# A Virtual Robot to support Programming Learning

João Tiago Aparicio, Instituto Universitario de Lisboa (ISCTE-IUL)., ISCTE-IUL, Lisboa, Portugal
Manuela Aparicio, Instituto Universitario de Lisboa (ISCTE-IUL) ISTAR-IUL, Lisboa, Portugal
Carlos J. Costa, ISEG (Lisbon School of Economics and Management), Universidade de Lisboa, Lisboa, Portugal

## Summary / Abstract

Computer programming is a valuable skill. But learning programming is sometimes difficult, time-consuming, and even frustrating. In this context, there is a need for new approaches to help in the process of learning programming. Here it is presented a possible solution. This solution is a progressive system, based on a virtual robot, supported in a generic and portable language. Those features allow answering to some needs. First, the virtual robot allows a certain level of result tangibility. Then, the use implementation as a progressive system allows for the support of the learning process. Finally, use of a generic and portable language allows for the installation of the system on many platforms.

## 1 Introduction

Nowadays, there is a growing need for people with programming-related skills. According to the digital economy report, only 10%, on average, of the population have programming skills, although more than 60% of working people say they do not have the sufficient skills to apply for new jobs. [17]. Many of the highest paying jobs require programming. On the other hand, many skills are supported in programming knowledge. For example, data science, mobile application development, artificial intelligence, machine learning or digital transformation are the knowledge that is related to computer programming. This situation is associated with the complexity of programmable devices, increasing number of devices that are programmable, the need for processing large quantities of data, the need to extract information from many devices or the necessity of developing mobile and web applications for the all the purposes. On the other hand, it appears that the process of learning programming is time-consuming and often painful. Some studies report students' difficulties in programming learning, leading to a need for new training techniques that improve the learning process. Based on previous research, this topic is not yet fully explored. In fact, there is still spacey for the proposal of new learning strategies. Therefore, in this context, the goal of this study is to reach a solution based on supported virtual robots. This study contributes to a new solution to enable students with a digital tool, in which learners can observe the immediate result of their programming effort, with an intuitive interface.

We start by presenting a brief literature review. In the following section, we present the solution. Then, we describe the implementation of the solution, illustrating for specific use cases.

## 2 Literature Review

### 2.1 Learning Programming - a Problem

The mismatch between supply and set of people with programming skills is a reality. It is estimated that a majority of people uses word processors and spreadsheets (54%), but only 10% on average have complex computer usage, encompassing software development, game programming, programming languages, e.g., Java, PHP, Pearl, or SQL[16]. This fact makes it very relevant to analyse the difficulties experienced by those who begin the process of learning programming. Usually, programming learning starts with features and variables instead of modelling. For modelling, students need to have the big picture of the system, and this capability requires experience and maturity. Studies demonstrate that the combination of syntax and semantics is somewhat tricky in initial phases because their mental model influences the learning process [11] [17]. Apart from those difficulties, students' cognitive domain is also crucial to the learning programming process [14]. Examples of programming learning difficulties referred in literature are [19],[20] [7][10]: data recall, identification of the problem meaning, program design, deciding which language to use, logic program translating, algorithm design and application into new situations, originality, program structure, variated skills and individual abilities, and integrating different modules [15].

### 2.2 Learning Programming - Some solutions

Studies demonstrated that visual features helped the student to learn programming because these characteristics allow students to combine theoretical topics mentally [12][13] [21][22] [23]. Apart from these constraints, one must not forget that learning process is time-consuming and laborious. Those problems are addressed by several strategies supported by various tools and methods, for instance, gamification, use of physical robots and virtual robots, visual languages and simplified languages.

However, there are often some limitations. For example, graphical languages tend to be limited in scope and of limited support. Acting like the wheels of the bicycles, in fact, end up by itself help little. Physical robots tend to add a lot of complexity, especially at the level of electronics and mechanics. Virtual robots are usually very simple, or they are limited to a relatively complex installation environment. The simplified languages suffer from the same problem of visual languages.

## 2.3 Virtual Robots

Robots seem to remain an excellent solution to support the process of learning computer programming [1] [2] [3] [4]. Specifically, the use of virtual robots has some limitation, by not having the tangibility of the real robots. Nevertheless, it has the power to focusing the study on the programming instead of electronics or mechanics' which are complex by themselves [4]. However, they should be supported in a language of general use and easy to use. JavaScript seems to be a useful and appropriate solution (e.g. [6], [1]).

## 2.4 Stages of Programming Learning

One aspect to keep in mind is the fact that there are several stages of the learning process. According to Cazola & Olivares [5], may be the following: (1) the expression language on numbers; (2) boolean type with boolean and relational operators; (3) string type with operators on strings; (4) variable declarations and assignments; (5) conditional statements; (6) loop statements; (7) code block, scope, break and continue statements;(8) functions without recursion;(9) recursion;(10) first-class functions;(11) objects and methods;(12) exception handling and (13) constructors and the prototype model.

It should also be noted that the level of knowledge is cumulative.

## 2.5 Components

One appropriate solution to consider is the existence of Components [6]. Those components may be implemented through libraries. These libraries typically aid in the development process, preventing the programmer from worrying with too much about fastidious detail. However, these libraries can also help programmers improve their skills.

