Learning Content of Educational Tasks in Computer Programming Training for 10-11 Year Old Children

Kosta Andreev Garov [1], Veneta Veselinova Tabakova-Komsalova [2]

[1] *University of Plovdiv Paisii Hilendarski, 24 Tsar Asen, 4000 Plovdiv , Bulgaria*

Abstract **– The lack of any programming training experience for small students provoked a number of pedagogical experiments in this area. The general conclusion is that training for 10-11 year olds should be based on solving learning tasks. Their proper selection, structuring and matching is an extremely important activity that contributes to a more effective and qualitative implementation of the objectives of computer programming training. This article is devoted to the tasks of computer programming training. It discusses different classifications of tasks adapted to the specifics of computer programming curricula for 10-11 year olds. Particular attention is paid to the classification of the tasks, according to the curriculum content of the study of a specific learning topic "Constructing consecutive actions". Specific tasks are described to assist teachers in teaching the curriculum.**

Keywords **– tasks, computer programming, computer modelling, school, teaching methodology, skills**

DOI: 10.18421/TEM64-26
https://dx.doi.org/10.18421/TEM64-26

Corresponding author: Kosta Andreev Garov,
University of Plovdiv Paisii Hilendarski,
Plovdiv, Bulgaria
Email: kosgar@uni-plovdiv.bg

Received: 07 August 2017
Revised: 27 August 2017
Accepted: 20 September
Published: 27 November 2017

1. Introduction

The widespread penetration of information and communication technologies into practice raises before the educational systems of a number of countries the problem of computer literacy of students. It is believed that part of the computer literacy is the knowledge and skills in the field of computer programming [12]. Historically, programming training began in universities in the 1970s. In 1986, Bulgaria became the fourth country in the world to introduce Informatics as a compulsory subject, and so computer programming began to be studied in high school. At the beginning of the 21st century, in most European countries computer programming was integrated into lower secondary education [3], and in three countries - Estonia, Greece and England it was introduced at an early stage, i.e. for 10-11 year olds [3], [4]. The new Law on Pre-school and School Education in Bulgaria [6] determines the study of computer programming elements in the elementary school level in the subject **"Computer Modelling"** in the obligatory education of students. The lack of any programming training experience for small students provoked a number of pedagogical experiments in this area. The general conclusion is that for 10-11 year olds lecture-based theoretical training is unacceptable, and that training must be based on solving learning tasks. Their proper selection, structuring and combining in acquiring, consolidating and developing certain knowledge and skills, as well as the skilful application of the knowledge and skills formed in solving creative tasks that predispose the implementation of integrative links with other disciplines, contribute to a more effective and quality implementation of the objectives of the computer programming training. The Computer programming training included in the course "Computer Modelling" for 10-11 year olds has an introductory nature and is aimed at mastering basic knowledge, skills and relations in programming [11]. It is based on modern digital devices and software suitable for the age of the students, which create conditions for positive emotional attitude and overall development of the child's personality. The priority of learning is not to teach students to encode algorithms in a particular programming language, but to learn about problem-solving habits [13] in a programming environment as we learn to solve mathematical problems. The training will help develop critical thinking - and in particular the development of **"computational thinking"** needed to understand a problem, as well as the ability to separate a problem into smaller parts that are more easily solved, etc. The basic concepts required in mathematics and other curriculum subjects will be studied - what is the algorithm, how data is presented, and different approaches to analysis.

2. Tasks in the school training course in computer modelling

The psychological-pedagogical literature presents a diverse classification of the tasks used in the school disciplines. Different classifications of IT and IT technology tasks adapted to the specifics of computer science curricula at school are made in [1] and [2]:

- Classification by content;
- Classification according to the didactic goals achieved by their solution;
- Classification according to the activities performed by the students to solve the tasks;
- Classification by complexity;
- Classification according to the type of cognitive activity of the students needed to solve them;
- Classification of computer modelling tasks according to the type of information activities carried out: operational tasks; conceptual tasks; basic practical tasks; accompanying practical tasks; assignments for course projects.

