Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 7, S. 105-109, 5, 2023

Araştırma Makalesi

IJANSER
https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 7, pp. 105-109, 5, 2023

Research Article

Virtual programming environments and simulations in favor of active learning of programming

Krisztina Czakóová

Department of Informatics/J. Selye University, Slovakia

(czakoovak@ujs.sk) Email of the corresponding author

(Received: 06 June 2023, Accepted: 20 June 2023)

(1st International Conference on Pioneer and Innovative Studies ICPIS 2023, June 5-7, 2023)

ATIF/REFERENCE: Czakóová, K. (2023). Virtual programming environments and simulations in favor of active learning of programming. *International Journal of Advanced Natural Sciences and Engineering Researches, 7(5), 105-109.*

Abstract – Microworlds are environments that allow the user to directly experience in a virtual simulation environment. It can also be understood as a carefully designed, constantly expanding software environment that focuses on learning certain concepts, phenomena or relationships within a clearly defined area. It is an environment where the students have a specific space for discovery and investigation, where to a certain extent they can define the goals and learning activities. Based on this concept, learning is supported, assisted and regulated, but not strictly directed or dictated in any way. One of the great advantages of microworlds (virtual program environments) is that they have a simple and low number of syntactic units. Microworlds are built on metaphors. They ensure the development of a colorful set of problems, which are directly related to the user's (student's) life experience. The instructions executed by an object in the given simulation environment are always visible. The order of execution allows students to understand the semantics of the input control constructs. In this paper, we would like to present child-oriented programming languages (microworlds also known as mini languages), which can be powerful tools for introducing students to the world of programming languages, as well as providing a virtual space for the development of algorithmic thinking. It is advantageous if the program environment has a structured editor, which helps the student to eliminate syntactic errors. These program environments are classified and evaluated by taking these aspects into account and analyzing additional characteristic properties.

Keywords – Programming, Virtual Environment, Active Learning, Computer Science, Education

I. INTRODUCTION

Getting to know the virtual environment of mini languages already at an early school age contributes to the development of thinking and imagination and to the acquisition of basic computer management skills. The attractiveness of the toolbox options provided by mini languages can attract the attention of even the most demanding users. These virtual simulation environments can be used well in elementary school programming education. By posing, solving and practicing interesting tasks arising from everyday life, students' motivation is strengthened, their creativity and problem-solving skills develop.

There are several reasons to teach mini languages. Above all, they open a new path and new opportunities for educational activities. Mini languages provide a strong basis for systematic and algorithmic processing of problems. From a didactic point of view, programming provides space for developing the foundations of logical and abstract thinking. ([15] - [17], [20])

II. PROPERTIES OF MINI LANGUAGES

Typically for mini-languages, the focus is primarily on the attractiveness of the central object of the microworld. It is very important to get as close as possible to the real world, which greatly supports cooperation during problem solving. The basic characteristics of mini languages are as follows:

- **Simplicity of semantics and syntax**;
- **Visualization** – in the case of operations with objects, the reactions are immediately visible in the micro world, presented on the computer screen;
- **Attractiveness and meaningfulness** – the language is aimed at the given target group of students (adjusted to their level of knowledge);
- **Dialogue mode -** all commands are executed in control or programming mode;
- **Modularity** – it has a mechanism for creating abstract instructions (procedures), which can be well used in solving sub-problems.

The most effective way to learn a programming language is through problem solving. It is advantageous if the program environment helps the student not only to eliminate syntactic errors, but also to find out about the names of structural constructions. ([13], [18])

III. MATERIALS AND METHODS

The question arises, how we can ensure the development of thinking in the case of children, most of whom do not show interest in the more complex environment of higher-level programming languages such as C languages, Basic, Java and so on. The solution to this problem can be child-oriented mini languages, the range of which is truly colourful.

