürütülmüştür. Haftada 4 saat olmak üzere 8 hafta sürmüştür. 34 birinci sınıf öğrencisi ile çalışılmıştır. Bilgisayar teknolojileri bölümü müfredatında zorunlu ders olarak Nesne Tabanlı Programlama I dersi yer almaktadır. Kurs müfredatı, veri türleri ve değişkenler, operatörler, kontrol yapıları, döngüler,

diziler, metodlar, sınıf yapıları ve kalıtım dahil olmak üzere Java programlama dilinin temel programlama kavramlarını kapsamaktadır. Ders süresince öğrenciler hem Alice programı hem de Java programlama dili ile örnek ve uygulamalar yaptmışlardır. Öğretim Gagne'nin dokuz aşamalı öğretim modeli kullanılarak gerçekleştirilmiştir. Uygulama sürecinin ardından 34 öğrenci arasından seçilen 18 öğrenciyle iki odak grup görüşmesi yapılmıştır.

Uygulama sürecine katılan 34 öğrenci kolay örnekleme yöntemi kullanılarak seçilmiştir. Uygulama sürecine katılan 34 öğrenciden 18 öğrenci çalışma grubunu oluşturmuştur. 18 öğrencinin seçiminde amaçlı örnekleme yöntemi içerisinde yer alan maksimum çeşitlilik örneklemesi kullanılmıştır. Araştırmanın yapıldığı meslek yüksekokulunun akademik takvimine göre odak grup görüşmelerinden önce Nesne Tabanlı Programlama I dersinin ara sınavı yapılmıştır. Ara sınav puanları dikkate alınarak üç performans düzeyi belirlenmiştir (üst düzey: 100-67, orta düzey: 66-34, alt düzey: 33-0). Odak grup görüşmeleri için ara sınavda her performans düzeyinden üçer öğrenci seçilmiştir. Her odak grupta 9 öğrenci ile görüşmeler yapılmış olup, görüşmelere toplam 18 öğrenci katılmıştır.

Araştırmacılar, Alice programı ve Alice programı ile tasarlanan öğrenme ortamı hakkında öğrenci görüşlerini belirlemek için yarı yapılandırılmış görüşme formu geliştirmiştir. Beş uzmanın görüşü alınarak gerekli düzenlemeler yapılmıştır. Görüşme formunun son hali 9 maddeden oluşmaktadır.

Öğrencilerle yapılan görüşmelerin sonuçlarının analiz edilmesinde içerik analizi kullanılmıştır. Ses kayıt cihazı aracılığıyla görüşmeler kayıt altına alınmıştır. Kaydedilen görüşme verileri kodlama amacıyla yazıya aktarılmıştır. Kodlama güvenirliği sağlamak için kodlama, araştırmacı dışında başka bir kodlayıcı tarafından tekrar yapılmıştır. Kodlayıcılar arasındaki fikir birliğini hesaplamak için (eşleşen madde/(eşleşen madde + eşleşmeyen madde)* 100) formülü kullanılmıştır. Kodlayıcılar arası güvenirlik %91.66 olarak bulunmuştur.

Bulgular

Analiz sonucunda dört tema ortaya çıkmıştır. Temalar; Alice programının programlama dili öğreniminde yararları, Alice programının olumlu özellikleri, Alice programının olumsuz özellikleri, Alice programının gelecekte kullanımıdır. Alice programının programlama dili öğreniminde yararları temasında; programlamaya olan etki, görsellik (program çıktılarının görsel olarak görülebilmesi), sürükle-bırak şeklinde üç kategori ortaya çıkmıştır. Programlamaya olan etkisi kategorisinde temel kod kavramlarının öğrenilmesini kolaylaştırması kodu en çok görülen kod olmuştur. Kalıcılığı arttırması, görsellik (program çıktılarının görsel olarak gösterülebilmesi) kategorisinin en yaygın kodu olmuştur. Sözdizimi hatalarını önlemesi kodu, sürükle-bırak kategorisinde en yaygın olan koddur. Alice programının olumlu özellikleri temasında altı kod ortaya çıkmıştır. Kullanım kolaylığı ve sürükle-bırak mantığıyla çalışılması en yaygın kodlardır. Alice programının olumlu özellikleri temasında iki kategori ve beş kod ortaya çıkmıştır. Sınırlı dil desteği, Alice programın özellikleri kategorisinin en çok görülen kodudur. Alice programında kod yazılamaması ve bu nedenle kod yazma pratikliğinin kazanılamaması, Alice programının programlama üzerindeki etkisi kategorisinde en yaygın kod olmuştur. Öğrencilerin çoğu gelecekte programlama dili öğretmeleri gerektiğinde Alice programını tercih edeceklerini ayrıca Alice programını kullanmaya devam edeceklerini ifade etmiştir.