Computer modelling tasks according to the learning content. The following main topics are included in the study of computer programming in elementary school [9]:

• Constructing consecutive actions;
• Constructing repetitive actions;
• Visual programming environment;
• Working with text and sound in a visual environment;
• In the world of animation. (http://www.mon.bg/?go=page&pageId=1&subpageId=1689).

In this work, we will discuss in more detail the tasks of the topic "Constructing consecutive actions" by presenting the terms of specific tasks. The design of consecutive actions is a major activity in the creation of computer programs [14]. That is why initial programming training should start with the study of the topic "Constructing consecutive actions". The tasks below can be solved with both a digital device and without a digital device. Digital blocking environments such as Scratch [7] and [10], Alice, KODU, etc. can be used to solve tasks with a digital device [8]. The conditions of a large number of tasks can be found on the platform http://code.org [5]. The public organization established in 2013, **code.org**, has launched a major online learning project and many free educational programming materials can be found on its site. Solving the following tasks with a digital device can be done directly at http://code.org.

Learning tasks for the topic "Constructing consecutive actions" can be classified into the following groups.

1. Arrangement tasks with consecutive action of image-forming blocks

1.1. Arranging parts of an image with a paper puzzle (task solved without a digital device);

Task 1: In a square mesh of a certain size (for example: 10x10 squares), place the following shapes in the order and place defined by the teacher (Fig. 1.) by performing the following sequence of actions:

1. On the first row, count two squares and place the blue square.
2. On the eighth row, count four squares and place the tip of the red triangle.
3. On the seventh row, count six squares and place the yellow circle.
4. On the first row, count seven squares and place the orange square.
5. On the second row, count seven squares and place a green triangle.
6. On the third row, count seven squares and place a second green triangle.
7. On the fourth row, count seven squares and place the third green triangle.

The result is getting a meaningful picture (similar to a paper puzzle).

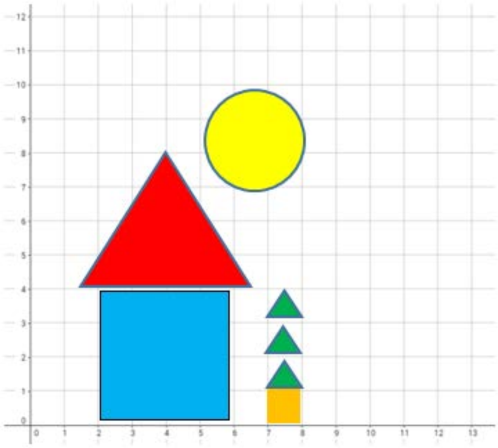
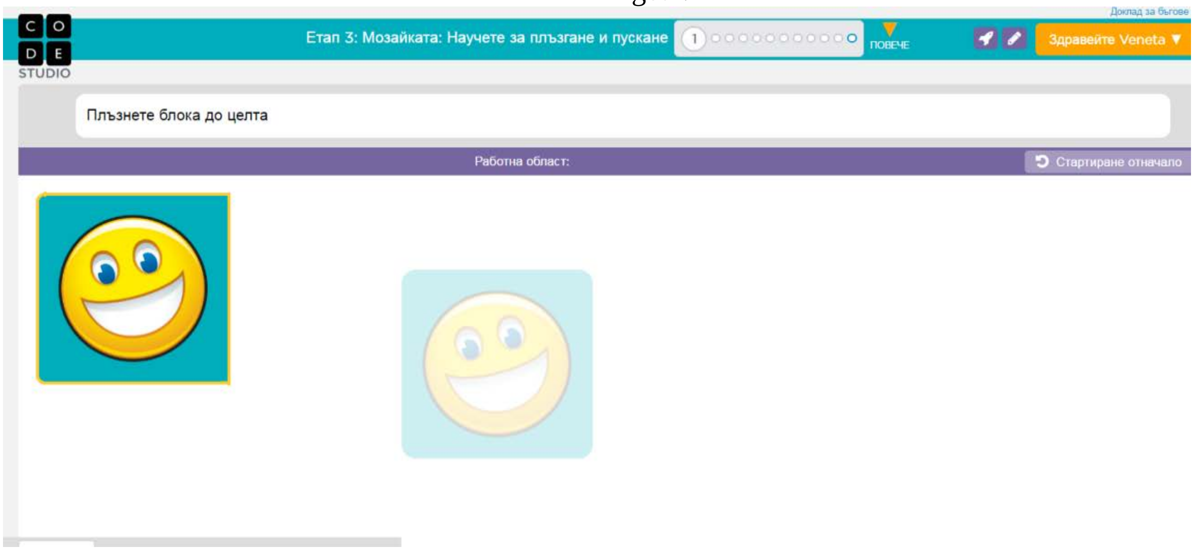