Other characteristic features on the basis of which mini languages can be classified are: the principle/technique used to represent the source code (the way of giving commands to control the object of the microworld), i.e.:

- *Puzzle-like or block-based programming (Scratch)* – each command is represented by a building block, which also contains text or an expression. The shape of the building element is given, so it is not possible to place it incorrectly in the program, as a result of which the program will be formatted correctly by itself. By combining the individual commands, we can create one or more lines, which together form a readable program.
- *Tile-based programming (Baltie 3, Kodu Game Lab)* - by eliminating the text into images and icons, the programming process is simplified, but the program itself remains in the background. A sequence of instructions can be created, some of which may also have parameters. The program itself is drawn in the form of a large picture, which is made up of many small tiles (icons). This type of environment is suitable for those who cannot yet read well. [3]
- *Textual programming (Imagine Logo, Karel 3D)* - the language used to enter commands is similar to everyday speech. The result of the specified instruction is immediately displayed on the screen. These environments provide space for creating models and simulations. The program's environment offers a wide range of usage options, while being easy (intuitive) to handle and attractive, it is also suitable for the implementation of more complex and demanding programming tasks.

Programming environments designed and developed for beginner programmers could be classified and evaluated based on several additional properties. We can mention the programming style, the way how the program can be controlled (procedure-based, object-oriented, event-based), or the used program constructions, program structures and concepts (loops, variables, parameters, conditions, definability of own procedures). It is also interesting to mention the mechanisms for regulate the management of the elimination of the possibility of errors during the program structure (shape matching, provides a range of valid options). It has summarized on the Table 1.

Table 1. Characteristics and highlights of the selected mini languages (virtual environments and simulations)

Program environ-ment	Programming style	Code represen-tation	Preventing (syntax) errors
Scratch	object oriented, event based	block-based	shape matching, select of valid opt., syntax directed edit
Kodu Game Lab	object oriented, event based	tile-based	select of valid opt., syntax directed edit
Imagine Logo	procedural, object oriented, event based	textual	-
Baltie 3	procedural	tile-based	shape matching, select of valid opt.
Karel 3D	procedural	textual	-

It can be seen from the previous overview that the number of options is large. We are convinced that

each teacher chooses the "best and most suitable" from the given virtual program environments for joint creative work with students based on their own experience ([10] - [12], [19]). A modern teacher is always looking for new procedures that he could apply in teaching in order to activate his students to solve problems creatively. ([4] - [9], [14])

Table 2. Content of the curriculum focused to programming (ISCED 1, ISCED 2). [1]

	Problem analysis	**Interactive implementation of a solution** (Using programming language.)
3 - 4. year of primary school	Recognition of true / false statements. Decision making and selection of appropriate commands. Outlining possible solutions.	Object control with a series of commands (sequences). Select the appropriate command.
5 - 6. year of primary school	+ And / or / not. + Command line and repetition (for loop) stepping. + Deciding the truth of a statement.	+ Faulty command detection. + Knowledge of the structure of language (commands). + Programming and the program executing (run).
7 - 9. year of primary school	+ Recognition of connections (repeating parts, recognizing the problem of decision making).	+ Algorithm - programming language context. + Loops and conditions.

In accordance with the content standard listed in Table 2, we suggest using virtual programming environments and simulations according to their characteristics (Table 1) to teach programming as follows (see Table 3).

Table 3. Recommended virtual programming environments according to options of their programming constructs. [2]

	Recommended programming environment for using	**Programming constructs**
3 - 4. year of primary school	**Baltie 3** **Kodu Game Lab** (tile-based programming)	Series of commands (sequences). FOR loop.
5 - 6. year of primary school	**Scratch** (block-based programming)	Series of commands (sequences). FOR loop. While loop. Count loop. Conditional.
7 - 9. year of primary school	**Scratch** (block-based programming)	Series of commands (sequences). FOR loop.
	Imagine Logo* (textual programming)	While loop. Count loop. Conditional. Variable. Procedures/Methods*. Parameters*.

We recommend using the project teaching method or applying problem-based teaching.

IV. RESULTS

Our goal is to change the students' attitude towards creative activity based on their own beliefs and achievements. We consider it of primary importance that the student is sufficiently motivated during programming and performs meaningful activities. So that he acquires the necessary skills and knowledge in a spontaneous and playful way. Learning is facilitated by self-discovery. The objects are controlled by a language that the student understands. The behaviour of the object can be observed directly on the screen and the execution of the entered commands can be monitored. The student has the opportunity to draw new ideas in the conceptual space of the environment, the possibilities of which are unlimited.

