Learning to Program: a game or a boogeyman

Adão C. CAMBRAIA
Instituto Federal Farroupilha – Campus Santo Augusto, cambraia@sa.iffarroupilha.edu.br
Santo Augusto, RS, Brasil

and

Maria Angélica F. OLIVEIRA
Instituto Federal Farroupilha – Campus Santo Augusto, mariaangelicafo@gmail.com
Santo Augusto, RS, Brasil

ABSTRACT

This paper presents the problem of the programming learning and presents the use of Scratch programming language to ease the process of learning to program, linked to the knowledge production of other disciplines, using Piaget as the background for the reflections. The Scratch language was first used in Programming I discipline in the Integrated Computer Science Course at Instituto Federal Farroupilha - Campus Santo Augusto and, as a result of the initial work, a huge involvement of the students is perceived, making learning more attractive and changing that “boogey” look of the program.

KeyWords: Programming Learning, Scratch, Integration, professionalizing education, logical reasoning.

1. INTRODUCTION

The teaching and learning process of programming is usually seen by many students of the computer sciences as the “boogeyman”, because it is a subject that requires logical reasoning, critical thinking and concentration, which makes the learning stressful, difficult and frustrating to some students. At Instituto Federal Farroupilha – Campus Santo Augusto, the Integrated Computer Science Course is that one with technical subjects of computer science and also the common issues. According to the Institutional Development Plain (PDI) of the Institute, the articulation between the knowledge areas of basic and professionalizing education is necessary to develop the integrated curriculum; and these, on the other hand, with the research and extension area. “The integration demands the relation between the general and specific knowledge to be continually built throughout the formation, under the work, science and cultural axis”[9]. The integration of the knowledge areas is not that easy referring to the common issues, for example: Portuguese, Mathematic, and Physics. The perfect integration of them with the computer science area can be presumed. In the technical aspects, the teaching-learning program presents difficulties: a) for the students – to understand the new language, to assimilate problems; to realize the connection between the abstract and the concrete (we usually use Mathematics to introduce to the programming); b) for the teachers: the languages are commonly used in connection with Mathematics, which makes it difficult to relate with the concrete aspects. We then get into a vicious circle: on one hand, the teacher unable to work, even though he/she increases and diversifies the ways to teach; on the other hand, the student unable to learn. These aspects are responsible for the high failure and evasion rates of the Computer Science Course, especially the subsequent courses.

In the hunt after a solution to this complex situation, a research with the second year students of Programming I subject in the Integrated Computer Science Course was made. With this research, we intend to understand and make practical a change to the teaching of programming into something more interesting for the student and, thus, to make the understanding more efficient. This way, the paper proposes the use of Scratch programming language to reach the goal, facilitating the concepts organization and allowing the students to check their results as the programming grows. We then start to discuss some important aspects in our research. This paper is organized as follows. In the second section we discuss the teaching of computer programming. In the third section the Scratch language is presented. In the fourth one, the proposal of using Scratch in the Integrated Computer Science Course and its first results are presented. Finally, in the fifth section some considerations about this paper and future claims are presented.

2. PROGRAMMING TEACHING

The programming teaching is focused on giving the students the opportunity and the knowledge to develop programs and solutions to the everyday problems. To [8], learning to program is a difficult process to many

1 Subsequents are technical courses for those who have done or are doing the High School. Their curriculum consists only subjects of the technical area.
students, which is noticed in the subjects that introduce to the programming teaching, such as Algorithms, when we realize that many of them can not understand and learn simple abstract concepts. Facing the problems, they cannot find a solution; but, when the teacher proposes a solution and gives them explanations, they pretend they understand. However, the same difficult appears again in the next exercise. To [2], this inefficiency in the process creates a lack of motivation that can result in the students giving up. However, the Instituto Federal Farroupilha reality creates another situation, because the Computer Science Course is integrated with the Secondary School, and for this reason the students do not evade, but only continue the course because of the secondary school. In this way, we observe that this lack of interest in the programming is also reflected in other technical subjects, damaging the students’ performance in the course.

The programming subject, by itself, is considered the “boogeyman” by most students of Computer Science. Authors such as Rocha (1991), [3] emphasize the teaching-learning failure which have been recurrent in many institutions. Facing the truth, a certain number of efforts have been developed in order to reduce this problem, which reinforces the weaknesses mentioned, presenting ideas to revert this scenario [8]; [2]. However, there are only a few positive and effective results in solving these problems.

