The International Journal of E-Learning and Educational Technologies in the Digital Media (IJEETDM) 2(1): 31-36
The Society of Digital Information and Wireless Communications (SDIWC), January - 2016 (ISSN: 2410-0439)

A Study of Teaching Problem Solving and Programming to Children by Introducing a New Programming Language

Seyyed Meisam Taheri , Minoru Sasaki, Jiangcheng Oliver Chu, Harrison Thuku Ngetha

Department of Human and Information Systems Engineering, Gifu University, Yanagido, Gifu-shi, 501-1193, Japan
s3812005@edu.gifu-u.ac.jp , sasaki@gifu-u.ac.jp , j.oliverchu@berkeley.edu , s3812007@edu.gifu-u.ac.jp

ABSTRACT

Technology has been growing rapidly, and this fact that computers, gadgets and electronic devices have become part of our life is deniable. Computer skills and programming languages have become as important as other essential subjects in schools. Researchers have done some researches in the area of teaching programming and problem solving to novices, this study aims to study whether it is necessary to teach problem solving and programming in early ages. This study also aims to review previous studies in this area. We attempt to introduce a new programming language which has been developed to simplify the method of learning and coding for novices and young learners. We have been developing a new programming language which the children and young learners considered to be the learners of this language; however professional programmers would be able to code with this language as well. There are many popular and powerful languages, Python for instance is easy to understand and code with, however it is still complex and difficult to learn for children, and on the other hands programming languages like Java and C++ have their complexity. The target users are children who have no knowledge about programming. Beside some inappropriate teaching methods and resources, the complexity of programming languages and lack of knowledge about problem solving, young learners find programming terrifying. Having a programming language with a simple syntax and methods will encourage students and failure ratio will be decreased. Children will have a chance to learn how to code and solve a problem by using and interacting with an easy programming language. By doing this research and using valuable resources, we would expect to have positive and effective changes in science and programming in near future, as the today's children are the future of our society.

KEYWORDS

Technology, Programming, Learning, Novice Programmers, Programming Languages, Problem Solving

1 INTRODUCTION

Computing is considered as a career of 21st century and things around us these days are mostly computerized and computer based. Nowadays, students face a complex and changeable world. Students required to know new ideas, creativity and problem solving techniques to solve unconventional problems, however not everyone necessarily should become a problem solver and programmer [1]. In order to avoid the information explosion and to improve students' problem solving skills, improving their systematic ways of thinking, problem solving and programming skills will be essential. Children's capability to learn and do programming is not something new and it has been proven by some researchers which we are going to discuss in this study. Teaching programming to novices is a difficult task down to the complexity of the subject, difficult initial programming courses and negative views associated with programming which discourage students to pay enough attention to learn programming [2]. The reason of importance of Programming is that it helps the algorithmic thinking, a unique and different way of thinking for those who face problems in different disciplines [3].

Programming tools like Alice and Scratch are tools that can be used as a different way of teaching programming to children [4, 5]. These programming language tools use a visualized

environment with a drag and drop features and typing mode are minimized in them. They will let the user to create an animation based inputs, play games, or share videos through the internet. Green foot is another programming environment intended to teach young beginners and older novice users to do coding in Java. The framework is using a visualized two-dimensional grid which allows user to code [6]. Programming environments mentioned above need the ability to write and read which makes them difficult for children to learn and use. As other researchers and we believe, children have the ability to understand some programming concepts even if they are not yet at the reading stage, however as we discussed in previous studies problem solving techniques and learning the basic of programming are essential and need to be taught and learn beforehand. Novices and specially children need a simple environment and language that allows them to create and debag a program, practice the problem solving in less textual, easy syntax or visual mode. PiktoMir is one example of such environment that allows children to program a virtual robot behavior by using few pictograms [7, 8]. Kiss experience shows the LEGO-Mindstorm is a good tool for learning programming, because the students can interact with a robot and use its different functions and write programs and not facing any syntax error [9].

The structure of this paper is as follow: in the Literature review section, we review relevant works which have been done in the area of this research. In the section of introduction of the proposed language we will discuss and introduce the new programming language which we temporary named it Mint language. We will also review and discuss different aspects of Mint and the ways that new programming language will be helpful in assisting children to learn coding.

