Introducing CS-major Students to parallel programming using StarLogo

T. FOLIAS, V. DAGDILELIS, M. SATRATZEMI, G. EVANGELIDIS
Department of Applied Informatics
University of Macedonia
156 Egnatia Street, GR-540 06 Thessaloniki
GREECE

Abstract: - One of the most common difficulties an instructor faces during an introductory course in computer programming, is the lack of appropriate pedagogically efficient "tools" that will help his/her students understand fundamental notions of programming. Especially, complicated notions are, most of the times, only partially taught or left uncovered for a more advanced course. However, the use of modern technology and computers in education, programming included, is not confined only in computer science courses, but finds applications also in sciences like Math, Biology and Sociology. As a result, there has been also a demand for simpler and friendlier software environments that will provide the vehicle of knowledge dissemination for these sciences and, more generally, for education. Starlogo - introduced by Mitchell Resnick at MIT Media Laboratory – is an programming environment that simulates decentralized systems and provides an intuitive interface that can be used even by elementary school students to explore systems and worlds in which thousands of objects participate and interact with each other. Of course this same environment can be used for an introduction to programming and especially to massively parallel programming. In this paper we initially present a number of different approaches that have been used so far by instructors to improve the didactic experience of their students with the power of the Starlogo environment. Using the Starlogo environment, we also organized a series of experimental Starlogo courses. In this paper we present the way these courses were organized and some preliminary results of our experimental approach.

Key-Words: - educational software, Starlogo, parallel programming, modeling

1 Introduction

The rapid growth of technology the past recent years has provided both teachers and students with a great variety of new educational tools that enhance and ease the learning process. New software environments, educational games and interactive multimedia applications have been developed in order to satisfy the demand for simpler and friendlier virtual learning environments that will provide the vehicle of knowledge dissemination for Math, Biology, Computer programming, Sociology and, more generally, for education. However, complicated notions are, most of the times, only partially touched or even left uncovered by these applications. Most of the times, the obstacles a teacher has to overcome when trying to address such notions are implementation complexity and the steep learning curve required by the software environment being used. Thus, complicated and time consuming – in terms of teaching - notions are left behind. From the students' perspective, having only 'hammers' in their educational toolset, they only see nails! [1]

Starlogo[2] comes to fill this gap by providing a simple and easy to understand interface, yet powerful enough to teach complex natural phenomena like the way an ant colony gathers food, a free economy distributes its wealth or traffic jams are formed. Starlogo was introduced by Mitchell Resnick at MIT Media Laboratory back in 1994 and after several versions is currently an open source project written in Java and consequently available to all computer platforms. The environment simulates decentralized systems and provides an intuitive interface that can be used even by elementary school students to explore systems and worlds in which thousands of objects participate and interact with each other. [3] [4]

The rest of the paper is organized as follows: In section 2 we outline previous examples of the use of Starlogo in different educational sets, in section 3 we describe an experimental course in parallel programming with the use of Starlogo that was carried out at the University of Macedonia and in section 4 we present the results of this pilot course. Finally, in section 5 we conclude our research on the evaluation of Starlogo as an educational tool based

on the feedback we acquired during our experimental course.

2 Related Work

In MIT [3] Starlogo was used by two high school students and the help of a teacher to understand why and how traffic jams are formed. The students, after discussing the possible causes of traffic jams, applied a set of simple rules in the form of Starlogo commands and simulated a highway. They expected that the real cause of traffic jams are obstacles like traffic lights or a police car patrolling. However, they found out this was not the case. After correcting wrong initial assumptions they discovered that traffic jams are formed because of the different speeds of each car and that traffic jams move always in the opposite direction of the highway lane. This last observation triggered the teacher to explain how waves are formed.

In [5] P. Aderson and C. Seaquist describe the learning difficulties that freshmen students face during introductory math courses and particularly when learning to solve differential equations. Most of the students do not understand whether they have to memorize specific techniques for a limited number of differential equations of closed form, focus on theory or spend more time on equation modeling. To anticipate this confusion, Aderson and Seaquist used Starlogo as an alternative. Using 'population dynamics' in small Starlogo programs, they successfully presented to their students notions like exponential growth, logarithmic growth and Volterra-Lotka predator/prey systems. The students could actually interact with the systems and change the initial population, birth rates and as a consequence get different results.

In Duke University, Florida, Starlogo was used along with four other educational environments to teach programming to non-computer science students [6]. After a few weeks the students were able to implement a termite colony and investigate how they gather food and if there is some kind of central control in this process. After evaluating all five environments, Starlogo gathered 75% of students' preference.

In Florida Institute of Technology, Andy Tinkham and Ronaldo Menezes [7] developed a simulation environment for robotics using Starlogo. Starlogo was used in order to visualize notions and models of robot behavior that were impossible to implement otherwise due to the large number of robots required.