Even with the difficulty of learning a programming language, Seymour Papert, in his book “Logo: computers and education”, had idealized programming for children. The author writes that, when he was a child, he used to play with gears in his grandparents’ garage, as he could easily assemble and disassemble differentials. Once at school, Mathematics was easier for him than it was for his mates. Later, when he studied Piaget and how children learn, he could understand why he had more facility at school. He understood the when he received a mathematical task, for example, he associated it with the gears and was then able to comprehend the meaning of the subject, solving it easily. According to the author, children can learn just by the use of this gear system. However, this system had an affective meaning to Papert but could mean nothing to any other teenager.

When he wrote about Logo, a primitive version of Scratch, Papert (1986) said that the motivation is not the creation of a final product, but a process: the mistake found can be used as a challenge to reach the goal. This way, people with different skills and interests can discuss with each other about the mistake. The idea is to find the reason of the mistake, and not put it out and forget it. Depending on how the computer is used, people get happy when they can delete mistakes without leaving the trace. In the case of Scratch, the opposite attitude is suggested. The mistakes are seen as members of the learning process, because they help us to understand what just happened. Teaching the computers to “think”, children start a journey to discover how they think. Thinking about the way they think turns children into epistemologists, an experience that few adults had.

Scratch is a programming language, created at MIT (Massachusetts Institute of Technology), inspired by LOGO and Squeak (Etoys), which intend to be easier and more intuitive. Because it does not require a previous knowledge in other programming languages, it is ideal for those who are beginners in programming. It uses a graphical interface that allows structuring programs as assembling blocks. Each language block contains a separate command, which can be put together if they fit. And the commands can be changed through the menus. It was developed to help in the computing concepts learning and also in other knowledge areas. With Scratch it is possible to create cartoons, games and other interactive programs.

There is a variety of initiatives in activities that show the use of Scratch programs in different areas. Some example projects can be found at www.scratch.mit.edu or http://kids.sapo.pt/scratch/. In these sites it is possible to provide the programs made by students and find many others already made or under developed. The idea is to reproduce the programs and improve them. The reproduction is stimulated because it improves the understanding about computing and the subject. Therein, the student learns how to run a particular function, for example, and how to make a character move on the screen and identify parts of a cell. The students begin to cooperate to achieve their goals. Even a program that is considered to be ready can be improved by another person. In this context, learning communities can be set, so everybody can share information and, at the same time, challenge themselves to do their best.

In most cases the game is used as an understanding tool to learn the program, which leads to the construction of the computing knowledge. During the time, the students have been using this knowledge to develop projects in different areas. To illustrate this issue [1] reports the Scratch use in a Secondary School, as he describes how the Science teacher developed a work that enhanced the production of knowledge in her area, using Scratch. One of the projects can be found at http://scratch.mit.edu/projects/telle/54060.

In this work, we propose the development of activities in the programming area for the students to know the software. At this point, it is possible to create games, simulations and cartoon, improving the knowledge about computing and, at the same time, the students are proposed to develop concepts in others areas, usually in extra time. The student chooses the area and the concepts he wants to work with and establishes a dialogue with the teacher to clarify eventual doubts.

4. LEARNING THROUGH THE SCRATCH INTERFACE

The concern is now to understand the interface as a learning possibility Mathematics and Physics. [4] says that this interface is “a contact surface, of translation, of articulation between two spaces, two species, two different reality orders: from a code to another one, from analogical to digital, from mechanic to human...
Everything that is translation, transformation, transition, it is from the interface order”.

Many Physics concepts involve interface systems; the materials and equipment provide the interfaces with the physical world. This research proposes the use of Scratch as a program to study physics, where programming concepts can also be explored in an integrated teaching process. Though this proposal is complex, it can promote an integrated study between different knowledge areas in the classroom. Facing the necessity of controlling, learning to program and to be good using Scratch, it is interesting that this process gets a meaning in the student’s concrete life; for example, by simulating the construction of electric circuit.

The human action and interpretation contribute to new ways to represent the physical world. Concepts and symbols allow us to understand the physic world. So, a concept such as electric current is only a representation, which demands an equipment to enable a quantitative analysis to be understood.