Figure 1. Arranging a puzzle on instructions

1.2. Dragging and dropping blocks in a customized visual environment (tasks with a digital device at http://code.org);
Dragging and dropping are actions that can be performed with the mouse on the computer in the following ways - left-click on the item and hold it by moving it to the designated location and dropping the button.

Task 2: Arranging figures in a digital environment (solving on http://code.org) (Fig. 2.). With dragging and dropping actions, move the block to the specified goal.

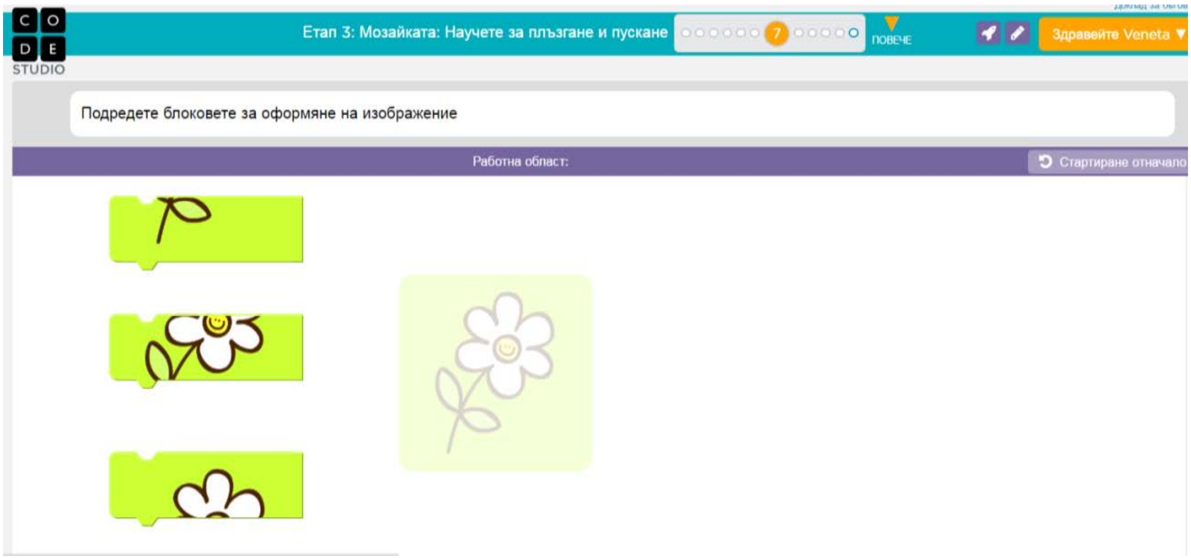


Плъзнете блока до целта

Figure 2. Arranging figures

1.3. Arranging parts of a whole image in a visual environment (tasks with a digital device at http://code.org).

With the dragging and dropping actions performed consecutively, arrange the blocks in Fig.3., so as to obtain the specified image.



Подредете блоковете за оформяне на изображение

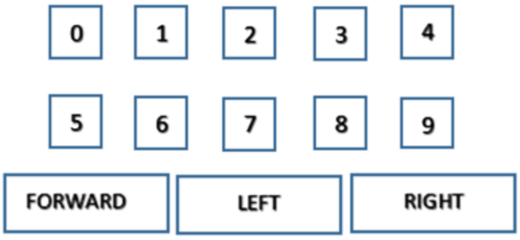
Figure 3. Arranging a puzzle with consecutive actions in a digital environment

2. *Tasks for arranging blocks for moving an object in the four directions*

In this group of tasks, an object moves through consecutive actions of mosaic blocks, and the tasks can be solved without and with a digital device at http://code.org;