2 LITERATURE REVIEW

2.1 PiktoMir

PiktoMir is a visual environment which has been developed to teach programming to young learners. This program has a story about a robot which is living in PiktoMir and it presents by the instructor as an introduction to motivate children. The robot named Fidget cannot move and do any action autonomously, and it should cover coating damages which occur when the shuttles launch. Fidget is able to move and do some actions according to the commands which input by users, some commands are as following: FORWARD, TURN RIGHT, TURN LEFT, and FILL. Fig1 illustrates the programming environment and its features.

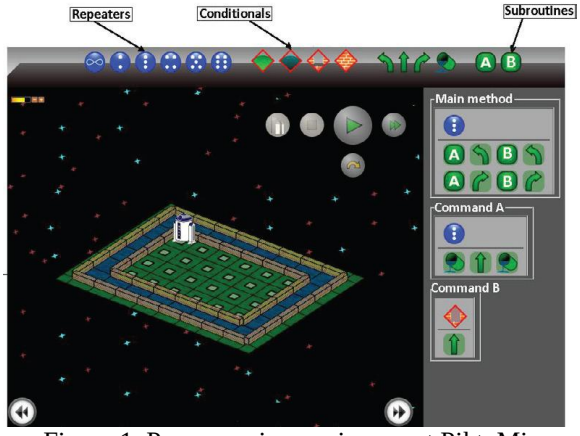

Figure 1. Programming environment PiktoMir

PiktoMir's author conducted a study at the Moscow kindergarten. Participants in this study were 42 pre –school students, 22 boys and 20 girls between 5.5 to 7 years old, as well as, 35 kids aged between 6.5 to 7 years old were attended as a senior group. Experiment has been done during 8 weeks with two consecutive classes (20 min each class) every week. In the beginning of the course, kids learned about introduction of PiktoMir and games without using computer, for the next step they started to practice everything they learned with computer. In the end of the all sessions, kids were given a test to check their comprehension of the basic of programming concept. The test contained three blocks of tasks. The first block focused on linear programs. The test divided to three sections, tasks to use subroutines, loops and linear programming. Kids were required to navigate the robot by following the instructions (Fig 2). These tasks were designed to examine the ability of children to execute the algorithm.

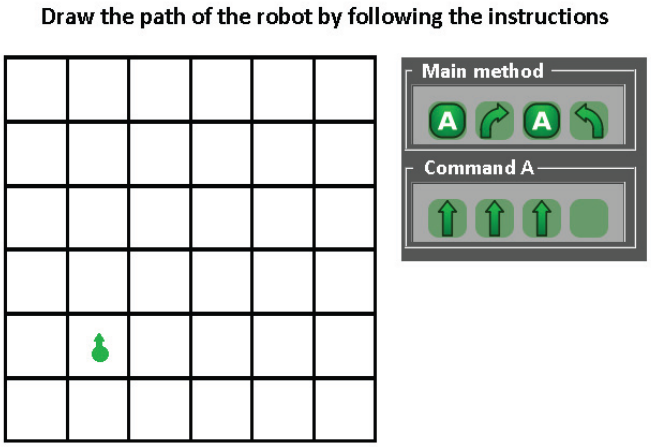

Figure 2. This task was contained in the block focusing on using subroutines

The result of this experiment is to focus on a specific programming concept skill. After accomplishing some tasks almost all children except one of them (41 of them) learned how to write a linear program. 75% of them learned definition and use of loops and sub routines. Seven of children were younger than six years old which two of them were able to pass the test. This allows the authors to assume that children younger than age 6 have some difficulties to understand the programming concepts like loop. This research shows that PiktoMir can be used to teach basic of programming to preschoolers. In addition, most children found this program fun and enjoyable to use [3].

2.2 Alice

Alice has been developed as a project named "Learning to Program with Alice" in Carnegie Mellon University (United States) which aims to bring young people into the world of programming, through a completely visual 3D and animated environment [10]. As the children are fast and visual learners, learning Programming in a visual and 3D tool which changed from a text base programming environment to a visual entertaining model will be really effective way for them to learn with. Users learn more about object orientated programming as they can see a graphical representation of an Object, loop statements and some other statements. Alice interface gives the user the ability of dragging items which represents the actions or statements. In fact, by using this type of programming, students do not face syntax errors and some text base programming problems that affect the motivation of students in programming courses [11]. Fig 3 illustrates the user interface of programming Alice.