It is necessary to comprehend the development to understand the learning process. To [5], development explains learning, because it is recognized as a process in connection with the totality of knowledge structures. On the other hand, learning is caused by situations and a psychological experimentation. Therefore, it is essential that the developing person observes the operation development, because knowing demands the change, the transformation and the comprehension of this transformation process. [5] believes that the knowledge begins when the child assimilates something from the physic or social environment. This assimilated content causes a number of perturbations to the person because it carries something new to the assimilating structure, something unknown up that moment. This rebuilding is arrangement. It is this movement, this action of getting the lost balance back that creates a new person, because new assimilations occur. [5] names as “the main equilibration” the mechanism of evolution or development of the organism. This is the knowledge working up.

To [5] the development is related to the following factors: maturation, experience, physical environment effects (social transmission) and the equilibration. Piaget believes that children are not miniature adults. Children have different physical and cognitive structures that change during their development and pass by various stages. A set of cognitive structures correspond to each one, a schema which consists in a pattern of behavior or thought that emerges from the mixing of simpler and primitive integrated units in a more organized and complex scenario. The experience, although widely used by empiricists, does not have the same meaning in the Piaget work. The experience is associated with the constructed knowledge, not with the object but with the relationship established between the person and the object. The social transmission is made through education, the social and observation of the others. The equilibration is an internal mechanism that permits the balance between adaptive processes of assimilation (that sets the new information in existing cognitive structures and systems) and the arrangement (that modifies or changes the existing structures and systems). This way, the equilibration is understood as the mechanism to keep the dynamic balance of the person, a self-regulation. It is a game of adjustments and compensations to reach people progressively more complex coherent structures.

Piaget considers four periods in the evolutionary process of the human specimen, which is recognized by what the individual can best perform in age group. These are stages of intellectual development for children and teenagers, which are: sensorimotor (up to 2 years old), preoperative (2 to 7 years old), concrete-operative (7 to 11 or 12 years old); formal-operative (12 years old onwards).

The intelligence in [6] is the organism’s adaptation ability to new situations, the construction of new structures. It is related to the external world and the biological adaptation. To the neuroscience, the intelligence is the object of researches that investigate how brain can learn and how the memory works in the molecular and cellular level in the cortical areas. In the education process, researches intend to understand factors such as: learning, memory and emotions; neural plasticity, activities that stimulate cerebral cortex areas; the brain answers to primitive heritage, like pictures, images and symbols.

Associated studies with learning and the human brain are highlighted in the Gestalt Psychology Theory. Through behaviorist psychology, the emotions, perceptions, perform important role in the learning process. In this theory the brain is a dynamic system in which an interaction occurs between the elements that participate in the perception and stimulus received by the senses in favor of learning. Skinner tried to incentivate the study of human nature, behaviors and controls, operant conditioning, listing concepts and providing models to teaching/learning. However, it is in Piaget genetic psychology that we find the explanations about the learning genesis.

To Piaget, intelligence is an individual capacity, where the human being is a homo faber of himself/herself, who is building his/her intelligence while reality representation is established. For him, intelligence is an adaptation way. While structuring his theory, he organized observations, experimentation and conclusions about the development of human intelligence, putting it as an individual surviving ability, starting in the newborn reflexes. The educator has to establish a dialogue between theoretical and practical knowledge and promote an explanatory speech, of the symbolic expressions and representations involved in the process of human development and learning.

5. SCRATCH PROPOSITION IN THE INTEGRATED COMPUTER SCIENCE COURSE: FIRST RESULTS.

From the studies run about the Scratch language and its positive effects that can be reached with the programming, for example, the frustration which comes from the failure in advancing in program implementation
that is replaced by the motivation of building the first program. The results can be seen even though they are very simple. On a picture, the command “effects” is applied, followed by some interact instructions with the mouse; effects are obtained as the user moves the mouse.

This facility to program is also a result of Scratch interface organization that is divided in the following parts: category commands; scripts area, stage; sprite. The following commands can be found in the area: movement, appearance, sound, pen, control, sensors, numbers and variables. Inside, each item has different commands that can be dragged to the script area forming the program. The stage is where the programming results can be seen and the sprite is an object found in the middle of the stage. If the goal is to move the sprite on the screen, which category of commands should be used? Just click on the movement category that a series of commands will appear to move the sprite, just choose the one that best fits with the problem; it is a completely intuitive language.

