

Engaging Students of Senior High School in Simulation Development

Katerina GLEZOU, Maria GRIGORIADOU

*Educational and Language Technology Laboratory, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens
Panepistimiopolis, Ilissia, 15784 Athens, Greece
e-mail: {kglezou, gregor}@di.uoa.gr*

Received: September 2009

Abstract. In this paper we present a small-scale study investigating the use of the MicroWorlds Pro multimedia programming environment as an authoring tool for constructing models, simulations and multimedia applications with students of Senior High School. We implemented the cross-thematic educational scenario “Free fall simulation development” as an open and flexible framework for activities in actual classroom circumstances, exploring two alternate instructional strategies: a) simulation development from scratch and b) use of a preconstructed microworld, and we observed how the students collaborate and interact with the programming environment. The findings highlight the overall process and the differences in the students’ levels of engagement and performance, indicating some special features of the programming environment that contribute to or cause difficulty in the creation of an effective learning environment. We attempt to contribute to the discussion on the main parameters of designing, developing and implementing an effective constructionist approach aimed at engaging students in simulation development as a cross-thematic multimedia project.

Keywords: Logo programming, microworlds, MicroWorlds Pro, simulations.

1. Introduction

The introduction and exploitation of Information and Communication Technologies (ICTs) in the classroom remains an open, composite and multifactoral issue.

The study we report here is part of our effort to extend our experience in designing learning environments that support learning through exploration, expression, construction, meaning negotiation and collaboration. We attempt to gain some further insight into the potential of using the multimedia programming environment MicroWorlds Pro as an authoring tool for constructing simulations and multimedia applications in the context of a cross-thematic educational scenario that promotes collaborative exploratory learning.

This paper presents a pilot implementation of the cross-thematic scenario “Free fall simulation development” as an open and flexible framework for activities (Glezou and Grigoriadou, 2008, 2009a). This scenario combines elements from Informatics, Physics and Mathematics, and places emphasis on building microworlds, simulations and multimedia applications as projects. The series of activities developed is characterized by

a gradually increasing degree of complexity and difficulty and is supposed to function as scaffolding during the gradual familiarization of the students with Logo programming language, by applying in action the constructionist reasoning and exploiting the gradually acquired experience providing students with tools that they are in a position to use.

We implemented two different instructional strategies: a) simulation development from scratch and b) use of a preconstructed microworld in actual classroom circumstances. We investigated how students of the 3rd grade of Senior High School interact with the programming environment. The findings highlight the overall process and the differences in the students' levels of engagement and performance, indicating some special features of the programming environment, which may foster or hamper students' engagement and learning.

The aim of this paper is to discuss the basic parameters of an effective constructionist instructional approach aimed at engaging students in a cross-thematic multimedia project, in the level of designing, development and implementation, as well as to provide the educational community with particular examples of practices for inspiration, implementation and reflection.

The remainder of this article is organized as follows. In Section 2 the literature review and the theoretical framework of the study are described. In Section 3 the research framework of the study and the educational scenario "Free fall simulation development" followed by the lesson plan which was implemented in accordance with the second teaching strategy are presented. In Section 4 evaluation findings are reported. In Section 5 conclusions are given. Finally, in Section 6 special issues of interest, limitations of the study and future research plans are discussed.

2. Literature Review – Theoretical Framework

A number of educational research projects have used computer-modeling tools in education and, mostly, in science learning (Schwartz, 2007; Wilensky and Reisman, 2006; Dimitracopoulou and Komis, 2005; Teodoro, 2002; De Jong and van Joolingen, 1998). Recent research focuses on further understanding design characteristics of computer-based programming environments which may promote or impede learning with models in science (Simpson *et al.*, 2005; Louca *et al.*, 2003; Louca and Konstantinou, 2002). Learning to formulate, analyze, test and revise models is a crucial aspect of understanding science, and critical to helping students become active, lifelong learners. In the teaching process, two different ways of exploiting simulations are used: model-using and model-building. Model-using is the process where we use a simulation created by someone else, while model-building is the process where the user builds the simulation directly. Typical uses of a simulation consist of "what-if" actions, changing parameters and, or, initial conditions and observing how these changes affect the system.

The true value of modeling emerges when students can elaborate concepts developed in precedent modeling activities, in order to deal with more complex modeling problems (Forbus *et al.*, 2001). The valid exploitation of modeling educational software cannot be

achieved through exploring the readily made models that come with the software, but through giving students the opportunity to express their ideas, examine the validity and the power limits of their ideas, and reach a point where they can gradually build scientific knowledge (Komis *et al.*, 1999).