Task 3 (without digital device):
a) Students are divided into two groups (teams). Each group prepares on a sheet a sequence of

movements of the pencil to draw a specific letter of the Bulgarian alphabet intended for implementation by the students from the other group through the following blocks:



Each team analyses the different details of the letter, and then uses this information to create an algorithm (sequence of instructions) on how the other team can draw the specified letter. The teams exchange the algorithms and draw the letter based on the instructions given to them. For example, when pupils have to describe the writing of the letter "P" they arrange the cards in the following order: Forward 5, Right, Forward 2, Right, Forward 5.

b) Select the correct block from the proposed three blocks to draw the man's foot in Fig. 4 with one line.

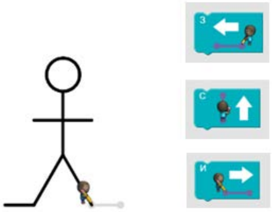
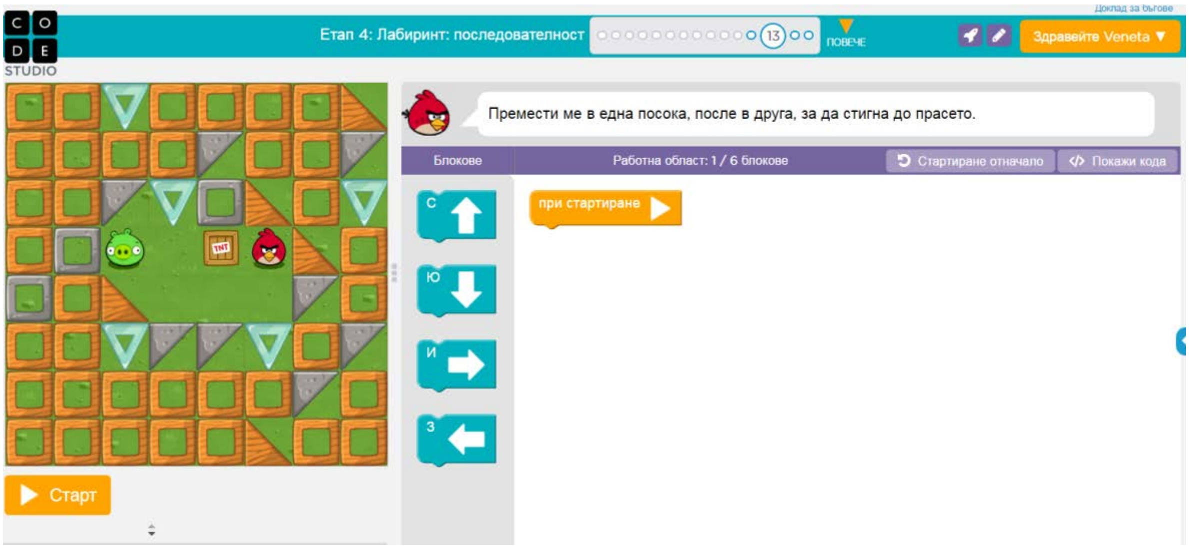


Figure 4. Figure drawing

Task 4 (with a digital device): Select a sequence of actions and arrange the blocks in the work area of the proposed task from http: // code.org on Figure 5 to help the chicken reach the pig.



