Disguised Programming as a Teaching Aid for Students with Special Needs

Paulo C. M. Guerreiro, *paulocmguerreiro@gmail.com*
DEEI-FCT, Universidade do Algarve, Campus de Gambelas, 8000 Faro, Portugal

Vítor M. M. Vieira, *vitor.vieira@gmail.com*
DEEI-FCT, Universidade do Algarve, Campus de Gambelas, 8000 Faro, Portugal

Fernando G. Lobo, *flobo@ualg.pt*
DEEI-FCT, Universidade do Algarve, Campus de Gambelas, 8000 Faro, Portugal

Abstract

This paper describes a project that uses constructionist technologies in the development of a new approach for teaching basic programming skills with the Logo language for students with special needs.

The necessity and motivation of the learning process has been studied, discussed, and stimulated for a long time. But it has normally been directed to “normal” children, while neglecting those with special learning needs. Meanwhile, one finds a growing number of students with special educational needs. Schools are now better equipped (computer wise) than ever before, but are not taking advantage of these equipments to better teach their students.

Our aim with this research project is to motivate and teach basic programming skills for students with learning difficulties. Under this context, we compare two different approaches for reaching the proposed goal, one based on LEGO’s Robotics Invention System v2.0 (RIS) and another based on an application developed on purpose by ourselves.

Keeping in mind that we’re searching for a tool that children can use, even those with learning disabilities, we need an application that is alluring and easy to use. LEGO’s Robotics Invention System v2.0 (RIS) is an example of such an application. With RIS, one creates programs simply by dragging blocks of code into the program. Together with this simple interface, we have the MindStorms Robot itself, which provides additional motivation.

Not finding another application to suit our needs, we took upon ourselves the development of a new one. This new application was entirely written with the Logo language, using LCSI MicroWords EX™ Robotics Edition. As we propose to teach basic programming skills, something often considered dull and complex, we decided that the best approach resided in a sort of game, which is something most children find appealing. Like any good game we needed an interface that was intuitive and easy to use. But more than that, we needed it to be versatile; something that could be molded to our needs.

After the new application was up and running, time came to evaluate which of the two tools provided a better motivational and learning experience. For this, we requested the collaboration of two groups of students, each using one of the applications. Performing similar tasks, like going from point A to point B or drawing geometric figures, we can observe and analyze the motivation of each group, while they use both applications.

Keywords

Logo programming, LEGO MindStorms, Programming in Disguise

Introduction

The necessity and motivation of the learning process has been studied, discussed, and stimulated for a long time. But it has normally been directed to "normal" children, while neglecting those with special learning needs.

Along with the development of new learning motivation techniques, the way to teach has also been submitted to significant changes, either due to the creation of new technologies, like portable projectors or the Smart Board that allow us to present lectures in a more dynamic and interactive way, or due to changes in the way the teaching-learning process is made to work. At the centre of the later approach, we can find the constructionist learning environment as advocated by Seymour Papert's theories (Papert, 1993; Papert 1996) where the formal roles of teacher and student loose their meaning. The teacher abandons his place as the rigid source of knowledge and assumes the role of a guide, steering the student in the right direction. On the other hand, the student ceases to be a mere receptor of knowledge and gains an active role in the acquisition and construction of his new knowledge.

Nowadays there's a growing number of the so called *Special Educational Needs* (SEN), students. These are children that are affected by diverse factors which interfere with their cognitive capacities, such as Attention Deficit Disorder (A.D.D.), psychological disorders and even cognitive impairments. SEN students should receive special integration and evaluation conditions and be inserted into classrooms with fewer students to provide a better and more personalized teaching. However, mostly by the lack of resources (human or otherwise), these students end up in classrooms that just cannot respond to their special needs.

At the same time, schools are better equipped (computer wise) than ever before and yet these resources are not being used to help the students who struggle in their academic progression.