V. CONCLUSION

An important prerequisite for creative learning is the provision of an environment that supports creativity. Such an environment can be influenced both by the choice of the right activities and by the appropriate motivation for creative activity. In order for the creator to go beyond his possibilities and overcome limitations, he needs a space for thinking outside the usual structures, which creates unexpected situations and generates unpredictable events. Original ideas and spontaneous reactions of creators must be supported.

REFERENCES

[1] K. Czakóová, M. Csóka. Ozobot coding challenges in elementary school informatics. *In. INTED2022 Proceedings.* DOI: 10.21125/inted.2022.1491, pp. 5834-5840, Valencia: IATED Academy, 2022. ISBN 978-84-09-37758-9. ISSN 2340-1079.

[2] K. Czakóová. Microworld environment of small language as „living laboratory" for developing educational games and applications. *In. Proceedings of the 13th International Scientific Conference „eLearning and Software for Education" : Could technology support learning efficiency?* Volume 1, DOI: 10.12753/2066-026X-17-042, 2017/1, pp. 286-291. Bucharest: "CAROL I" National Defence University Publishing House, 2017.

ISSN 2066-026X ISSN-L, 2066-026X, ISSN CD 2343 – 7669.

[3] K. Czakóová, L. Végh. Tile-based programming languages and virtual environments for program codes simulation. *In. INTED2023: Proceedings of the 17th International Technology, Education and Development Conference*. DOI: https://doi.org/10.21125/inted.2023.1374, pp. 5308-5313. Valencia: IATED Academy, 2023. ISBN 978-84-09-49026-4. ISSN 2340-1079.

[4] O. Takáč, L. Végh. Usage of UAVS in the protection of cultural heritage in the teaching of computer science. *In: INTED2021 Proceedings. 15th International Technology, Education and Development Conference.* DOI: 10.21125/inted.2021.2084, pp. 9987-9992, Valencia: IATED Academy, 2021. ISBN 978-84-09-27666-0. ISSN 2340-1079.

[5] J. Udvaros, O. Takáč. Developing computational thinking by microcontrollers. *In: ICERI2020 Proceedings. 13th annual International Conference of Education, Research and Innovation.* DOI: 10.21125/iceri.2020.1474, pp. 6877-6882, Valencia: IATED Academy, 2020. ISBN 978-84-09-24232-0. ISSN 2340-1095.

[6] O. Takáč, D. Hrubý, V. Cviklovič. Possibilities of navigation of mobile agricultural robots on the principle of the geometrical objects detections. *UNIV Agricultural Sciences & Veterinary Medicine Bucharest*, Vol. 11, pp. 206-208, Bucharest: USAMV, 2011. ISSN 2284-7995.

[7] L. Végh, O. Takáč. Using interactive card animations for understanding of the essential aspects of non-recursive sorting algorithms. *In: Janech J, Kostolny J, Gratkowski T, eds. Proceedings of the 2015 Federated Conference on Software Development and Object Technologies: SDOT 2015.* Advances in Intelligent Systems and Computing, vol 511. DOI: 10.1007/978-3-319-46535-7_25, pp. 336-347, Springer International Publishing, 2017.

[8] O. Takáč, L. Végh. Creation of 3d models of real objects in the teaching of computer science. *In: ICERI2021 Proceedings. 14th annual International Conference of Education, Research and Innovation.* DOI: 10.21125/iceri.2021.1291, pp. 5723-5727, Valencia: IATED Academy, 2021. ISBN 978-84-09-34549-6. ISSN 2340-1095.

[9] L. Végh, O. Takáč. Mobile coding games to learn the basics of computer programming. *In: EDULEARN21 Proceedings. 13th International Conference on Education and New Learning Technologies.* DOI: 10.21125/edulearn.2021.1590, pp. 7791-7799, Valencia: IATED Acedemy, 2021. ISBN 978-84-09-31267-2. ISSN 2340-1117.

[10] V. Stoffová, M. Zboran. Educational Robotics in Teaching Programming in Primary School. In: Singh, Y., Singh, P.K., Kolekar, M.H., Kar, A.K., Gonçalves, P.J.S. (eds) *Proceedings of International Conference on Recent Innovations in Computing : ICRIC 2022, volume 1*. DOI: 10.1007/978-981-19-9876-8_51. Lecture Notes in Electrical Engineering, vol. 1001, pp. 669-682, Singapore: Springer, 2023. Print ISBN 978-981-19-9875-1, Online ISBN 978-981-19-9876-8.