It is important to study and identify the particular ways that students use computer-based environments that support student inquiry and modeling practices. It is possible that there are features of different programming environments that support learning and others that hamper learning (Louka and Konstantinou, 2002).

The building of programming knowledge and the development of programming skills present many difficulties mostly for novice programmers (DuBoulay, 1989; Brusilovsky *et al.*, 1997; Grigoriadou *et al.*, 2002). Various instructional approaches have been proposed in order to face the difficulties encountered by learners and to promote their active involvement in the teaching-learning process (Soloway, 1986; Spohrer and Soloway, 1986; Pane and Myers, 2000). Many educational programming languages and programming environments have been developed to support the teaching of programming and to facilitate novice learners. The success of Logo – even though it wasn't developed for the learning of programming – and of the turtle graphics boosted the development of the mini-language approach for the teaching of programming. The key concept of the mini-language approach is based on designing a short and simple language to support the first steps in the learning of programming. Regardless of the penetration level to programming and the student's age, the study of a mini-language has positive results in the development of algorithmic reasoning and problem solving skills (Brusilovsky *et al.*, 1997).

As novice programmers come for the first time in touch with a programming environment, they usually meet with a relatively complex interface and a variety of features, functions and tools incorporated in the environment which is not needed from the beginning. An appropriate environment for novice learners should be developed in gradual steps and according to their acquired experience. Through applying subtotals of the programming language, novice programmers show better learning results in less time and feel less discouraged thanks to a less complex and more intuitive interface (DePasquale, 2002).

Programming skills are consolidated when students can use concepts that they have developed in previous programming activities in order to deal with problems of increasing complexity. By decreasing the interface complexity in programming environments and by implementing controlled access to language structures, we can achieve an improved understanding of programming, as well as a reduction of the cognitive load and pressure imposed on the learners (DePasquale, 2002).

Logo is considered an important tool in the hands of teachers and students for the development of their exploration skills, creativity skills, problem solving skills and for the cultivation of logical-algorithmic reasoning (Papert, 1980; Harel and Papert, 1991; Clements and Meredith, 1993; Kafai and Resnick, 1996; Turcsányi-Szabó, 1998; Hoyles *et al.*, 2002; Dagiene, 2003; Resnick *et al.*, 2003). The users become, at the same time, users and designers ("good" programming languages involve the programmer in the role of the language designer) as they design and construct tools and objects for the solution of problems. This double role of the user leads directly to the notion of constructionism.

Constructionism involves two interweaving types of construction: knowledge construction through construction of artifacts with personal meaning (Kafai and Resnick, 1996; Harel and Papert, 1991).

The microworld concept has been present for over four decades now and the exploitation of microworlds in education has triggered the interest and attention of many researchers and instructors who plan, experiment with and explore alternative constructionist approaches in various thematic fields (Harel and Papert, 1991; Clements and Meredith, 1993; Kafai and Resnick, 1996; Turcsányi-Szabó, 1998; diSessa, 2000; Hoyles *et al.*, 2002; Resnick *et al.*, 2003; Kalas, 2006; Brouwer *et al.*, 2007; Glezou and Grigoriadou, 2009a, 2009b). A microworld must be defined at the interface between an individual user in a social context and a software tool possessing the following five functional attributes: a) it is domain specific; b) it provides a doorway to the domain for the user by offering a simple example of the domain that is immediately understandable by the user; c) it leads to activity that can be intrinsically motivating to the user – the user wants to participate and persist at the task for the same time; d) it leads to immersive activity best characterized by words such as play, inquiry, and invention; and e) it is situated in a constructivist philosophy of learning (Rieber, 2004). The best microworlds have an easy-to-understand set of operations that students can use to engage tasks of value to them and, in doing so, they come to understanding powerful underlying principles (diSessa, 2000).

Logo-like environments can be used to plan and develop microworlds that offer students the possibility to express and exploit their thoughts, ideas and feelings and support the process of building knowledge by creating learning environments rich in speculation and opportunities for experimentation (Hoyles *et al.*, 2002; Resnick *et al.*, 2003). Pre-constructed microworlds are meant to operate as starting points and idea generators for building on them, changing them or decomposing parts of them in order to construct a new artifact (Kynigos, 2006; Glezou and Grigoriadou, 2009a). It is difficult to provide students with building blocks sufficiently powerful to create models, yet sufficiently flexible and transparent to encourage students to question their inner workings. Attempting to strike the right balance between functionality (the tools do a useful job) and transparency (the tools can be inspected, manipulated and modified) remains a key priority for future iterations (Simpson *et al.*, 2005).