Taking all of these facts into consideration, we test two different applications that can be used to surpass the difficulties presented by the SEN students. We developed a Logo application called "*A Minha Viagem*" (My Trip), which is a sort of game that can be used to help the students construct their own knowledge. This application is intended to teach basic programming skills in the Logo language and the Logo language itself. At the same time we use LEGO's MindStorms and Robotics Invention System v2.0, trying to find which of the applications provides a greater motivation source for two groups of SEN students.

The main motivation behind this work is the search for good applications and/or techniques to facilitate the teaching of basic programming skills, particularly to SEN students.

In the following sections we describe the steps taken to setup the experiment, what was taken into consideration in the development of the new application, the progress of the test groups, and the conclusions we were able to infer based on this study.

Setting up the Experiment

Translating LEGO's Robotics Inventions System v2.0 (RIS)

Since we're going to work with Portuguese students, the RIS' interface language needed to be in Portuguese. This was achieved by translating the strings in key text files that contained all the words and phrases (warnings, messages, menu options, buttons captions, among others) used by RIS. The end result can be seen in Figure 1.

The main difficulty of this task is presented by the fact that one word may differ its meaning given different contexts. Nonetheless, we were able to complete the translation without compromising the original meaning of the words/phrases.

Since we constantly needed to shift between the Portuguese and English version of the RIS, we created a patch to this end.

Figure 1. LEGO's Robotics Invention System v2.0 with the Portuguese translation.

For the same reason behind RIS' translation, there was also the need to encapsulate the Logo functions and error messages needed to perform the experiment.

Encapsuling MicroWorlds EX™ Robotics Edition Logo Language

To follow the intuitive syntax of the Logo language, we felt the need to analyse the proper way to translate the English functions to their equivalent in Portuguese. However, in order to better visualize the turtle's movement, we not only translated the commands but also made some of them progressive. A good example is the "*virar esquerda*" (turn left) command, which not only encapsulates the "lt" command but also provides a glide effect while turning the turtle.

To make the error messages understandable by the students, we took the same approach as the one used with the Logo functions — all the error messages were translated into Portuguese. With this task complete, the students had available to them an intuitive Portuguese interface.

Defining our Goals

This was one of the most important stages in our case study, due to the fact that given two different approaches, the RIS and our Logo application, we needed the presence of common variables to allow us to evaluate the test subjects' response to the applications. It's obvious that by using two distinct programming concepts, we can only evaluate the test subjects' response to the applications and whether or not they can complete the given tasks. These tasks can be as simple as moving the robot and turtle from point A to point B, or, once they are familiarized with the instructions they can be given more complex tasks, such as, moving the turtle and the robot on a trajectory that represents a geometric figure. At a later stage the drawing can also be done using loop instructions. With these simple tasks alone, a child can learn the concept of function, how different parameter values affect a function's output and the need to use a way to repeat a block of instructions.

In our application, the tasks can be performed by exploring specially designed theme areas. In RIS, the tasks after being programmed are downloaded into the MindStorms Robot and can then be executed.

The test groups

To be able to assess the students' motivation, we recruited the help of five students from the 6th grade, all of which presented special education needs. The students were divided in two groups. The first, comprised of three students, working with the RIS, and the second group, with two students, using a newly developed application which is presented in the next section.

Developing the new application

We needed to create an application designed to captivate the student's attention, by creating an appealing and intuitive user interface capable of transmitting our defined goals.

After working with LEGO's Robotics Invention System v2.0, we found its interface very appealing with all its programming blocks that one can simply drag into the program. This is a very intuitive way to program and encourages the young programmer to explore.

This is a characteristic that we wanted to incorporate into the design of our application. But, at the same time, this characteristic conceals (to a degree) the concept of function and the procedural aspect of a programming language. This lack of transparency can interfere in the learning process, not allowing the child to reach the objectives we hope to achieve.

