p-ISSN: 2184-044X e-ISSN: 2184-1489 ISBN: 978-989-53614-5-8 © 2022
https://doi.org/10.36315/2022v2end090

RESEARCH ON ONLINE PROGRAMMING EDUCATIONAL TOOL: CASE STUDY ON A THREE-IN-ONE ENVIRONMENT

Yu-Wen Pu[1], Po-Hsun Cheng[1, 2], & Li-Wei Chen[1, 2]
[1]*Software Engineering and Management Department, National Kaohsiung Normal University (Taiwan)*
[2]*Information Education Center, National Kaohsiung Normal University (Taiwan)*

Abstract

In the field of information education, the introductory course of programming is a threshold. It is often the first course faced by students with no or little background knowledge. For learning unfamiliar and difficult things, students will inevitably be vulnerable and give up their learning. Hence, it is necessary to help students correctly cultivate the concept of computational thinking to assist them in their learning process to reduce learning difficulties as well as enhance interest. Our research team designed an online programming educational tool that is based on Python, Scratch, and activity diagrams. The purpose is to help students learn the basic concepts in the introductory programming courses. Users can just connect to the website to learn and many convenient functions are added to record the learning process. The Python part is provided for non-primary school students to learn and the Scratch part is for primary school students to learn logical concepts. Also, there is an activity diagram to increase students' interest in learning and reduce learning difficulties. The questionnaire data were collected at the beginning and end of the course and the results were finally analyzed. By analyzing the questionnaires collected and analyzing the data content, we can see that the results of this experiment have improved the students' learning process. However, the operation of the tool has also received feedback from students. Our designed features need to be closer to the user, and it is expected to assist more learning content, not only at the beginning of programming but also in the future to assist in more advanced programming learning. Compared with traditional teaching, this research uses tools to assist learning so that students can better understand the working principle of coding.

Keywords: *Digital learning, information education, programming language, Python, Scratch.*

---

1. Introduction

Computational thinking ability was proposed and advocated by Professor Jeannette M. Wing (Wing 2006) of Carnegie Mellon University in the United States to define the concepts needed to solve problems effectively in different situations. This concept is not necessarily an ability necessary for daily life the same, but it can effectively understand and convey the concept of the computer, and it is very important to the process of programming, which is the basic concept of all educational courses in the field of information. Cultivating students' computational thinking ability in education can also train students' ability to identify and solve problems, which can not only be used in the field of information education.

Some studies have found that interactive learning materials are more useful than the use of programming textbooks. Students will have insufficient learning motivation due to difficulties in learning (Tan et al., 2009). Lack of a sufficient learning environment will lead to increased difficulties for students to learn programming. Therefore, it is necessary to create a learning environment for students to improve their learning motivation (Hwang et al., 2012).

We have developed a systematic tool to assist beginner programming learners so that they will not be discouraged by difficulties or difficulties when entering the programming field. We know from the literature that visual tools are useful for learning and help students better understand the programs they execute, so we add visual functions to the system to help students understand difficult programming languages and train students. Understand operational thinking concepts.

This research is to put the developed tools into the physical courses, conduct experiments, observe and analyze the collected data, and understand whether the tools achieve their goals.

2. Methods

We have developed an auxiliary tool for assisting programming learning. It consists of three functions: Python, Scratch, and activity diagram. The purpose is to use it in basic programming courses to help students learn basic programming functions and concepts. Just connect to the website. Learning can be carried out easily, and many convenient functions are added to record the learning process. This system quickly calculates and displays the results through the real-time calculation function.