[11] V. Stoffová, V. Gabaľová. Interactive simulation models for teaching and learning of computer science. *In. INTED 2023 : In. INTED23 Proceedings : 17th International Technology, Education and Development Conference.* DOI: 10.21125/inted.2023.1050, pp. 3945-3953, Valencia : IATED Academy, 2023. ISBN 978-84-09-49026-4. ISSN 2340-1079.

[12] V. Gabal'ová, M. Karpielová, V. Stoffová. Beginners Online Programming Course for Making Games in Construct 2. In: Singh, Y., Singh, P.K., Kolekar, M.H., Kar, A.K., Gonçalves, P.J.S. (eds) *Proceedings of International Conference on Recent Innovations in Computing : ICRIC 2022, volume*. DOI: 10.1007/978-981-19-9876-8_41. Lecture Notes in Electrical Engineering, vol 1001, pp 547–558, Singapore: Springer, 2023. Print ISBN 978-981-19-9875-1, Online ISBN 978-981-19-9876-8.

[13] D. Dancsa, I. Štempeľová, O. Takáč, N. Annuš. Digital tools in education. *In. International Journal of Advanced Natural Sciences and Engineering Researches*. Vol. 7 No. 4 (2023) : IJANSER. pp. 289-294. DOI: https://doi.org/10.59287/ijanser.717. E-ISSN 2980-0811.

[14] N. Forman, M.S. Avornicului, L. Szabó. Unlocking Success in IT: A Systematic Exploration of Soft and Hard Skills and Their Collaborative Dynamics across Information Technology Roles. *In. International Journal of Advanced Natural Sciences and Engineering Researches*. Vol. 7 No. 4 (2023) : IJANSER, pp. 246-252. DOI: https://doi.org/10.59287/ijanser.708. E-ISSN 2980-0811.

[15] L. VÉGH, K. CZAKÓOVÁ. Possibilities of using games in teaching and learning the basic concepts of object-oriented programming. *In. INTED23 Proceedings: 17th International Technology, Education and Development Conference.* DOI: https://doi.org/10.21125/inted.2023.1383, pp. 5329-5334. Valencia: IATED Academy, 2023. ISBN 978-84-09-49026-4. ISSN 2340-1079.

[16] H. Hyksová, V. Stoffova, V. Gabaľová. Teaching robotics and online programming using a virtual robot. *In. INTED 2022 Proceedings : 16th International Technology, Education and Development Conference*. DOI: 10.21125/inted.2022.1812, pp. 7140-7147, Valencia: IATED Academy, 2022. ISBN 978-84-09-37758-9. ISSN 2340-1079.

[17] D. Paksi, M. Csóka, N. Annuš. An Overview of Modern Methodological Approaches of it Education. In. EDULEARN22 - 14th International Conference on Education and New Learning Technologies. DOI: 10.21125/edulearn.2022.1363, pp. 5812-5817. Valencia: IATED Academy, 2022. ISBN 978-84-09-42484-9. ISSN 2340-1117.

[18] M. Csóka. Data Visualization as Part of High School Programming. *In. EDULEARN21: Proceeding of the 13th annual International Conference on Education and New Learning Technologies*. DOI: 10.21125/edulearn.2021.1916, pp. 9484-9489, Barcelona: IATED, 2021. ISBN 978-84-09-31267-2. ISSN 2340-1117.

[19] A. Katyetova, V. Stoffova. Analysis of ict competencies and skills of primary schools informatics teachers. In. INTED2023 Proceedings: 17th International

Technology, Education and Development Conference. DOI: 10.21125/inted.2023.1073, pp. 4040-4048, Valencia: IAED Academy, 2023. ISBN 978-84-09-49026-4. ISSN 2340-1079.

[20] Gubo S, Kmet T, Molnar A, Takac O. A MULTI-RANGE APPROACH FOR CULTURAL HERITAGE SURVEY: A CASE STUDY OF A MEDIEVAL CHURCH IN SLOVAKIA. In: 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI). ; 2020:000117-000122. doi:10.1109/SAMI48414.2020.9108724