From our personal experience, we often observe that someone loses interest in learning if he has to learn something not to his liking. On the other hand we constantly observe a high level of attention when someone is playing a game that might not even be transmitting any kind of knowledge. So, why not combine the intuitive interface of the RIS with the alluring sensation of playing a game? This is exactly what we tried to achieve when developing this project.

Our application consists in two distinct parts, "*Pátio da Brincadeira*" (Playground) and "*Mundo Virtual*" (Virtual World). In the first one the students can choose between three different theme areas where they can learn the Logo commands that will be used to play the game presented in the subsequent part.

Next, we'll explain each of the theme areas, what can be done in each of them, what can the students learn and how can they learn it.

Pátio da Brincadeira (Playground)

This section has a set of theme areas with the sole propose of teaching the basic Logo commands, the same ones that we encapsulated. In order to achieve this, we designed a template screen, like the one shown in Figure 2. The screen has three distinct areas.

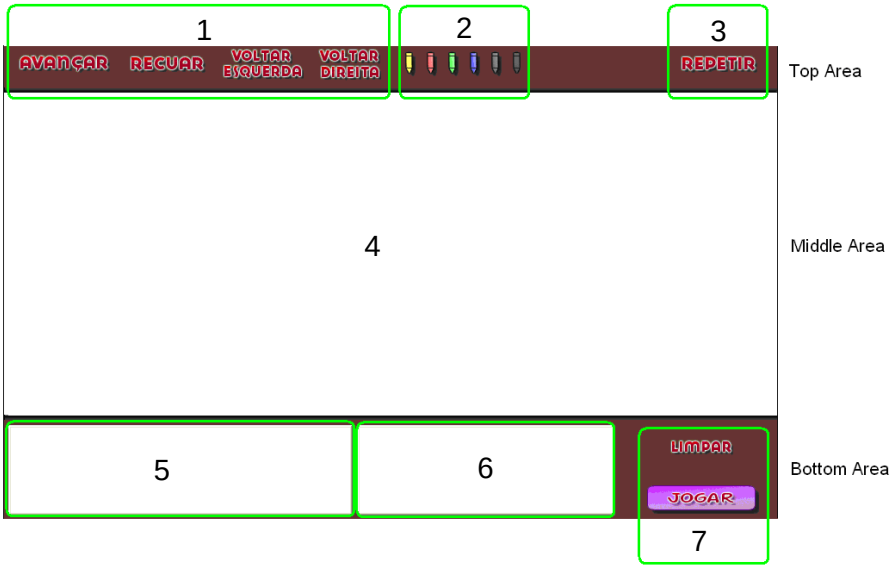

Figure 2. Template Screen for "Pátio da Brincadeira"

At the top area of the template we grouped the movement command buttons (1), the pen colour commands (2) and the repeating block command (3).

At the bottom area there is an input section (5), where the students can create the programs directly by typing at the keyboard or by pressing the buttons available at the top area. To the right of the input section there is a debugger section (6), where the currently running instruction is presented. Still in the lower area, we have available two more buttons (7), which can be used to remove the created program (*Limpar*) and to run (*Jogar*) the commands written in the input section. The run command plays a very important role as it tries to apply one of the key points presented by Seymour Papert (Papert 1996): "*an error should not be faced as a dead end, but instead as mistaken turn*". This is achieved by running the program from the beginning, command by command, allowing the child to observe the program sequence step by step.

Lastly, the middle area is where the student watches the result of the program he created. For a better usability, all the buttons shown in the user interface change their size and colour when the mouse hovers them.

This area can also be adapted by the developer to supply different themes in order to allow the student to explore a given subject. To this end, we developed three distinct themes, which can be seen in Figure 3, in order to be used in our case study. The first theme was created with the intention to provide the first contact with the application, so it doesn't offer an oriented theme. The second and third theme areas, and taking into consideration that the students have difficulties in some knowledge areas, were designed to explore geometric concepts, like angles and geometrical figures. At the same time we defined a set of exercises to explore the same concepts, like drawing geometric shapes. This way the students are not only learning how to program in the Logo language, but are also learning what is being taught to them in their classroom, and to their knowledge they are only playing a game. This way the game has two main roles, one being the provider of motivation and the other disguising the learning process itself.