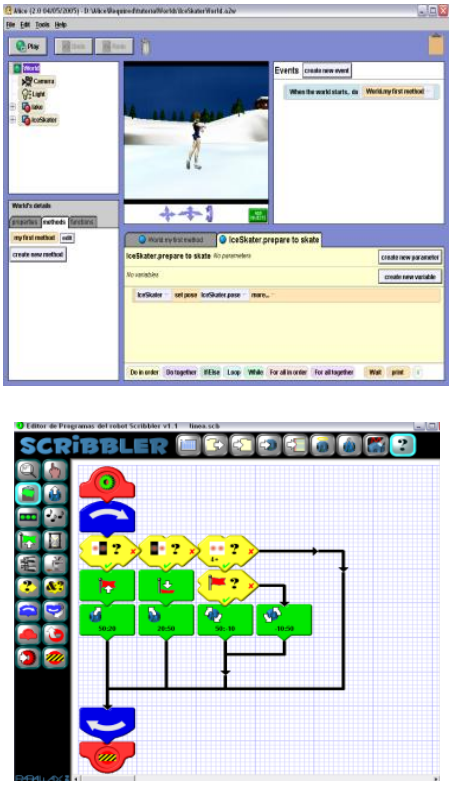

Figure 3. Alice and Scribbler Programming Interface

2.3 Scribbler

Scribbler is a robot which has the ability to be programmed, and some sensors indicate in this robot to detect object to find its path. This robot has a visual, graphical and user–friendly interface which allows user to program the robot easily; however traditional programming is still available and possible in this robot and some of languages like basic, Python and Java are supported. By interacting with Scribbler students will have a different experience and method to learn traditional languages. Students implement a code and control the robot through the interface, which is operational with icons, and by following the algorithm structure they will learn how to modify the robot's behavior [12].

2.4 Scratch for Budding Computer Scientists

Scratch is a programming environment developed by MIT's media Laboratory which allows user to create his/her own animation, game and interactive art. Scratch proposed to be as a first language for novice programmers in fundamental programming courses. The significant part of scratch is that it gives the user ability to program by using mouse; programmatic constructs are puzzle pieces which only can be fit together if they are syntactically correct. Scratch can be a gateway to other languages like python. This study goal was not to improve scores but instead to improve first-time programmers' experiences.

The result of this study was not only that Scratch excited students in the first time of use, but it also familiarized them with fundamentals of programming without any interfering of syntax. A survey conducted in this study to find out the effectiveness of Scratch in learning Java, most students (76%) responded a positive influence, mainly student without programming background. Those students (16%) who believed that Scratch was not effective to learn programming, all of them had programming experiences [13].

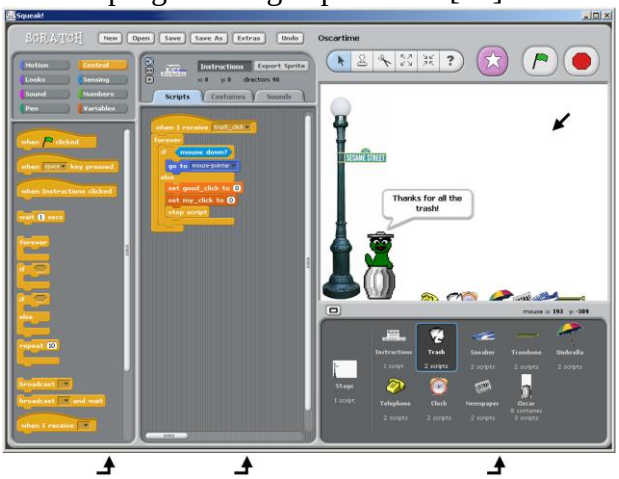

Figure 4. Scratch's interface [13]

3 MINT PROGRAMMING LANGUAGE INTRODUCTION

3.1 Aim of the language

The aim of this language is to implement an object-oriented language with beginner friendly syntax. As well as, an embeddable language usable wherever Java (more specifically, the Java Virtual Machine) can run, which is essentially on any modern operating system. The design philosophy of this language gives as much freedom and control to the programmer as possible, to let them build the application they want to build. The language itself is easy to learn, run and implement as this language described before aimed to be use for basic and young programmers, however in the advance mode of this language programmers are able to code and implement their advance codes.