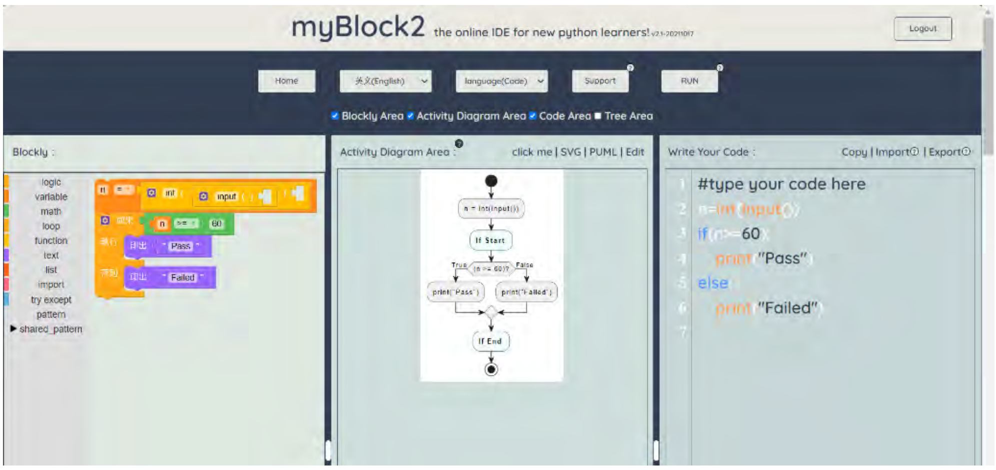
The tools developed in this study were put into the basic course of programming, integrated into the practical content of the course, observed the learning progress of students, collected and analyzed the questionnaire results, and observed whether the system tools were helpful to students' learning, and also Tools for review and improvement.

2.1. Tools

The teaching tool of this study is a platform built on a web page, which can be provided to users and write python language on the platform or write programs in a visual way such as scratch, and one of the two is converted through the system when writing. Another language is also presented on the page visually as an activity diagram so that students can understand the grammatical structure.

For non-undergraduate students, learning programming is mainly to cultivate the concept of computational thinking, often using visual programming language to combine methods in a building block-like way, which is easier to learn than directly typing code; in addition to basic computational thinking, undergraduate students also need to cultivate professional skills. learning programming languages is even more important. The tools are designed with two blocks, namely the block for writing Python, the Blockly area, and the Scratch-based block code area, and there are two other blocks, the activity diagram area, which renders the program as an activity diagram.

Figure 1. Assisted Program Learning System.



2.1.1. Blockly Area. The Scratch language developed by MIT converts grammar into visual building blocks and develops by stacking building blocks, making the programming language more intuitive, reducing the difficulty of memory, and making it easier to debug. We use this method to make a building block language and operate according to the rules of Python syntax.

2.1.2. Code Area. An area is written in the Python language, you can choose to execute it after completion, and the result will be displayed in the Console. It has basic compiler functions and can import and export the completed program for simple debugging.

2.1.3. Activity Diagram Area and Tree Area. The two areas can convert the written language into Activity Diagram and Syntax Tree in time. Users can observe the grammar structure filled in by themselves while writing, and strengthen the concept of grammar and logic.

2.2. Course

We put the tool into the programming basic courses of two different schools. There were 13 and 22 students participating in the research, including 20 freshmen, 10 sophomores, 2 juniors, and 4 seniors. Except for one double-major student, the others are major students, and most of them are freshmen. They do not have perfect operational thinking and logical concepts in the field of information, and they are the auxiliary objects of the tools developed.

We teach students to use tools at the beginning of the course and practice with practical content in the course. In order to analyze whether the tool is beneficial to learning and its operability of the tool for novices, a pre-test questionnaire is conducted before the course starts to use the tool. After using the tool and filling out a questionnaire before the end of the course, we use the google form to fill in online, which reduces the trouble of paper collection and facilitates the centralized processing of data.

2.3. Questionnaire

According to the literature, H. Rex Hartson's Affordance classification on human-computer interaction, we designed three types of questions in the questionnaire, including Cognitive Affordance, Sensory Affordance, and Functional Affordance, in addition to verifying whether the tool will have a positive impact on the user, also needs to analyze the operability of the tool through data, as a reference for the subsequent development of the tool (Hartson, 2003).

The questionnaire contains 15 Cognitive Affordance items, which are about whether the user can understand the meaning of the object and the purpose of each function. The Sensory Affordance item is about the user's perception. Through the perception of the tool to understand his Affordance, there are a total of 10 items, and there are 2 Functional Affordance items. Whether the entity operation of the tool makes the user feel the corresponding functionality, we design the options of each item according to the Likert scale of 11 points (Lewis et al., 2017).

2.4. Date analyzing

In order to understand whether the tool has an effective impact on the user, we divide the questionnaire into pre-test and post-test. Students take a pre-test before using it, and then take a post-test after using and using it for learning. First, use the SPSS analysis tool to analyze the scale. The validity and reliability were analyzed using Pearson product-moment correlation to test the validity of each item for its corresponding construct. The reliability of these items was analyzed using Cronbach's alpha coefficient.