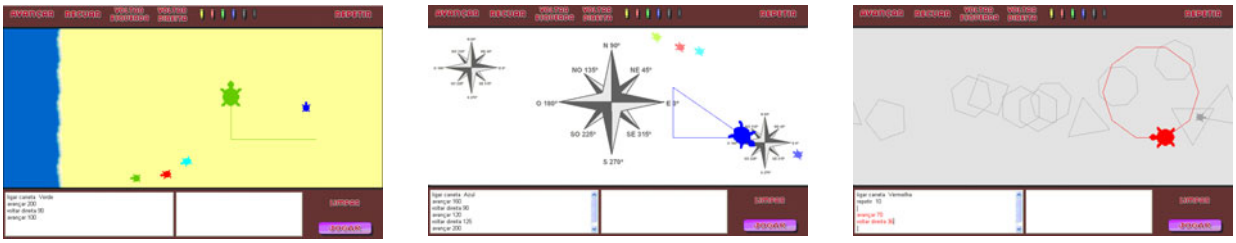

Figure 3. The three themes user in our case study.

Once the students feel comfortable using the application and the logo commands, they can pass to the next level, where they can play a set of small games to apply what they have been learning.

Mundo Virtual (Virtual World)

Mundo Virtual (Virtual World), the second part of the application, consists of two games: Praia (Beach) and Passeio (Stroll).

Praia (Beach)

This game acts as a link between the learning themes and the main game. This is done by keeping available the strictly necessary command buttons and by introducing a world, comprised of a few frames that the student can explore.

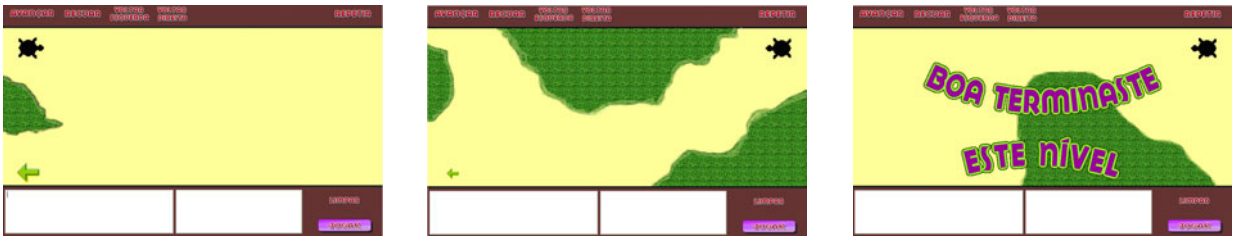

Figure 4. Samples frames from the "Praia" game

The objective of this game is to navigate the turtle around the grass and reach the green arrow that gives access to the next frame. The grass acts as a reflective barrier projecting the turtle in a random direction, this way the student cannot simply add instructions to the program in the hope that the turtle reaches its goal. In other words, the child has to give the correct instructions to the turtle so that she doesn't touch the grass. This is relevant due to the fact that, as in any written program, it always starts from its first instruction. This way, if needed, the student has to correct the code he has already written, thus providing a mechanism for him to face the error not as a dead end but as a mistake that needs to be corrected.

Passeio (Stroll)

This is the main game, where the students will be put to test. At this point the student possesses the knowledge he needs to complete the game. He knows how to play the game, what commands are available, and how to use them. But now he doesn't have access to any command buttons, so he is forced to write all the necessary code if he wishes to solve the game.

Our main goal to teach basic programming skills will be achieved if the students are able to finish the game. This is so because in order to complete the game, the students need to create a coherent sequence of instructions that will take the turtle to its final destination. Doing so, the students not only know the instructions, but have also used them in the proper way. Achieving all of this means that we managed to disguise the learning process.