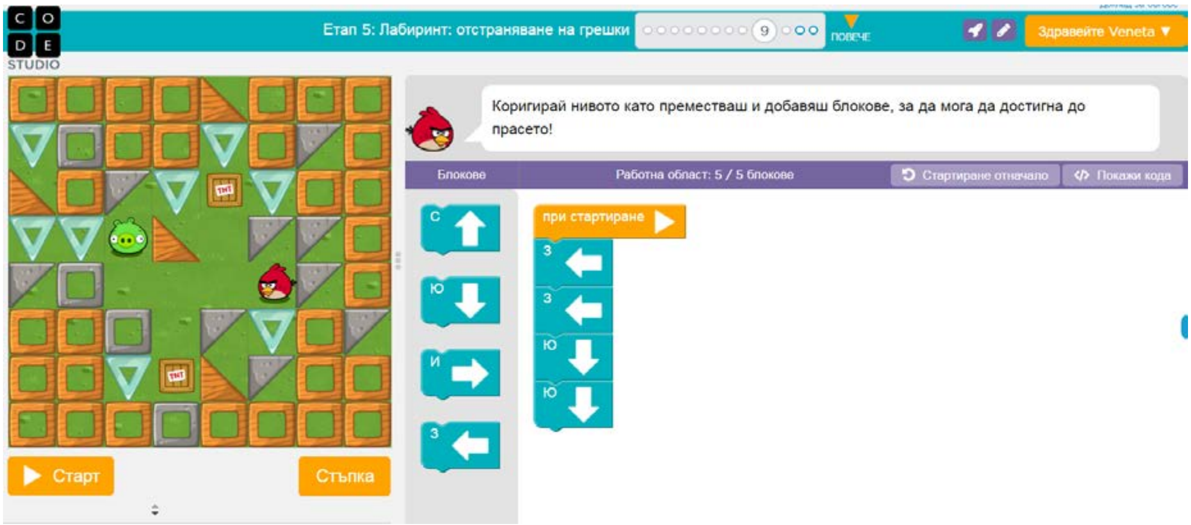
Премести ме в една посока, после в друга, за да стигна до прасето.

Figure 5. Arranging blocks to move in the four directions

With tasks like Task 4 students become familiar with and use the various types of motion blocks. Mistakes are made in solving different tasks, and students must learn to detect and correct errors in solutions.

Task 5. (Task without a digital device). Students are divided into two groups (teams). Each group receives one piece, one sheet of squares, a marker, and a set of cards as described in Task 3. They also have one drawing algorithm in which an error has occurred. Students must find the error and describe the correct algorithm with their set of cards.

Task 6. (Task with digital device in http://code.org)
In the arrangement of the blocks in the working area of Fig. 6. an error has occurred. Make the necessary adjustments, so the red chicken gets to the pig. Select the "Start" button to check if you have worked correctly.

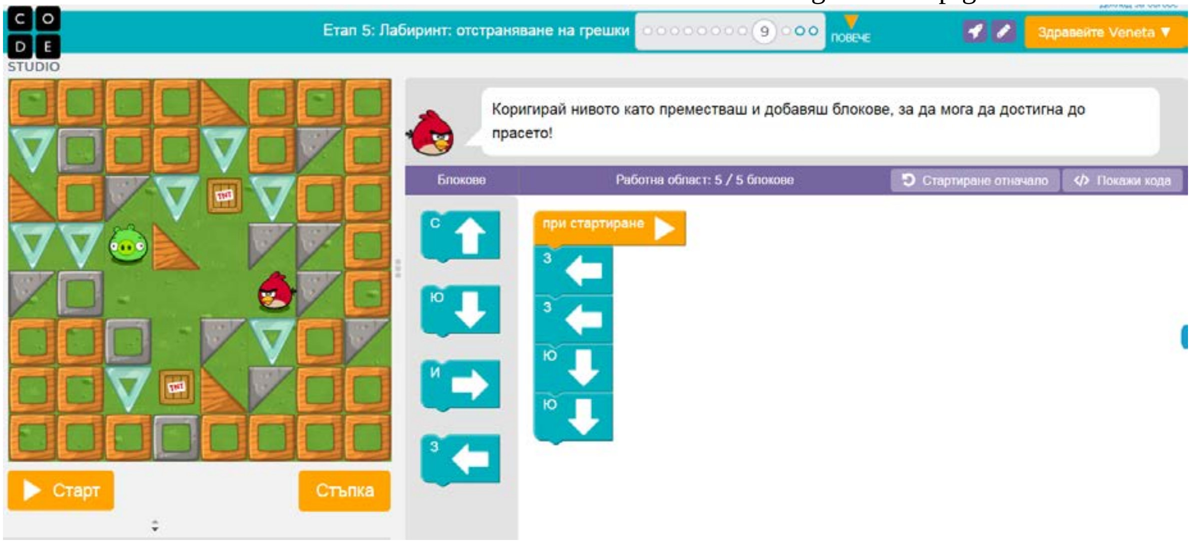


Коригирай нивото като преместваш и добавяш блокове, за да мога да достигна до прасето!