As LCSi (<http://www.microworlds.com/>) claims MicroWorlds is a Logo-like environment in which students can explore and test their ideas as they create animations, games, science simulations, mathematical experiments, and interactive multimedia stories, whatever, limited only by the limits of their imagination. Many researchers and educators have proposed various constructionist approaches in order to use MicroWorlds and promote students active involvement in the teaching-learning process (Brouwer *et al.*, 2007; Dapontes *et al.*, 2003; Glezou and Grigoriadou, 2009b). The Greek version of MicroWorlds was adapted and spread in the Greek educational community in 2002 and since then a significant number of Greek students, teachers, educators and researchers has become enthusiasts fans of it as a powerful multimedia programming environment and authoring tool that supports programming in Greek language and used it in different educational settings (classrooms, seminars, workshops, books) (Dapontes *et al.*, 2003;

Komis, 2005; Glezou and Grigoriadou, 2009b). It should be mentioned that the Greek version of MicroWorlds Pro is also used as the basic programming environment in the new didactic package (students' and teachers' books) for the Junior High Informatics course according to the novel Greek cross-thematic curriculum framework for compulsory education since 2006.

Open exploratory learning environments, such as Logo-like environments, require guidance which should not be didactical, but interventional and supportive, while quite often Logo's learning effectiveness is linked to the presence of the teacher's guidance in the form of discussion or in the form of worksheets (Dapontes *et al.*, 2003; Glezou and Grigoriadou, 2009b).

The creation of interesting and demanding environments encouraging the active and constructive participation of students is a great challenge for teachers. The planning of a learning environment includes extensive decision making for planning, which should be the result of conscious thought rather than an unconscious choice (Vosniadou, 2005). In Ackermann's words: "A rich learning environment is one that offers the freedom for genuine exploration, reflection, expression and negotiation, while at the same time providing help and support, when needed. Needless to say, it is not easy to decide how much freedom or guidance makes for a nurturing and yet challenging learning experience. And different people need different kinds of feedback, at different times, in different situations! A good clinician, like a good teacher, is someone who masters the art of providing the "right" amount of elbowroom in each singular case" (Ackermann, 2003, p. 35).

Learning occurs through a process of continuous changes in the individual's cognitive structures and is directly linked to the effects of the sociocultural environment (Vygotsky, 1978). At the same time, the context in which learning takes place, as well as the tools' mediation play a crucial part providing opportunities for active, exploratory and personally significant learning for the individual. The design, development and implementation of the proposed educational scenario and activities are based on a framework of general pedagogical principles that constitute the resultant force of the ideas developed through the last years in the field of cognitive and sociocognitive theories, based on the social constructive model. The basic axes for the design of educational scenario and activities suggested are:

- structuring exploratory and creative student-teacher roles,
- supporting the process of active knowledge building,
- exploiting students' previous knowledge, experiences and intuitions,
- using a cross-thematic approach, and,
- developing collaborative learning environments.

3. Research Framework

The present study is part of a wider research, which aims to explore the potential of implementing a cross-thematic educational scenario that promotes collaborative exploratory learning as a framework for activities and for a series of lessons. The aim of this research

is to bring forward the basic parameters of an effective interdisciplinary constructionist approach, in the level of designing, development and implementation.

We considered that MicroWorlds Pro (Greek version 1.1) is an appropriate multimedia programming environment for the development of microworlds, models, simulations, multimedia applications as projects, in the framework of implementing a cross-thematic educational scenario.

The basic research questions of the particular study are:

- Which instructional strategy supports better student engagement in simulation development: simulation development from scratch or use of a preconstructed microworld?
- How do the students collaborate and interact with the programming environment?
- Which are the special features of MicroWorlds Pro that contribute to or cause difficulty in the development of an effective learning environment?

It is a case study that uses ethnographic and action research elements since the researcher was also the teacher of the class.

In the framework of the educational scenario “Free fall simulation development”, after the setting of the teaching objectives, we structured a series of lessons in phases; each phase in stages and each stage in distinct steps. Then, we developed a) microworlds in MicroWorlds Pro, b) activity worksheets-lesson plans, c) student worksheets and d) teacher worksheets. The research tools were the above mentioned, as well as the Greek version of MicroWorlds Pro (Greek version 1.1).

