2ZC-02

An interactive environment to teach programming based on chain learning

Nikolai Shumilin Ian Piumarta Ryo Nishide Hideyuki Takada
Ritsumeikan University

1. Introduction

To learn programming requires several skills, such as knowledge of syntax and the ability to construct algorithms. In many cases this is a big problem, since in most reference training programs are a set of commands and their description, but no description of how and in which cases they should be used. Another method for learning is the use of special guides. The main problem in this case is that if a version of the programming environment changes some parts of the guide become obsolete. The next way is to use a sandbox for programming. But this way again requires the use of a the guide or presence of a teacher. Since all of the elements in the sandbox are available to the new user who does not understand how to use them, newbie programmers may be confused.

Our environment addresses these problems by using a *chain learning* structure to teach programming where syntax is not introduced until necessary. It can be used to teach children (beginner levels are designed for children), and it works everywhere (written in JavaScript for the browser).

2. Related work

There are several tools for teaching programming [1]. Some of them are big sandboxes, which can be programmed for a variety of activities. These are the environments such as Scratch [2], LOGO, and Robomind. There are also highly specialized software environments for a particular language that teach programming in only one language using a particular syntax. These are the environments such as Javarush and Codeacademy for specific programming languages.

In order to learn how to program in a big sandbox, you need to find a guide for programming or a teacher. When using these environments for the first time, the user does not immediately understand what he needs and what he should learn. When a user is provided many functions at once, there is a possibility that he tries the basic functions, but cannot understand more complex functions.

With highly specialized environments, learning complex syntax is another problem. The user solves a sequence of problems, with the help of which the user learns programming, from the simplest concepts such as loops and variables to complex high-level programming. But even the simplest of these problems can be solved only with a specific syntax, and complex syntax causes problems in the education of children.

3. Method

Our goal is to create a software environment where programming will be learned with the help of certain tasks, but independently of the program syntax. For realization, it was decided to use a chain learning structure. The chain learning structure is a sequence of rules and skills that build incrementally upon those learned earlier.

This structure helps to explain to the user the basics with the help of which he will continue to solve more complex problems [3]. This structure involves more than the user in the programming process, as tasks gradually become more complicated, but the user can solve each of them easily thanks to previous skills learned [4].

In our environment, each link in the chain is a single programming section (**Figure 1**) composed of several tasks. As an example, in our environment in the first level, the user learns simply how to move an object. In the next level, he moves the object in a specific way to solve a more complex task. Then, the user learns cycles, and in the corresponding task uses cycles to control the movement of the object. Every learning step builds upon the previous ones. The problems gradually become more complex, and in our environment they correspond to various *levels* of training.

The first level is the level of a beginner. At this level, users are offered the simplest problem to solve, such as moving an object to a finish line by using navigation buttons. We create a button for each specific program action. When you click on a button the corresponding program command appears in the console. This button interface is useful for those users who do not initially understand what commands must be entered. Thus, the user will remember the commands and on these levels will only see them in the console, and the user will feel comfortable. For those who cannot solve this problem, the screen displays a hint button. With this button, the user can see the a little hint to solve the task. Levels are constructed with the condition that the user must understand only one additional concept to achieve the goal. For example, the first level will be a challenge for navigation. To solve it, the user understands that it is necessary to press one button to move and reach the goal. The next level has more action buttons, so that the user can himself determine how to achieve a more complex goal. That is, each level is built on the basis of the previous level's knowledge.

The second level is the intermediate level. At Intermediate level, users are offered more challenges than at the beginner level (tasks becomes more complex, and actions can be selected both from buttons or written explicitly using a simple syntax familiar from the beginner level). The third level is advanced level. At advanced level, users are offered complex problems whose solutions require the skills of the previous levels. (All the actions are written in the user console, using the familiar syntax). All these levels are necessary in order to separate the possible knowledge levels of users. To allow them to start at an appropriate level otherwise users who know programming will be bored if required to start at the lowest levels designed for beginners.