3. Discussion

According to the analysis of SPSS, bivariate correlate the data and obtain the correlation between each item, as well as the two-tailed significant p-value. Based on the obtained p-value, $p<0.05$ for all items can be observed, and the quantitative scale can be concluded. The item is valid data; since the number of people in the data, the survey is 35, when N=35, the 5% value of the search significance on the r table product moment is 0.334. When the correlation coefficient is greater than 0.334, the item is valid. The values of the Correlations coefficient and the correlation coefficient of the total score of each item are all greater than 0.334, which shows that these items are effective.

After the data is tested for reliability, we obtain Cronbach's alpha coefficient. When $\alpha \geq 0.7$, the data is reliable. When analyzing the data, the coefficient of each item is listed (Cronbach's Alpha if item deleted). Information is reliable.

Table 1. Validity and reliability tests for each item.

Construct	Item Number	Validity Test (Correlations)	Reliability Test (Cronbach's Alpha if Item Deleted)
Cognitive Affordances	1	0.865**	0.989
	2	0.889**	0.989
	3	0.837**	0.989
	4	0.901**	0.989
	5	0.929**	0.989
	6	0.896**	0.989
	7	0.918**	0.989
	8	0.877**	0.989
	9	0.922**	0.989

	10	0.918**	0.989
	11	0.924**	0.989
	12	0.895**	0.989
	13	0.939**	0.989
	14	0.908**	0.989
	15	0.821**	0.990
Sensory Affordances	16	0.843**	0.989
	17	0.764**	0.990
	18	0.828**	0.989
	19	0.919**	0.989
	20	0.874**	0.989
	21	0.906**	0.989
	22	0.844**	0.989
	23	0.742**	0.990
	24	0.857**	0.989
	25	0.935**	0.989
Functional Affordances	26	0.899**	0.989
	27	0.921**	0.989

$p > 0.05$, $p \leq 0.05$, $p \leq 0.01$, and $p \leq 0.001$ indicate no significance and significance at the levels of 0.05(*), 0.01(**), and 0.001(***), respectively.

4. Discuss

After analyzing the data, we observed the reliability and validity of the data. We observed the data of the pre-test and post-test, as well as the feedback from the students in the classroom, and we can understand that the tools are helpful to the students.

However, this study only conducted one course and conducted operations in two different teaching environments. It is impossible to judge whether the tool has a more positive effect on students' learning through the students' performance.

5. Conclusions

With the rapid development of information technology, the social environment, and information technology have become inseparable, so school education must keep up with it and teach students the concept and thinking of information. However, the traditional education system cannot be changed overnight. Both students and teachers need to make efforts. The tool system we developed is to assist students in learning in basic programming courses, arouse their interest in learning, and enable students to better understand the concept of computational thinking through visual images.

Through the analysis of research data, it can be seen that for students, this tool has achieved its goal, which is to assist program learning, reduce the difficulty of learning, and make it easier to cultivate computational thinking and program logic.

There are still some limitations in this research. The research data is collected for beginners in programming, but they are all undergraduates. It should be possible to choose younger students such as high school students or even middle school students. In addition, our tools can be more interactive. With a visual interface, it will be easier to operate for younger users.

Acknowledgments

The authors express thanks for partial financial support from both of the Ministry of Science and Technology, Taiwan, under grant numbers MOST 110-2629-E-017-001 and the “STEM Field and Female Talent Education” project, Ministry of Education, Taiwan. Also, we thank all research colleagues in our joint laboratory, HIE+SM, SEM, NKNU, who have supplied their comments and ideas for us to achieve the research goal.

References

Hartson, R. (2003). Cognitive physical sensory and functional affordances in interaction design. *Behaviour & Information Technology*, *22*(5), 315-338.

Hwang, W. Y., Shadiev, S., Wang, C. Y., & Huang, Z. H. (2012). A pilot study of cooperative programming learning behavior and its relationship with students' learning performance. *Computers & Education*, *58*, 1267-1281.

Lewis, J. R., & Erdinç, O. (2017). User experience rating scales with 7, 11, or 101 points: does it matter?. *Journal of Usability Studies*, *12*(2), 73-91.

Tan, P. H., Ting, C. Y., & Ling, S. W. (2009). Learning difficulties in programming courses: undergraduates' perspective and perception. In *Computer Technology and Development, International Conference on ICCTD'09*, vol. 1, pp. 42-46.

Wing, J. (2006). Computational Thinking. *Communications of The ACM*, *3*(49), 34-35.