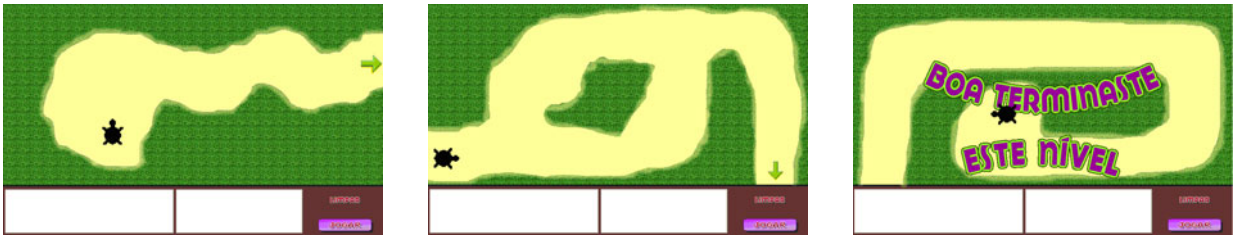

Figure 5. Sample frames from the "Passeio" game

Groups' evaluation

The two groups, the first with three students, using the RIS, and the second with two students, using our application, started the experiment by freely exploring, both RIS and our application, on their own. With RIS, the students controlled a Lego RCX robot, and in our application the students controlled a virtual turtle.

The first tasks given to the students were as simple as moving the robot and turtle from point A to point B. Once they were familiarized with the instructions they were given more complex tasks, such as moving the turtle and the robots on a trajectory that represents a geometric figure.

The experimental sessions were observed and recorded, and later analysed regarding the motivation provided by the applications to the students, and factors such as the time needed to perform a given task, the presented solution and the proper use of the available instructions.

The first good aspect is that both groups succeeded in completing the proposed tasks, and also did it without major difficulties. On a down point, both groups were reluctant to use the loop instruction even when directed to do it.

With the group using RIS, it was noticeable a greater motivation to start with, but it soon faded away. The group using our application showed less interest to begin with, but it never decreased. Just the opposite, the more they "played" the more the motivation grew. While using the application they are using Logo instructions, sequencing them to create a program, so they are actually programming. But, to their knowledge, this is nothing more than a game.

Conclusions

Both applications are able to teach basic programming skills. But our student testers with special education needs were more motivated and obtained better results when using the Logo based application than when using the Lego Mindstorms Robots. The result may be explained in part due to the fact that learning to program the Mindstorms Robot is inherently more complex than giving instructions under a Logo-based application.

Nevertheless, we truly believe in RIS' capabilities. In fact, it offers a larger array of possibilities than our application, but on the other hand, it is our understanding that the time it takes to create the program, download it into the MindStorms Robot, instructing the Robot to execute the program and actually execute it, is too long to see a response. This is where our application draws its greater strength; it takes no time to see a result from the turtle.

Like a small child that does not want to eat his spinach, but does so when his parents tells him "you have to eat it to be as strong as Popeye", we gave the students a game to provide the motivation they need to learn a subject that is usually perceived as dull and complex. Much more than that, as they see our application as a game, they managed to learn how to programme in a disguised way.

In our opinion the application that was developed and presented in this paper, is an appropriate tool for teaching the Logo programming language, particularly to SEN students. With it, we found that students could learn without realizing that they were learning.

Acknowledgments

We would like to thank the collaboration of the junior-high-school "Escola Básica 2º e 3º Ciclo Padre João Coelho Cabanita de Loulé, Portugal", their executive board and to all the teachers and students that participated in this study.

This work was sponsored by the Portuguese Science Foundation under grant POCI/CED/62497/2004.

References

Papert S. (1993), *Mindstorms: Children, computers, and powerful ideas* (2nd ed.), New York: Basic Books.

Papert S. (1996), *The connected family: Bridging the digital generation gap, Longstreet Press*, Inc.