We collected data from the researcher’s notes-diary after each didactic hour, the students’ notes-drafts, the filled-in worksheets, the microworlds and final projects of the students, as well as from semi-structured interviews of students. Then, the data underwent a qualitative analysis, whose results led to modifications in the ergonomics, the appearance and the function of the microworlds, as well as in the gradual ameliorative reshaping of the lesson plan and the worksheets.

The suggested cross-thematic scenario was implemented in the framework of the “Multimedia-Networks” course in two classes of the 3rd grade of A and B Arsakeio General Senior High School of Psychiko in Athens, during the first four-month period of the school year 2006–2007. “Multimedia-Networks” is an optional lesson without written examination, which students of all three orientations (theoretical, scientific and technological) can take up, and therefore it requires special treatment due to the heterogeneity of the students. This heterogeneity has to do with the basic computer skills and with programming skills in particular, as well as with the different cognitive level of the students in Physics and Mathematics. In addition, a significant 30% of students showed reduced interest and negative spirit towards the course.

All the students had already been exposed to the MicroWorlds Pro environment and the basic Logo commands in previous lessons (one didactic hour per class) that enabled the students to familiarize themselves in a first degree with the environment’s basic features and the construction of microworlds by adding cartoons and multimedia elements (e.g., sound, video).

In the present study, which took 6 didactic hours per class, the participants were 2 classes of 9 teams – 18 students (8 girls – 10 boys) and 11 teams – 23 students (14 girls

– 9 boys) respectively. The students were separated in small groups of 2 per computer of their own choice. Due to the odd number of the 2nd class, there was also one team (team #9 of 2nd class) with three students. Both classes had a similar programming background and similar averages of all disciplines.

We implemented two alternate instructional strategies (one for each class): a) simulation development from scratch and b) use of a preconstructed microworld.

The preconstructed microworld “Free fall simulation development” (Fig. 1) consists of four pages characterized by a gradually increasing complexity that corresponds to the particular first three phases of the scenario:

- on page 1 of the microworld we have the construction of the free fall simulation (corresponding to Phase A of scenario);
- on page 2 we have the construction of the free fall’s stroboscopic representation (corresponding to Phase B of scenario);
- on page 3 we have the construction of the free fall’s stroboscopic representation and the exploitation of a table exhibiting the values of time and position (corresponding to Phase C of scenario (Stage 4 – 7th Step, as described in lesson plan in Section 3.2));
- on page 4 we have the construction of the free fall’s stroboscopic representation and the exploitation of a table exhibiting the values of time, position and velocity (corresponding to Phase C of scenario (Stage 4–8th and 9th Step, as described in lesson plan in Section 3.2)).

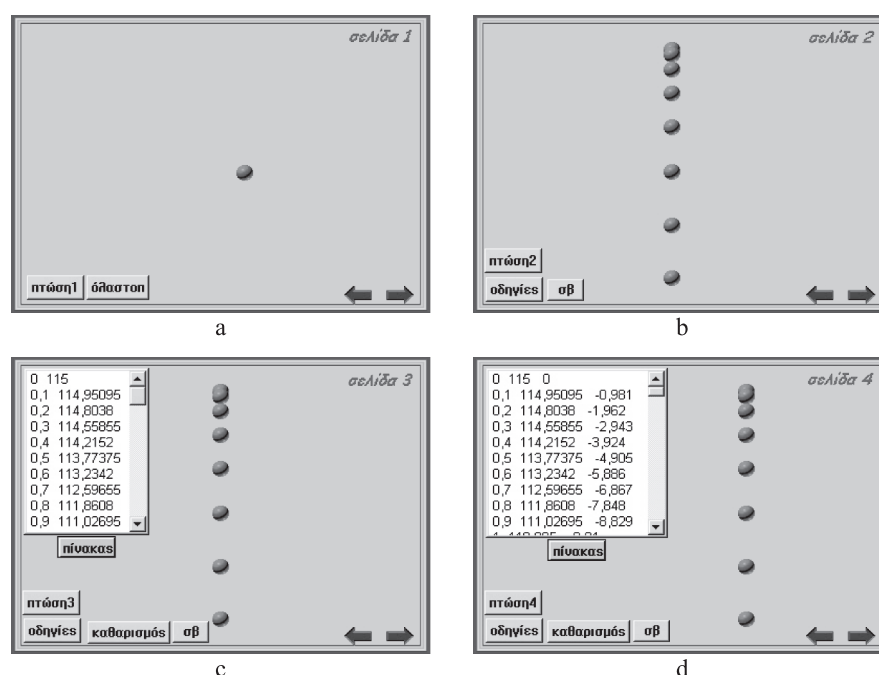


Fig. 1. Snapshots of the “Free fall simulation development” preconstructed microworld pages: a) 1st page, b) 2nd page, c) 3rd page and d) 4th page.