Figure.6 Detecting errors in a ready algorithm.

Task 7 (without a digital device):
Each of the teams formed in solving Task 3 to verify the correctness of the drawing they have drawn.

Task 8 (with a digital device): In the arrangement of the blocks in Fig. 7. an error has been made. Make the necessary adjustments so the red chicken gets to the pig.



Коригирай нивото като преместваш и добавяш блокове, за да мога да достигна до прасето!

Figure 7. Detecting errors and adjusting the order of the instructions

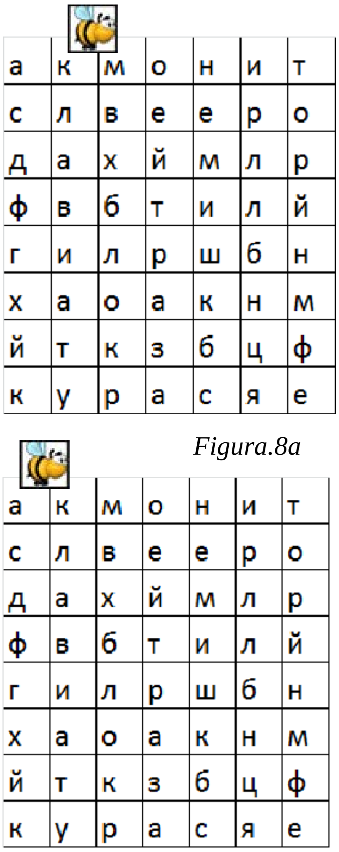
3. Tasks for arranging blocks for moving a word creation object

This group of tasks monitors a sequence of actions to comply with spelling rules when working with text.

Task 9 (without a digital device**):**
a) Arranging a word from the letters (created in task 3) in a table when moving in specified directions.

b) Formation of a word by moving on given instructions in a square mesh with letters by using blocks from task 3:

1. Moving east and then south to write the word **monitor** (Fig.8a)
2. Moving south and then east to write the word **keyboard** (Fig.8b.)



а	к	м	о	н	и	т
с	л	в	е	е	р	о
д	а	х	й	м	л	р
ф	в	б	т	и	л	й
г	и	л	р	ш	б	н
х	а	о	а	к	н	м
й	т	к	з	б	ц	ф
к	у	р	а	с	я	е

Figura.8a

а	к	м	о	н	и	т
с	л	в	е	е	р	о
д	а	х	й	м	л	р
ф	в	б	т	и	л	й
г	и	л	р	ш	б	н
х	а	о	а	к	н	м
й	т	к	з	б	ц	ф
к	у	р	а	с	я	е

Figura.8b

Figure 8. Movement in a table to form a word

4. Tasks to indicate the correct ordering of blocks of multiple responses

When solving the tasks in this group, it indicates a sequence of actions for correctly arranging blocks of multiple responses.

Task 10 (without a digital device): Help the chicken gets to the pig in each labyrinth given in Figure 9., by properly connecting labyrinths with the sequence of instructions to achieve the goal.