3 Our Experiment

Research data so far has shown that a "successful" programming course has to take into consideration many important factors that relate to the organization of the course. One of these factors is, for example, the programming style – parallel, functional, imperative or object-oriented programming. Another important factor is the "programming environment" that is being used: the programming *language* and the programming *interface*. Additionally, the type of programming exercises and assignments in terms of complexity and length of required solution are two factors that may help the students or affect them negatively towards the course.

Taking into consideration the complexity of an introductory course with the desired characteristics (mentioned in previous paragraphs) we set up our course based on previous parallel attempts by other researchers, adjusting the course to our audience and our didactic objectives.

During our course we performed a systematic record of the solving process followed by the students: the student's actions and programming on each workstation was recorded in a video file with corresponding screen capturing software. In addition, we handed out questionnaires to the students in order to evaluate the environment used – e.g. interface friendliness, language characteristics etc.

Our experimental course with Starlogo took place at the department of Applied Informatics of University of Macedonia, at Thessalonica, Greece during the spring semester of 2006. The focus of our course was to introduce parallel programming to undergraduate computer science students with no prior knowledge on the specific field. The course was organized into five two-hour weekly lab sessions. During the first hour of each session, a lecture was given with new notions on parallel programming presented on a projector along with examples in the Starlogo interface. Also the necessary Starlogo commands were explained and questions were posed to students to verify that the content was understood. In the second half of the session, students were asked to work on small assignments in groups of two. Having 18 students participating in the experimental course, 9 groups were formed. Each group worked on a single workstation. The Starlogo version used was 2.22 and for the reason of the particular experiment the Starlogo interface was translated into the Greek language.

During the first session the students were introduced to the decentralized way of thinking by examining how simple Starlogo commands like FD (forward) can produce massive behaviour changes on a large group of Starlogo turtles that execute the same command concurrently. By the end of the first

session, students were able to interact with the Starlogo interface, identify the various roles of the environment – turtles, patches, and observer – and execute basic Starlogo commands. The students were asked what would happen if having a large number of turtles on the same patch with random orientation the command FD 10 was executed (i.e. forward 10 steps). The majority answered that the turtles would scatter randomly on the patches and where surprised to find out that in fact the turtles formed a circle with their bodies with the initial patch being the centre of this circle. In the rest of the session students were presented existing models in Starlogo and were asked to guess the behaviour of the turtles before actually running the model. This helped them understand that no central control was used in the models and that all turtles executed simple commands that caused specific patterns in the turtle population as a whole.

On the second session the students learned how Starlogo turtles could communicate with the patches on which they moved and how they could exchange information. The notion of random walk was also introduced. After explaining the corresponding commands an assignment was handed out to each group. The aim was to simulate a simple virus spread model, where a contaminated turtle infected every turtle found on the same patch during a random walk. The assignment was split in 9 different cases of incremental difficulty and by the end of the second session most of the groups completed almost half of the cases. The rest was given as homework for next week.

During the third session, students were introduced to Starlogo procedures and learned to use the graphic interface in order to trigger specific procedures and watch variables in Starlogo monitor controls. They were also introduced to the notion of breeds and how they could use it in differentiating the behavior of different turtle groups. To elaborate on this notion, the second homework they were given included the simulation of the 'Turtle-Hoare Race'. [8]

During the fourth session, the students learned how turtles could avoid collisions and detect walls on their environment. The goal of the session was to model the behaviour of a large population in a closed environment with a single exit in cases of a panic (fire, earthquake etc.) This was the objective of the third and final assignment that was given to the students.

The last session was dedicated to a written exam with multiple-choice type of questions.

4 Results

In this paragraph we focus on the most important results of our experiment, since the presentation of our findings in full extent would require much more space.

After recording and evaluating the students' feedback both in the lab and their homework we identified the following interesting results.

Students had a rather positive impression regarding the programming environment they worked on. Their evaluation, presented on a Lickert scale 1-5, provided a general average value of 3.36 . They liked the fact that the programming environment is not complicated, provides continuous feedback to the user and that in general, supports adequately the students during the process of solving a (programming) exercise. However, they underlined the fact that the error messaging system needs a serious improvement (average value 2.44).

The Starlogo programming language was evaluated positively as it provides a "compact" language, it does not require extended coding to solve a problem and the commands have simple form, syntax and semantics.

The programming assignments that the students worked on during the course were evaluated positively -because they were different from what the students had seen in other programming courses, triggered students' interest and in spite of their phenomenal difficulty, were implemented easily with Starlogo. The courses as a whole were also evaluated as satisfactory.