In the 1st class, the students were asked to gradually build the microworld of free fall simulation, from scratch, following the steps of appropriately structured worksheets which correspond to the first three phases (Phases A, B and C) of the scenario.

In the 2nd class, the students were asked to experiment with a given preconstructed microworld (Fig. 1) and extend/modify it as explicitly described in the lesson plan in Section 3.2.

In the last two phases (D, E) of the scenario, the students of both classes were asked to proceed to the construction of a variety of simulations, multimedia applications and webpages.

It is worth mentioning that only the first three phases (Phases A, B and C) of the scenario differentiate in the 1st and 2nd class whereas the last two phases (D, E) remain common for both classes.

In the following, we present the educational scenario: “Free Fall Simulation Development” discussing special issues of interest.

3.1. Educational Scenario: “Free Fall Simulation Development”

The cross-thematic scenario “Free fall simulation development” creates an open and flexible framework of activities, where elements of Informatics are linked with Physics and Mathematics focusing on the development of microworlds as models, simulations and multimedia applications.

The “Free fall simulation development” scenario’s research item, as its title suggests, is the development of a free fall simulation (without friction and resistance). The construction of a free fall simulation in the MicroWorlds Pro environment requires the knowledge of the free fall laws (Physics knowledge of the 1st grade of Senior High School according to Greek educational system) and, at the same time, programming skills in Logo language. This is a circular process of successive transitions between the actual motion of the object and its simulation.

The scenario’s activities evolve in five distinct phases:

- Phase A – construction of the free fall simulation.
- Phase B – construction of a stroboscopic representation of the free fall.
- Phase C – construction of a stroboscopic representation of the free fall and exploitation of a table exhibiting the values of time, position (ordinate Y) and velocity.
- Phase D – construction of a variety of free fall simulations.
- Phase E – construction of a multimedia application and webpages / presentation of the project.

The structure of the evolutionary phases and respective activities is characterized by a gradually increasing degree of complexity and difficulty (Dapontes, 2005).

The students would gradually reach a higher level of familiarization with Logo programming and the MicroWorlds Pro environment, while constructing their knowledge, developing skills based on previous experience, analyzing, composing or/and expanding Logo code.

The selection of the specific successive activities is based on the view that the construction of new situation facing models with a gradually increasing degree of complex-

ity and difficulty, by re-using already created models, encourages the systematization of knowledge and bridges the gap between the simple and the more complex.

Each new activity is based on the skills acquired during the previous activity, functioning as scaffolding for the development of reasoning processes and as a systematic way of familiarizing with the environment's tools (Glezou and Grigoriadou, 2009a).

The estimated total time for the activities corresponds to 5–9 didactic hours and the suggested hour distribution is the following: Phase A: 1 didactic hour, Phase B: 1–2 didactic hours, Phase C: 1–2 didactic hours, Phase D: 1–2 didactic hours, Phase E: 1–2 didactic hours.

The scenario's nature is indicative and not exemplary. The teachers, based on their personal interest, can reform it or build new personal scenarios and activities.

As above mentioned, the suggested cross-thematic scenario was implemented in the framework of the "Multimedia-Networks" course and in the field of Informatics, as far as the activities were concerned, our aim was to cultivate the skills of designing, developing, exploring and managing microworlds as models, simulations and multimedia applications, helping students familiarize, at the same time, with the basic concepts of programming, understanding and writing Logo programs.

According to Greek educational system, the particular scenario and its activities could be incorporated as well in the Physics course of the 1st grade of Senior High (in the didactic unit, where free fall laws are included), in the Informatics Applications course of the 1st and 2nd grade of Senior High, by properly adjusting the goals and the didactic process.

It should be noted that the selection of the activities, the activities' teaching-learning goals, the final planning, the time schedule and the lesson plan are determined accord-

Table 1
Structure of the lesson plan (according to the three elements: Phase, Stage and Step)

Phases	Stages	Steps
Phase A	Stage 1	Step 1 Step 2
	Stage 2	Step 3
Phase B	Stage 3	Step 4 Step 5 Step 6
		Step 7 Step 8 Step 9
		Step 10
Phase D	Stage 5	Step 11 Step 12
	Stage 6	

ing to the students' level of knowledge and previous experience, their interests and the didactic goals set by the teacher.