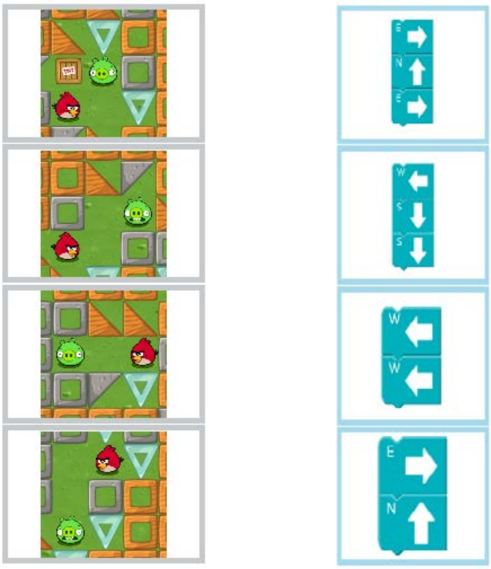
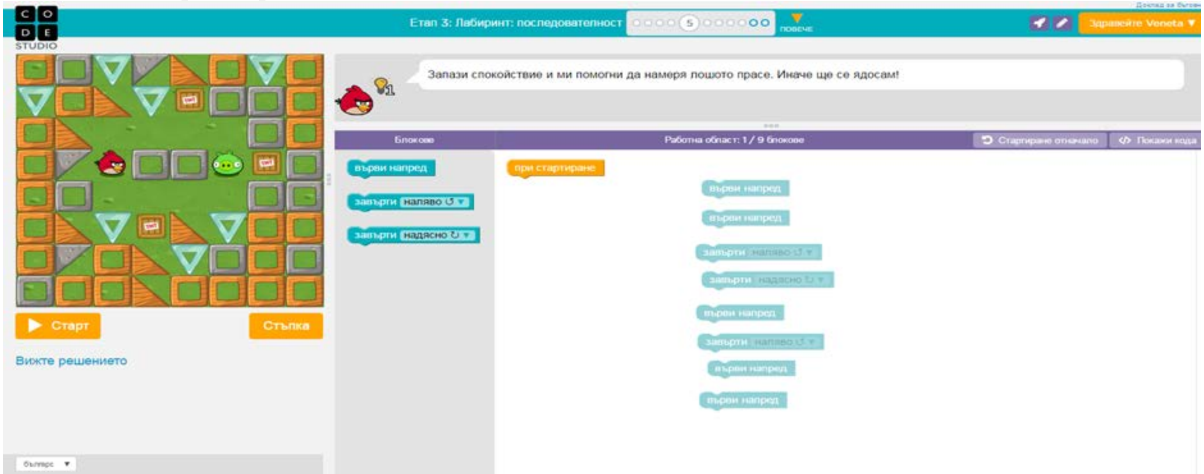


Figure. 9. Linking a situation with the sequence of instructions to solve it

Task 11 (task with a digital device): Arrange the blocks in the work area so that the red chicken gets to the pig (Fig.10.). Find and place the missing block in the right place.



Етап 3: Лабиринт: последователност

Запази спокойствие и ми помогни да намеря пощото прасе. Иначе ще се ядосам!

Блокове

Работна област: 1 / 9 блокове

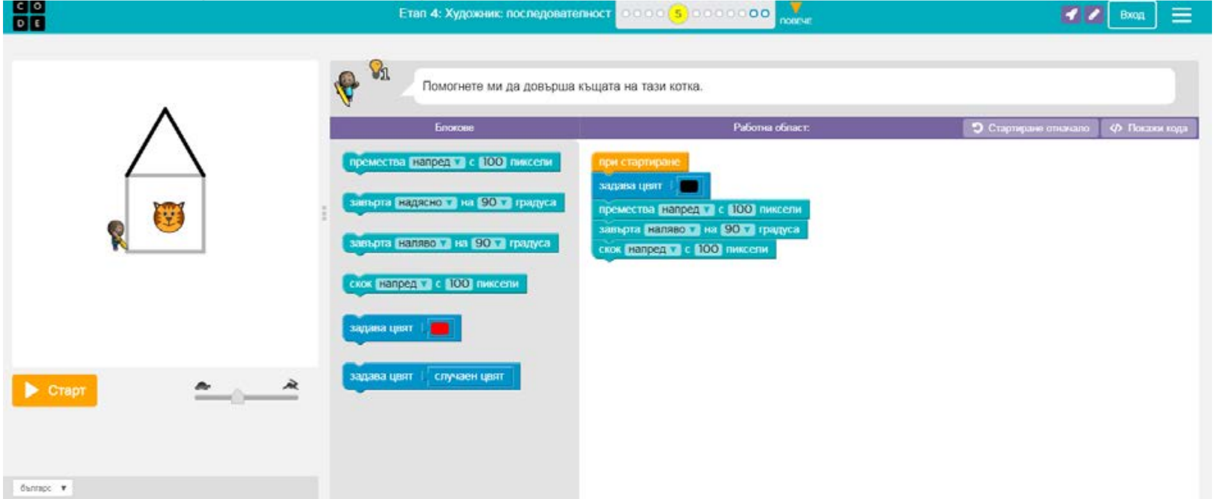
при стартиране
върви напред
върви напред
завърти наляво ↺
завърти надясно ↻
върви напред
завърти наляво ↺
върви напред
върви напред

Figure 10. Selecting a block for solving an assignment

5. Tasks for consecutive actions in combined use of motion blocks

Here, the students are introduced to new blocks of motion, including movement in a certain direction (forward and backward) at a certain distance (pixels) and turning in a certain direction (left, right) at a certain angle.