The Starlogo environment in analogy with the traditional Logo language, visualizes the results of a program with a vivid interaction: the "behavior" of the turtles. Our experimental courses allowed us to verify the fundamental pedagogical hypothesis behind Starlogo: the "turtle world" plays a significant role in understanding the functionality of a program. It helps students to identify logical errors and increases their overall progress.

In general, the Starlogo environment and language, as well as the courses the students attended were evaluated positively. However, after analyzing the solutions they provided on specific problems and their answers on a set of questions that were posed to them at the end of the pilot course, we identified certain difficulties. We organized these difficulties into two distinct groups:

1) Implementation difficulties of mathematical conditions in procedural form: students had, for instance, difficulties in understanding how to implement statements including probabilities like: "If a turtle meets an infected one, then with probability

70% it gets infected as well". Although they new how to produce a random number on a given interval, none of them managed to find the correct solution. This is a well known source of difficulties for novice programmers.

2) Difficulties of misunderstanding regarding the function of specific Starlogo commands. For instance, an important difficulty they faced was when introduced to the *loop[ ]* command. This command actually repeats forever the code inside the brackets. Most of the students were adding Starlogo commands after and outside the loop command brackets expecting that it would execute, although this was not possible. As far as the assignments are concerned the "Virus spread" model was implemented by 100% of the students, the "Turtle-Hoare Race" by 80% and the "Panic Room" model was completed only by 20% of the students.

In particular we have to stress the fact that students find it hard to predict the outcome of a set of Starlogo commands, as one can easily find out from the questions we posed to the students. This fact is responsible for the trial-error strategy they tended to use during the solution of a programming problem, a strategy ineffective when complicated problems are concerned.

These results lead us to the conclusion that students may lack the knowledge and understanding of fundamental programming notions. For example, they do not seem to understand completely the fact that the commands of a program – regardless the programming environment – are executed in a deterministic way. The behavior of students appears to strengthen our opinion that when facing new programming notions (like the forever loops) they tend to interpret the result of a command based on their previous experience and not according with the formal semantics of the code that is executed.

The difficulties they face during the solution of a trivial programming exercise which they haven't seen before, as well as their difficulty in predicting the results of a program, show that they haven't deeply understand basic programming notions. As a result they are not able to apply their knowledge on new problems or even model new problems with an algorithm.

5 Conclusion

In order to introduce students to parallel programming with a simple an easy to approach method while at the same time evaluating the use of modern modeling environments in various scientific fields, we organized a short introduction to parallel programming with the Starlogo programming interface and language.

Starlogo, a member of the Logo-like family of languages, is a massively parallel programming language. Although it has been used by many students while investigating various scientific phenomena, we do not have enough feedback for the nature of difficulties these students face in the specific environment.

The first results from the analysis of our introductory courses confirm the research findings presented so far and our initial assumptions: the Starlogo interface is friendly to the user and Starlogo commands are easy to use.

Also it is obvious that the visual presentation of a program's execution helps students significantly in understanding its logical structure and functionality. What is more, this visual aid provided by the environment of Starlogo is the determinant factor of their quick progress even when coming across complicated programming models like the parallel programming.

However the correct use of Starlogo commands, especially the totally new ones, is not easily achieved by the students. The latter, tend to use these commands in an inappropriate way and often they are not able to predict the result before their execution. Our hypothesis is that the students have not mastered the programming level required in order to face new programming notions easily, especially when these notions belong to different programming paradigms.

Our preliminary results have shown that both the courses we organized and the problems we selected for the students were satisfactory. The main findings of our pilot course allows us to organize our next courses in a more efficient way, taking into account our these results, since we can focus on the source of the most common difficulties for the students and allocate much more time to make the learning curve of Starlogo easier.

References:

[1] Resnick M., StarLogo: An Environment for Decentralized Modelling and Decentralized Thinking. *CHI Conference Companion*, 1996, pp. 11-12

[2] StarLogo, http://education.mit.edu/starlogo/

[3] Resnick, M., *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press, 1994

[4] Resnick, M., Bruckman, A., and Martin, F., Pianos Not Stereos: Creating Computational Construction Kits, *Interactions*, Vol. 3, No. 6, 1996

[5] Anderson P., Seaquist C.R., Using StarLogo to Introduce Differential Equations, *12th-Annual International Conference on Technology in Collegiate Mathematics*, 1998

[6] Rodger S.H., Using Animation, Virtual Worlds, Pair Programming and Activities to Introduce Computer Science, *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, Duke University, 2001

[7] Tinkham A., Menezes R., Simulating robot collective behavior using StarLogo, *ACM Proceedings of the 42nd annual Southeast regional conference*, Huntsville, Alabama, 2004

[8] Colella V., Klopfer E., and Resnick M., *Adventures in Modeling*, Teachers College Press, 2001