Below is presented the lesson plan which was implemented in the 2nd class following the teaching strategy of the use of the preconstructed microworld.

For clarity reasons, the relationships of the three elements (Phase, Stage, and Step) of the lesson plan are summarized in the Table 1.

3.2. Lesson Plan (Implemented at the 2nd Class)

Phase A

Stage 1 (1st and 2nd Step): Revocation of Previous Knowledge – Connection with Everyday Life

In the 1st step the students are asked to re-invoke their previous knowledge or/and experience that has to do with the free fall and the equations that regulate it.

In the 2nd step there is a discussion about the concept of simulation, of chronophotography or stroboscopic representation and of the modeling process.

Stage 2 (3rd Step): Experimentation with the Free Fall Simulation

In the 3rd step the students are asked to open the relevant microworld of MicroWorlds Pro, then click on the button "fall1" in page 1 (Fig. 1a) and observe the movement of the turtle shaped as a sphere.

The students are encouraged to go to the Procedures Tab of the environment and study the initial code of the program (Table 2).

The initial code in the Procedures Tab of the environment includes three procedures: start1, movement1 and fall1 (as presented in the column named Procedures in the Table 2) and, in parallel, the explanatory remarks (as presented in the column named Explanatory Remarks in the Table 2). It should be noted that the procedure fall1 calls the sub-procedures: start1 and movement1. For clarity reasons, the presented code has been translated and adapted from Greek to English.

What follows next is a gradual analysis of the code, the program's structure, the basic commands and procedures' syntax, as well as the use of the variables mentioned in the code.

Finally, there is a discussion whether it is possible to improve/extend the code. A crucial question is emerging: how can we modify the previous code in order to program a stroboscopic representation of the free fall?

Phase B

Stage 3 (4th, 5th and 6th Step): Development of a Stroboscopic Representation of the Free Fall

In the 4th step the students are asked to go to page 2 of the microworld and click on the button "fall2". The students observe that nothing happens and, then, go to the Procedures Tab, where the procedure fall2 does not exist.

Table 2

The code of the program “Development of the Free Fall Simulation” and explanatory remarks on commands (Phase A)

Procedures	Explanatory remarks
to start1	;initialization procedure
t1,	
make "y0 last pos	;storage of position ordinate y
make "g -9.81	;storage of value of acceleration of gravity
make "t 0	;storage of start counting time
make "dt 0.001	;storage of time step
end	;procedure definition end
to movement1	;movement procedure
sety :y0 + 0.5 * :g * :t * :t	;turtle's placement to ordinate y
make "t :t + :dt	;time increasing up dt
movement1	;recall of movement procedure
end	;procedure definition end
to fall1	;fall procedure
start1	;call of initialization procedure
movement1	;call of movement procedure
end	;procedure definition end

The students are encouraged to define the procedure fall2 as well as the prerequisite component sub-procedures start2 and movement2 (Table 3) by re-using the initial code (as presented in Table 2) and modifying it appropriately in order to proceed to the development of a stroboscopic representation of the free fall.

As the students proceed to the sequence of activities, in order to gain time while keyboarding the appropriate code, we recommend students to copy-paste the code of the corresponding initial procedures from and to the Procedures Tab, so that they can later modify/extend it appropriately and move on to the next step. In each step the corresponding initial code is extended by adding to it a new command or a small sequence of commands in order to scaffold the familiarization with gradually increasing complexity.

In the 5th step the teacher introduces the conditional statement for the immobilization of the turtle and the command stamp and analyzes the appropriate use of them.

In the 6th step, after their attempts (maybe some successful or/and unsuccessful) to modify the code with the necessary feedback and discrete guidance from the teacher, the students succeed in developing the necessary code for the construction of a stroboscopic representation of the free fall. Afterwards, the students are asked to go back to page 2 of the microworld (Fig. 1b), click the button “fall2” and observe the free fall simulation evolve in a stroboscopic representation.