Besides, the language provides a series of resources that allow changes in the stage and sprite. A list of sprite and background options is available to the programmer to add on the stage, it is possible to only change the sprite or to insert new ones. With this device, students can create a diversity of activities: dialogues, stories involving many sprites (characters) and at the same time they change the stage (background). Multimedia animations.

With a little more knowledge it was possible the development of a game proposed by Paddle2see, a highly regarded “scratcher” in Hampden, Maine, USA, the author of this idea; “There is a flock of sheep with fractions written on the back. There are three or four fences and, preferably, a sheepdog. The student must command the dog to push the sheep against the fence, separating them in equivalent groups. If the sheep gets to the fence it does not belong to, the other ones inside the fence run away. In the end, the points got by the student are added”. It is a complex game which involves lots of programming concepts that students implement by cooperating with each other. This article proposes the use of Scratch, which enables the learning of a programming language and also allows the interaction between the subjects that make up the curriculum of Integrated Computer Course.

In figure 01, the relation between the three areas (defined by the National Curriculum Parameters) and programming is showed. On one side the areas can be visualized; Languages, codes and technologies; Human Science and its technologies and Nature Science and its technologies. The students are proposed to develop some concepts of one area; they have to have an idea which they want to implement. With this idea, they start thinking about the instructions, a series of tests area directly run in Scratch – as this is an intuitive language, it helps them to choose an appropriate command for the situation.

As the programming goes on, the mistakes can be found and corrected by returning to the program to review some command that was mistakenly used; it can then be replaced by another one that is more appropriate. This relationship with Scratch eases the identification of mistakes in understanding some concepts, which provides a return to the knowledge areas to enable explanations. With this feedback the understanding in the area gets better and the student learns to program.

In these first Scratch studies, we noticed that while the program is under development, the students have the possibility to see the results, and it enhances the confidence and the sensation of creating a product. It may look simple, but these impressions help the students to like programming better.

6. CONCLUSION

No doubt the programming subject is one of the most terrifying in the computer science course and for this reason it may explain the high failure and evasion rates in this course. This way, the dissatisfaction with this reality has forced the teachers of the Computer Science Course into a research for alternatives to change the scene. Scratch appears in this context.

Scratch is a programming language considered to be more accessible because it does not require previous knowledge of any other programming language, which makes it ideal for teaching and for those who are beginners in programming. Its interface allows building programs using building blocks, which represent a programming command and makes it possible to develop and work the language concept in an interactive and animated way, as if it were a game; this is its main attraction.

Up to the moment, Scratch has been used as an experiment in the Programming I subject, i.e., exclusively for programming teaching without concerns about the other teachers using it in their classes. The first impressions of using Scratch were positive, which has created more interest from the students in the subject. The involvement and the wish of learning were visible, different in comparison with the frustration expressions, before the use of the new language.
To be effective, this research needs to expand its use to the teachers of other areas, which permeates an information work and comprehension of Scratch efficiency also in producing knowledge in other areas. This work was developed in a second year class of the Integrated Computer Science Course, but we intend to use it also in the first year, in algorithms subject and that the teachers of the common issues adopt it as an additional device to the educational computing classes. So, in the second year the regular subjects of the common issues can continue the use of Scratch combined to the Programming I and in the other areas of language programming. It is interesting to emphasize that in the Computer Science Course we plan to develop some extension work with Scratch in elementary schools around the region, with the purpose to introduce the students to the program, so they can join the integrated courses with a brief knowledge of scientific and technological production.

Using Scratch in the dynamics of knowledge construction means enhancing the equilibration process (Piaget, 1976). From the moment that the student chooses the subject to create the algorithm up to encoding the program, he passes through lots of unsettlement process. When he chooses a subject, an assimilation process occurs (he understands the concept, according to his perception) and he starts programming and better understands the concept; then he notices that it is not exactly how he thought before; it takes him back to the books until the settlement process occurs.

Obviously, this is not only about incorporating a technology to the classroom and imagining that the construction knowledge process becomes more efficient. It is entering a new culture in the educational space. A culture in which the teacher is not the only one who knows things and the student puts himself as the subject of building his knowledge. With Scratch the child programs the computer and not the computer programs the child.

7. REFERENCES