3.2 How the programming language works

- The programming language (Mint language) is directly interpreted, using an interpreter written in Java.
- Mint lists and tables can hold objects of any type, but Java is statically typed with Array Lists only being able to hold a series of objects of the same type. Because of this type problem, all Mint objects aside from integers are actually referenced using Pointer objects – each pointer contains type information and the address of the Mint object. Integers are stored directly in the pointers, should be unchanged.
- To evaluate an expression like x = 4 + 3 * 5, Mint searches for expressions that can be simplified first, such as 3 * 5 becomes 15, and then expressions that can be simplified later, such as 4 + 15 becomes 19. Then 19 will be stored into x. Therefore, order-of-operations is always followed.

When creating user-defined objects, there is actually no difference between the current scope (a namespace for user variables) and a freshly created object. In fact, both variable scopes and objects are implemented using a Java class called MintObject. Scopes can be transformed into objects by using "return this" inside a subprogram, and variables inside an object can be put back into the current scope using "inherit object_name". This may be a little advanced for children and/or beginners who are first learning the language, however after learning the basic of programming they will be able to upgrade their knowledge and start learning the object –oriented programming in advance courses.

When you divide two integers, instead of rounding 1/3 to 0.333333… the language will actually store the numerator 1 and the denominator 3 inside a single object. This is for improved internal accuracy of calculations. However, the internal fraction 1/3 will naturally print as 0.3333333... to avoid unwanted behavior and to avoid confusing beginners. Most languages do not do this, and simply round the numbers immediately, which can cause additional accuracy problems.

3.3 Advantages of the language

Language is very simple and syntax is clear and clean. Instead of excessive symbols, clearly chosen keywords make the language easy to read.
"Functions" (which are called subprograms) are first class which can be passed around like regular objects.
Lists can hold objects of any type (heterogeneous sequence) instead of limiting yourself to an array that must hold things of the same type.
Tables (which implement HashMap-like functionality) do not have some of the limitations other languages impose. Mint can have multiple identical keys in tables, etc. Object system is simple and understandable.

3.4 Disadvantages of the language

We strongly believe that there are languages that are very powerful and bug free. As this language still in the developing phase it still has some problems. The problems and issues we already have found and we are trying to improve it, here called as disadvantages of this language.

When dealing with large amounts of data processing, Mint language runs more slowly than other languages. Mint is directly interpreted, and does not compile to virtual byte code and this is another reason for its slowness.
Table syntax {["a", 1], ["b", 2]} is more verbose than Python/JavaScript's {"a": 1, "b": 2}.
Table modification is currently handled by adding extra pairs. This gives programmer more freedom but is more confusing than direct assignment like a["key"] = 3 in other languages.
Mint does not have the widespread and helpful community of Python, Java, Ruby, and other languages.

3.5 Structure and Syntax

Blocks are started immediately when control flow statements are used, such as if, while, for, and for-each. Blocks close with the keyword "end", similar to how it is done in Ruby.
No indentation issues like you might have in Python. Only newlines are significant for separating individual lines.
Lines can also be separated or terminated with a semicolon with no ill effects. This is for people who prefer this kind of syntax.
Functions (known as subprograms) start with the word "sub", and close with "end".
Lots of extra control flow. Traditional for-loop, for-each-loop, switch statements, and repeat-loop. "Repeat 12" will repeat the block 12 times which is easy to understand for beginners.
Functions can be nested inside each other. Functions can return other functions. Nested function is the way you put methods inside the objects. There is no class keyword - "sub" takes care of everything. Using "return this" will return a new object with all the methods the programmer will define.

3.6 Ease-of-use and user friendliness

The caret ^ does exponentiation like in mathematics or on a handheld calculator and it does not perform bitwise XOR, which may be confusing.
Implicit multiplication works. Like x = 2(2) + 8(10). Like in math, programmer does not have to use the star symbol if programmer does not want to use it.
In Mint, tables remember the order in which you place key-value pairs. Python and JavaScript forget the order, so when you pass over the dictionary or JS object, you may get the keys in random order.
Easy to access standard library, using the import keyword. Import "filename" will also import another file in the current folder. More

complicated imports like import "myfolder/stuff/something" can be used.