Table 3

The code of the program “Development of a Stroboscopic Representation of the Free Fall” and explanatory remarks on commands (Phase B)

Procedures	Explanatory Remarks
to start2	;initialization2 procedure
t1, sety 115	;turtle’s placement to ordinate y = 115
make "y0 last pos	;storage of position ordinate
make "g -9.81	;storage of value of acceleration of gravity
make "t 0	;storage of start counting time
make "dt 0.001	;storage of time step
make "N 0	
end	;procedure definition end
to movement2	;movement2 procedure
if ycor < -115[stopme]	;stoppage condition check
sety :y0 + 0.5 * :g * :t * :t	;turtle’s placement to ordinate y
if :t = :N[stamp make "N :N + 1]	;turtle leaves trace
make "t :t + :dt	;time increasing up dt
movement2	;recall of movement2 procedure
end	;procedure definition end
to fall2	;fall2 procedure
start2	;call of start2 procedure
movement2	;call of movement2 procedure
end	;procedure definition end

Phase C

Stage 4 (7th, 8th and 9th Step): Development of a Stroboscopic Representation of the Free Fall and Exploitation of a Table Exhibiting the Values of Time, Position and Velocity

In the 7th step the students are asked to go to page 3 of the microworld (Fig. 1c), click on the button “fall3” and observe the evolution of the stroboscopic representation of the free fall and the simultaneous record of the values of time (t) and ordinate (y) in the text box “Table”.

In the 8th step the students are encouraged to go to the Procedures Tab and analyze the code of the procedure fall3, focusing on the syntax of the command that answers for the record of the values of time and of ordinate y . Then, the students follow the clarifications that have to do with the lists and are asked to modify properly the code of the procedure fall3 on the Procedures Tab, in order to define the procedure fall4 so as to simultaneously record the values of time (t), ordinate y (y) and velocity (v) on the text frame “Table” of the environment.

In the 9th step, with the appropriate feedback and guidance from the teacher, the students finally define the procedure fall4 in order to build the code for the simulation of the free fall with a stroboscopic representation and a simultaneous recording of the values

of time (t), ordinate y (ord y) and velocity (v) on the text frame “Table”. Afterwards, the students are asked to go to page 4 of the microworld (Fig. 1d), click on the button “fall4” and observe the evolution of the free fall and the simultaneous recording of values.

Phase D

Stage 5 (10th Step): Development of a Variety of Simulations

In the 10th step the students are asked to create develop the code of various procedures and insert new pages in the microworld, by importing turtles and various objects of their choice in it, in order to make a variety of free fall simulations, experimenting in the microworld’s environment.

Phase E

Stage 6 (11th Step): Development of a Multimedia Application and of Webpages

In the 11th step the students are asked to create a multimedia application and webpages of their own choice, combining the free fall simulations they made in a unified whole (see Figs. 2 and 3).

Stage 7 (12th Step): Presentation of the Project

In the 12th step the students present the project they made in class, they express and discuss their impressions and experience, as well as the problems they faced.

4. Findings

From the data analysis of this pilot implementation (the application is in evolution so we present only indicative facts here) we should stress the following:

The process had a positive effect on both classes.

The students showed interest in their interaction with the MicroWorlds Pro environment and remained active during the lessons, especially in the phase where they had to create their own simulations and multimedia applications. Students that were not initially interested in the “Multimedia-Networks” course were motivated and cooperated satisfactorily. Yet there were five groups (Teams 1, 3, 7 of 1st class and Teams 1, 7 of 2nd class) that cooperated only fragmentarily. The students, in many cases, were enthusiastic after being exposed to the “purely” programming part of the lesson and admitted having a “live” interaction with the environment. While investigating each program in the activity sequence, students went repetitively through a cycle of prediction, modeling in Logo, model-testing and consideration of the limitations of any current model’s scope of application.

Actions such as the planning of the desired interface surfaces, the recognition of a command sequence and the definition of a procedure are considered to be indicative of the high level skills and goals’ development, such as the recognition of a procedure function and the definition of new procedures. The repetitive shift between the code analysis

and its reforming/extension, which allows the shift from simple to complex and the gradual familiarization with the programming language within the scaffolding process, has proved to be equally effective.

During the process of developing Logo code and of debugging, the questions posed by the students were often fixed and imperative. The teacher needs to proceed in subtle handlings, in order to accommodate the students with no more support than what they need.

It was observed an increased difficulty due to the students' lack of familiarity with the syntactic rules of Logo, something which often led to their disappointment and their urge to quit. This applies particularly to students who had never wished or had the chance to familiarize themselves with programming. Actually it was more obvious among the students of theoretical orientation. The following mistakes were observed on a regular basis: the students left no space between the operators, in the procedure name they left a space between two words (e.g., movement 1 instead of movement1), they used the letter "o" instead of 0 (zero), they omitted semicolons before the variable, or forgot to put the word end at the end of the procedure definition.

As the students of both classes proceed to the sequence of relevant activities during the phases A, B and C, in order to gain time while keyboarding the appropriate code, we recommend students to copy-paste the code of the corresponding initial procedures from and to the Procedures Tab, so that they can later modify/extend it appropriately and move on to the next step. In each phase the corresponding initial code is expanded by adding to it a new command or a small sequence of commands in order to scaffold the familiarization with gradually increasing complexity.