Sonuç, Tartışma ve Öneriler

Bu çalışma programlama dili öğretim ortamlarından Alice programının etkili bir şekilde kullanılmasını sağlamak için meslek yüksekokulu bilgisayar teknolojileri bölümünde öğrenim gören birinci sınıf öğrencilerinin görüşlerinin belirlenmesi amacıyla yapılmıştır. 18 öğrenciyle yapılan odak grup görüşmelerinin sonuçları incelendiğinde, öğrencilerin Alice programının

programlama dili öğreniminde yararları, Alice programının olumlu ve olumsuz özellikleri ve programın gelecekte kullanımına ilişkin görüşler belirttikleri görülmüştür.

Araştırma sonuçlarına göre öğrenciler çoğunlukla Alice programının programlama dili öğreniminde yararlarına ilişkin temel kod kavramlarının öğrenilmesini kolaylaştırdığını, kalıcılığı arttırdığını ve söz dizimi hatalarını önlediğini belirtmişlerdir. Alice programında kodlar sürükle bırak mantığı kullanılarak eklenir. Animasyonda görsel olarak hangi kodun hangi sonucu verdiği görülebilir. Böylece kodların çalışma mantığı daha iyi anlaşılabilir, kullanıcıların programlama dersine olan ilgisi artabilir ve özellikle öğrenmede kalıcılık artabilir. Öğrencilerin genellikle karmaşık bir süreç olarak gördükleri programalama eğitimine karşı olan ilgilerinin artması, programlama yoluyla özgün teknolojiler üretmeye yönelmelerini beraberinde getirmektedir.

Öğrenciler programının kullanımının kolay olmasını ve sürükle-bırak özelliği ile çalışılmasını Alice programının olumlu özellikleri olarak belirtmişlerdir. Kullanıcılar Alice programıyla çalışırken kodları ekrana sürükleyip bırakmaktadır. Böylece sözdizimi hatalarının tespit edilmesiyle vakit kaybetmezler, ilgilendikleri tek husus kodların işlevleridir. Kodların işlevinin ayrıntılı bir şekilde anlaşılmasıyla özgün uygulamalar ve programlar geliştirilebilir. Böylece bilgi teknolojileri alanında yenilikçi ürünler ortaya çıkarmak mümkün olabilir.

Öğrenciler Alice programında kod yazılamamasını ve bu nedenle kod yazmada pratiklik kazanılamamasını sevmediklerini, ayrıca ileri düzey programlama yapmak için yetersiz gördüklerini olumsuz özellik olarak belirtmişlerdir. Alice programı programlama dilini öğrenmeyi kolaylaştırsa da ileri düzey programlama için yeterli olmayabilir. Programlamaya yeni başlayan kullanıcılar için daha uygun olabilir. Öğrencilerin büyük çoğunluğu gelecekte Alice programını kullanacaklarını belirtmişlerdir. Bu ifade Alice programının öğrencilere programlama dili öğrenmede fayda sağladığı şeklinde yorumlanabilir.

Çalışmanın tamamı dikkate alındığında Alice programının programlamaya yeni başlayan kullanıcılar için faydalı olacağı düşünülmektedir. Eğitim ortamlarında programlamayı sevmek ve programlama mantığını anlamak için görsel programlardan yararlanılabilir.