Libraries should start by supporting a very reduced set of functionalities, encapsulating a high level of complexity. In practice, it is intended that students with little knowledge have interesting and tangible results.

# 3 Proposing Solution

In the following figure, we present a schematic representation of the proposed solution. This solution merges the solutions previously presented: Virtual Robot, Stage programming learning and the use of a library encapsulating several components.

The library is composed of smaller components, corresponding to higher difficulty topics and language structure.

The interface allows for the implementation in the virtual robot context. The interface is composed of two parts. In the top is the "world," corresponding to the output of the code. In the bottom, is where the student writes the code.

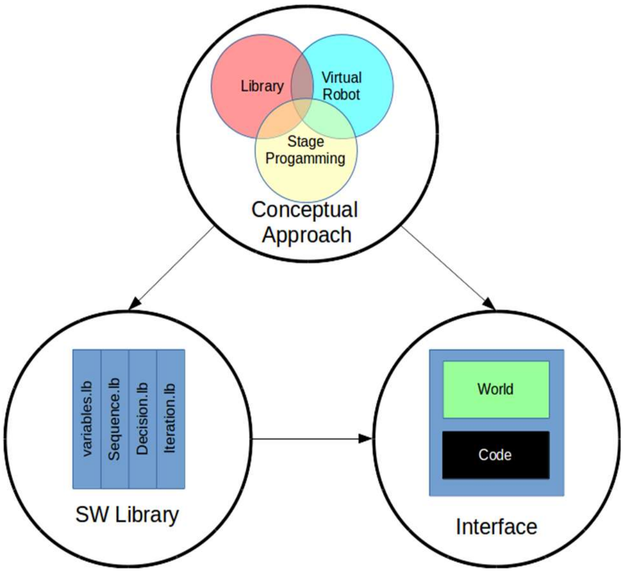

**Figure 1** Solution proposed

# 4 Implementation

The implementation of the proposed solution is done fundamentally using JavaScript, CSS, and HTML. In fact, the language used is JavaScript, as suggested by other studies (e.g. [6], [1]). The following figure shows the interface of the developed application.

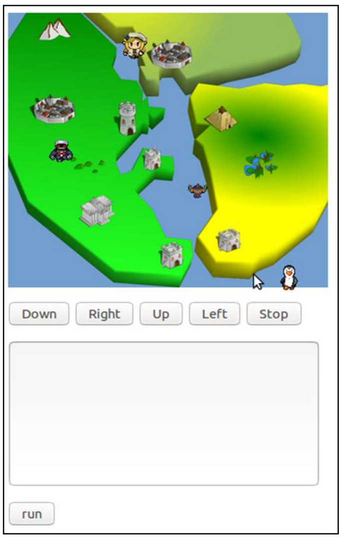

Figure 1 – Print screen of the web application.

This application was designed as web app and as mobile application. It was developed to run on Android devices, but we may easily port it to iOS as a hybrid app.

The concept of variable, attribution may be presented using a more traditional interface. In fact, it was even decided to create a small area where the user may see messages.

With this feature, the student may explore the use of variables. He may experiment the arithmetic operations but also operations with strings. On the beginning, it will be difficult for the students to manipulate the "world."

For example, in order to implement a simple sequence structure, the following code may be included in the library:

```
function code(){
program=document.getElementById("code"
).value;
var step =[];
var step = program.split(';\n');
num=step.length;
i=0;
var myVar=setInterval(move,1000);
function move(){
   console.log(i+"-"+step[i]);
   if (step[i]=="down") down(1);
   if (step[i]=="left") left(1);
   if (step[i]=="right") right(1);
   if (step[i]=="up") up(1);
   if(i>=num) clearInterval(myVar);
   i++;
}
}
```

Then, the commands "down", "left", "right" and "up" can be used. By using those simple commands, the student may change the position of the character and navigate through the "world".

The decision control structure may be implemented using an if statement. The student may use the decision control structure, allowing the character to react to specific situation. For example, it is possible, to avoid the end of the "world".

A loop is a block or sequence of statements that the system may execute several times. The following figure shows the implementation of loop control structure. In the first one, the "for" statement is used. This implementation of loop is the same as the one supplied by JavaScript and is similar to many other languages, like C, JAVA or PHP. The procedure "show" allows to print information in the screen. Another alternative is the "repeat" that works like a function. In this case, it repeats the function step as many times as presented in the first argument of the command repeat. Having the advantage of being different from the "for." It allows the student to develop an abstract way of thinking, as long as he may notice that there are several specific ways of implementing a particular concept. On the other hand, it also allows the student to be introduced to the concept of function and code modularity.

Nowadays, it is essential that the students understand the concept of autonomous behaviour. The students must know how randomness works. Making them understand the usefulness in the development of games or simulation systems.

A possible way may be to make a character move randomly. In the following example, the character zero moves randomly through the "world." In fact, the "randomMove" command generates two values, corresponding to the coordinates (X and Y) of the position of the character.

It is also possible to make several characters move simultaneously and randomly.