Mint is still a work in progress. In the future, Mint's syntax trees will be converted to bytecode for faster execution. Table behavior will be changed to better suit the programmer's needs, and to be more understandable. As well as, other features can be suggested and will be taken into consideration.

4 FUTURE WORK AND CONCLUSION

In this study, we have studied to identify importance of programming and problem solving in programming to be taught to children and young learners. The difficulty and complexity of programming languages and problem solving tools have been discouraging novices to learn and following programming as a passion. We have studied and reviewed other researches work, most researchers believe that it is a need to teach children and they also develop and design new methods to make it easy and possible for children. Furthermore, we have decided to design and develop a new programming language which give children and novices this opportunity to interact and learn programming language and problem solving with a simple and user friendly language environment. For the future work, we have plan to improve our language structure and environment, add some features and make it as simple as possible for novice programmers. As children are visual and fast learners, we are trying to make the language more visual like and the commands and syntax would be very close to English common vocabulary and statements.

5 ACKNOWLEDGMENT

Special thanks to staff of Gifu University and Sasaki. Ito laboratory members.

6 REFERENCES

1. Taheri, M., Sasaki M., Ngetha, H., " Evaluating the Effectiveness of Problem Solving Techniques and Tools in Programming " *Science and Information Conference (SAI), 2015*, vol., no., pp.928-932, 28-30 July 2015. doi: 10.1109/SAI.2015.7237253
2. Taheri, M., Yamamoto, H., Tripathy, H., Novel Assessment of Different Intelligent Tools for Problem Solving, *Computer Science and Engineering*, Vol. 3 No. 3, 2013, pp. 67-75. doi: 10.5923/j.computer.20130303.03.
3. Rogozhkina I., Kushnirenko A., PiktoMir: teaching programming concepts to preschoolers with a new tutorial environment, Procedia - Social and Behavioral Sciences, Volume 28, 2011, Pages 601-605, ISSN 1877-0428, available at : http://www.sciencedirect.com/science/article/pii/S1877042811025535)
4. *Cooper, S., Dann, W., Pausch, R. (2000). Alice: a 3-D tool for introductore programming concepts. Journal of Computing Sciences in Colleges,15, 5.*
5. Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., and Resnick, M. (2004). Scratch: A Sneak Preview. *Second International Conference on Creating, Connecting, and Collaborating through Computing*. Kyoto, Japan, 104-109.
6. Henriken, P., and Kooling, M. (2004). Greenfoot: combining object visualization withinteraction. *In Companion to the 19th annual ACMSIGPLAN conference on Object-otiented programming systems, languages, and applications*, 73-82.
7. Kushnirenko A.G. and Leonov A.G. (2009) PiktoMir: programming games for primary school children. http://www.ito.su/main.php?pid=26&fid=8248&PHPSESSID= 12d7b5d09fc001d29
8. Kushnirenko A.G., Leonov A.G., Roytberg M.A., Yakovlev V.V. (2010) PiktoMir: programming for preschoolers. http://window.edu.ru/window_catalog/files/r67202/Pereslavl-2010.pdf
9. *Kiss G.,: Using the Lego-Mindstorm kit in German Computer Science Education. 8th IEEE International Symposium on Applied Machine Intelligence and Informatics (pp 101-104). IEEE Xplore digital library Digital Object Identifier: 10.1109/SAMI.2010.5423759, (2010a)*
10. Cooper D., Pausch R., Learning to Program with Alice. Carnegie Mellon University, www.alice.org
11. Salcedo, S.L.; Idrobo, A.M.O., "New tools and methodologies for programming languages learning using the scribbler robot and Alice," *Frontiers in Education Conference (FIE), 2011* , vol., no., pp.F4G 1,F4G-6, 12-15 Oct. 2011
12. Salcedo ., Londoño S., Orozco Idrobo A., "New tools and methodologies for programming languages learning using the scribbler robot and Alice." *Frontiers in Education Conference (FIE), 2011*. IEEE, 2011.
13. Malan, David J., and Henry H. Leitner. "Scratch for budding computer scientists." *ACM SIGCSE Bulletin*. Vol. 39. No. 1. ACM, 2007.