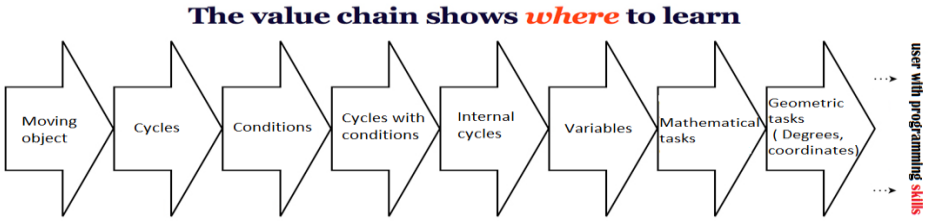

Figure 1: Stages of Chain learning structure in the environment.

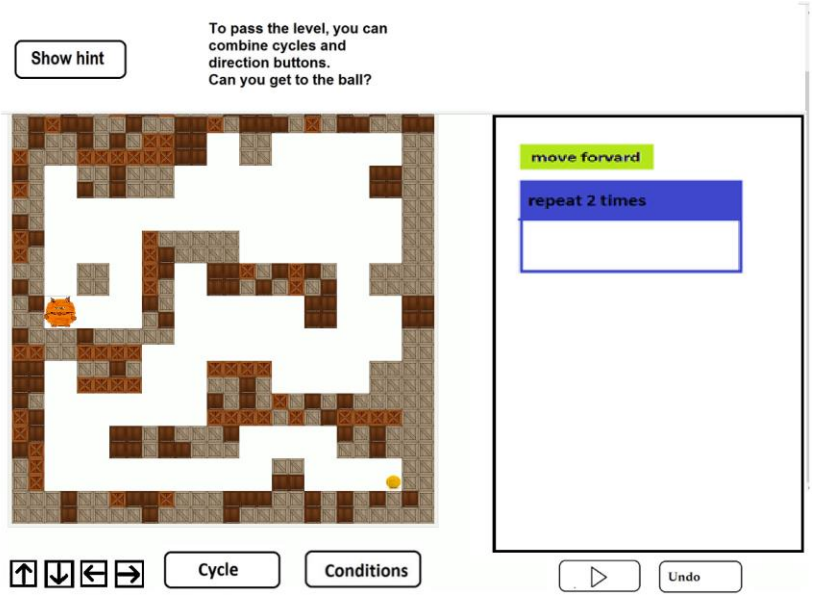

Figure 2: Environment interface

4. Implementation

To create a software environment, scripting language JavaScript was used. The program is written in this programming language to run on any device, so that users do not need to have a special device. Another advantage of this programming language is that using this language you can create absolutely any programming constructs. When creating a software interface (**Figure 2**) we take into account features such as ease of use and comprehension. Since beginners should understand what they need to do, the main task is to design the levels so that the user does not feel bored. This will make the learning process more effective

Early prototype will comprise three stages with various mechanic in each section (beginners, intermediate, advanced). This is necessary to show how the chain learning works in this environment. These levels will be used on the basis of which the chain learning can be to conduct an experiment to evaluate the efficiency of the training compared to normal training.

Conclusions

In this environment, programming is presented in a simplified form as a sequence of tasks. Each new task will be more complicated than the previous. Clearly, beginners need to feel that they can make gradual and continual progress in learning to program by solving simple tasks. However, difficult tasks are not the only reason people hesitate to learn programing; complicated syntax/environments or too many concepts presented in an illogical order can also prevent successful learning.

Performing simple tasks, step by step, can teach the user basic programming skills without knowledge of syntax. The same chain learning method can be applied to more complex programming tasks, thereby creating interesting challenges, not only for beginners but also for experienced programmers.

References

[1] Caitlin Kelleher and Randy Pausch, Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers, ACM CSUR 37(2), 2005.
[2] John Maloney, Mitchel Resnik, Natalie Rusk, Brian Silverman, and Evelyn Eastmond, The Scratch Programming Language and Environment, ACM TOCE 10(4), 2010.
[3] Aliza Aufrichtig, The Prerequisites and Privilege of Autodidacticism: What you already need to know to teach yourself, ACM XRDS 20(4), 2014.
[4] Lara-Porras, A.M. Lozano-White, L. Interactive Teach-Yourself Resource, International Conference on Education and Management Technology, 2010.