Task 12: Help the artist (Fig.11.) to finish the cat's cottage by using the blocks and arrange them correctly and with specific values in the work area. Check with the "Start" button if the sequence of blocks you selected is correct and you have correctly set the direction, distance and angle.



Помогнете ми да довърша къщата на тази котка.

премества напред с 100 пиксели
завърта надясно на 90 градуса
завърта наляво на 90 градуса
скок напред с 100 пиксели
задава цвят
задава цвят случаен цвят

```
при стартиране
задава цвят
премества напред с 100 пиксели
завърта наляво на 90 градуса
скок напред с 100 пиксели
```

Figure 11. Sequence of drawing actions

3. Conclusion

In the last three years in some schools of Plovdiv District, we organized and conducted a pedagogical experiment for conducting computer modelling training with 10-11 year old students from seven different schools. The experience we have gained shows that the following main results are achieved with the help of the tasks of "Constructing consecutive actions"**:**

- Students easily grasp the concepts of drag, drop, block and algorithm.
- They are able to apply formally-defined algorithms in a visual environment by arranging blocks in a specified linear sequence by sliding and dropping, and using the built-in blocks to enter and adjust the program.
- Students acquire skills to test and correct a certain sequence of instructions.
- Using a visual digital environment, they can create storyline stories according to a set story.

The tasks proposed in this article are related to the content of the learning content "Constructing consecutive actions". In the future, work can be developed towards the development of systems of learning tasks and other topics of training content.

Acknowledgements

The authors express their appreciation for the scientific project FP17-FMI-008 "Innovative software tools and technologies with applications in research in mathematics, informatics and pedagogy of training" at the Scientific Research Fund of Plovdiv University Paisii Hilendarski for the partial financing of the present work.

References

[1] Garov, K. (2010). For the Informatics and Information Technology Training Tasks, Collection of Reports of the National Conference "Education in the Information Society", Plovdiv, p. 95-101.

[2] Dureva, D. (2003), Problems of the Methodology of Informatics and Information Technology Education, University Publishing House, Southwest University "Neofit Rilski", Blagoevgrad.

[3] Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding-priorities, school curricula and initiatives across Europe*. European Schoolnet.

[4] Kafai, Y. B., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, *95*(1), 61-65.

[5] An Hour of Code for every student: http://code.org, (accessed 28.July.2017).

[6] Law on Pre-school and School Education , Retrived from: http://dv.parliament.bg/DVWeb/showMaterialDV.jsp?idMat=97877, (accessed 28.July 2017).

[7] Glushkova T. (2016), Application of Block Programming and Game-Based Learning to Enhance Interest in Computer Science, Journal of Innovations and Sustainability, Volume 2, Number 1, ISSN 2367-8151.

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

[9] Tabakova-Komsalova, V., Glushkova, T. (2016). Some ideas for training of computer modeling in elementary school, *Educational Alternatives*, *14*, 383-390.

[10] Ford, Jerry Lee. (2009). Scratch programming for teens.Boston, MA : Course Technology, ISBN 9781598636994 1598636995

[11] Tumlin, N. (2017, March). Teacher Configurable Coding Challenges for Block Languages. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 783-784). ACM.

[12] Yasmin B. Kafai, (1996). Constructionism in Practice: Designing, Thinking, and Learning in a Digital World, Routledge, p. 339, ISBN 0-8058-1985-1.

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

[14] Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to Learn: Where Does It Belong in the K-12 Curriculum?. *Journal of Information Technology Education: Research*, *15*, 283-303.