Naturally, all those atomic and straightforward cases may be sophisticated. In fact, we may join several situations and create more interesting problems. Challenging the students and making them gain interest.

In the context of autonomous behaviour, the robot may react to the environment. It is also possible for the student implementing several functions related to each character. For example, the character 0 is followed by character 1. The user, through the buttons, may control the character 0.

# 5 Conclusions

The solution presented here aims to contribute to the solution of the problem of learning programming. This solution is based on a virtual robot, where the novice student may access a simulation of a robot wich is less complex. But, it is also an important starting point for development in robotics. On the other hand, this solution presented a progressive approach, as long as it is supported in a set of libraries specially oriented to a specific learning stage. This solution is supported in a generic and portable language (JavaScript). In fact, instead of supporting the solution in a particular language or supporting in visual languages, it was decided to support the solution in a generic language widely used. The implementation also allows tackling the main basic programming scenarios.

# 6 References

[1] K. R. Ebelt "The effects of a robotics program on student's skills in STEM, problem-solving and teamwork" Professional paper, Montana State University, 2012.

[2] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, "A review of the applicability of robots in education," J. Technol. Educ. Learn., vol. 1, pp. 209–0015, 2013.

[3] C. Kim, D. Kim, J. Yuan, R. B. Hill, P. Doshi, and C. N. Thai, "Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching," Comput. Educ., vol. 91, pp. 14–31, Dec. 2015.

[4] A. Liu, J. Newsom, C. Schunn, and R. Shoop, "Students learn programming faster through robotic simulation," Tech Dir., vol. 72, no. 8, p. 16, 2013.

[5] W. Cazzola and D. M. Olivares, "Gradually Learning Programming Supported by a Growable

Programming Language," IEEE Trans. Emerg. Top. Comput., vol. 4, no. 3, pp. 404–415, Jul. 2016.

[6] P. Wu and P. Chang, "JavaScript for teaching accelerated programming basic courses" J. Appl. Glob. Res., vol. 4, no. 8, 2011.

[7] G. Rudolph, "Instructional technology for the teaching of novice programmers at a university of technology," Dissertation Master Information Technology, Cape Peninsula University of Technology, 2015.

[8] T. Rajala, M.-J. Laakso, E. Kaila, and T. Salakoski, "VILLE–Multilanguage tool for teaching novice programming," Computing Science Technical Report, vol. 827, 2007.

[9] W.-Y. Hwang, C.-Y. Wang, G.-J. Hwang, Y.-M. Huang, and S. Huang, "A web-based programming learning environment to support cognitive development," Interacting with Computers, vol. 20, no. 6, pp. 524–534, 2008.

[10] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," presented at the Acm Sigcse Bulletin, 2005, vol. 37, pp. 14–18.

[11] J. C. Spohrer and E. Soloway, "Simulating student programmers," Ann Arbor, vol. 1001, p. 48109, 1989.

[12] P. Chandler and J. Sweller. Cognitive load theory and the format of instruction. Cognition and Instruction, 8, 1991. 293-332,

[13] S. Garner, "The Use of a Code Restructuring Tool in the Learning of Programming," in Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, New York, NY, USA, 2004, pp. 277–277.

[14] V. Renumol, S. Jayaprakash, and D. Janakiram, "Classification of cognitive difficulties of students to learn computer programming," Indian Inst. Technol. India, 2009.

[15] S. Cummins, "Feedback 2.0: An Investigation into using sharable Feedback tags as Programming Feedback," Durham University, 2010.

[16] OECD, Measuring the Digital Economy. Paris: Organisation for Economic Co-operation and Development, 2014.

[17] S. E. Walter, K. Forssell, B. Barron, and C. Martin, "Continuing Motivation for Game Design," in CHI '07 Extended Abstracts on Human Factors in Computing Systems, New York, NY, USA, 2007, pp. 2735–2740.

[18] R. S. Rist, "Plans in programming: definition, demonstration, and development," in Empirical studies of programmers, 1986, pp. 28–47.

[19] M. Piteira and C. Costa, "Computer programming and novice programmers," New York, NY, USA, 2012, pp. 51–53.

[20] M. Piteira and C. Costa, "Learning Computer Programming: Study of Difficulties in Learning Programming," in Proceedings of the 2013 International Conference on Information Systems and Design of Communication, New York, NY, USA, 2013, pp. 75–80.

[21] C. J. Costa, M. Aparicio, and C. Cordeiro, "A Solution to Support Student Learning of Programming," in Proceedings of the Workshop on Open Source and Design of Communication, New York, NY, USA, 2012, pp. 25–29.

[22] C. J. Costa, M. Aparicio, and C. Cordeiro, "Web-Based Graphic Environment to Support Programming in the Beginning Learning Process," in Entertainment Computing - ICEC 2012, M. Herrlich, R. Malaka, and M. Masuch, Eds. Springer Berlin Heidelberg, 2012, pp. 413–416.

[23] C. J. Costa and M. Aparicio, "Evaluating Success of a Programming Learning Tool," in Proceedings of the International Conference on Information Systems and Design of Communication, New York, NY, USA, 2014, pp. 73–78.