The analysis of students' answers on the worksheets revealed that: (a) students in all ability levels made gains in programming and modeling skills; (b) students made significant gains in their ability to answer items covering initial conditions, condition value, variables, and program flow.

In the 2nd class, the preconstructed microworld had a positive effect, especially for the students who had no experience at all in programming. We consider that the preconstructed microworld functioned as a good starting point, as a solid ground for various alternative explorations, modifications and extensions, as a vehicle for collaboration and negotiation which promoted a better performance.

In the phase where they were asked to create a variety of simulations, the students demonstrated their inventiveness; they built alternative personal models with different levels of detail and variations of methods. Following the teacher's suggestion, some groups looked for relevant material on the Internet and in the installation folders of the softwares available (like Modellus: another computer tool for modeling and experimentation), in order to enrich their pages with images, videos etc. Groups of 2nd class were more creative; they construct a richer variety of multiple more detailed alternate models. Three groups of 2nd class went on further to create simulations of different phenomena such as uniform translatory motion and symmetrical oscillation.

In Fig. 2 we present indicative snapshots of the microworld of a students' project work (Team 3 of 2nd class), as resulted from the preconstructed microworld modification. We

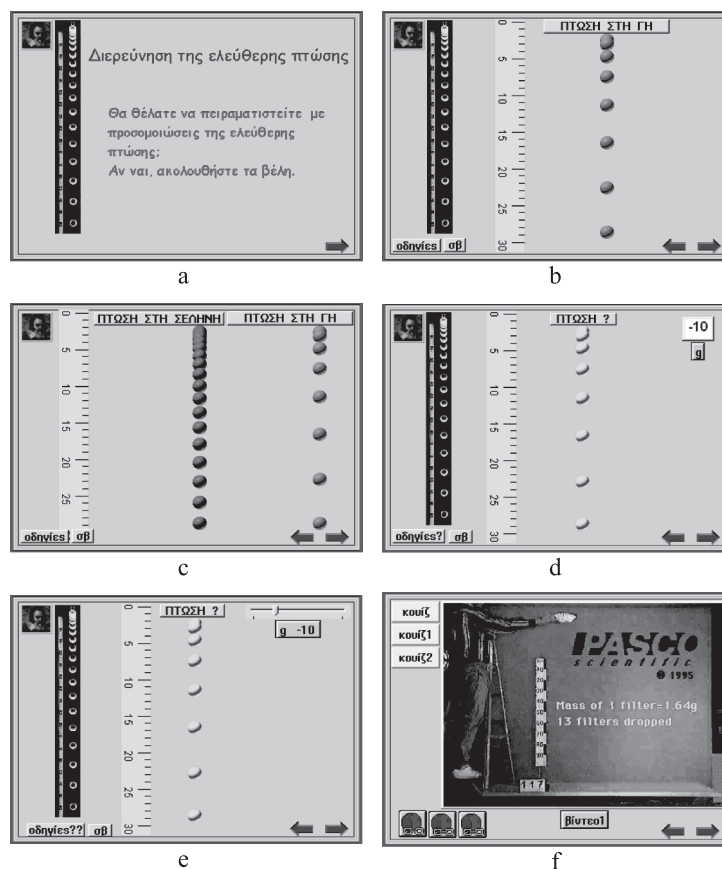


Fig. 2. Snapshots of the microworld of the students’ project work (Team 3 of 2nd class).

notice enriched multiple pages with additional material that was found on the Internet and in the installation folders of softwares available (like Modellus: another computer tool for modeling and experimentation).

In Fig. 2a the students added an introductory page, calling to “Free fall investigation” and enriched with images (like an image of Galileo, an image of Free fall stroboscopic representation).

In Fig. 2b the students defined a new procedure as “Fall_on_earth”, renamed properly the corresponding button and added one turtle shaped as a ruler to measure the distance between the ball’s stamps.

In Fig. 2c we see a parallel stroboscopic representation of the free fall on the earth and the moon (by clicking buttons corresponding to appropriate procedures defined as “Fall_on_earth”, “Fall_on_moon”).

In Figs. 2d and 2e, the students proceeded to the parameterization procedure. They suitably defined the parametric procedure as “Fall_?” and then used a text box (Fig. 2d) and a slider (Fig. 2e) to input the value of variable g (where g : acceleration of gravity). In

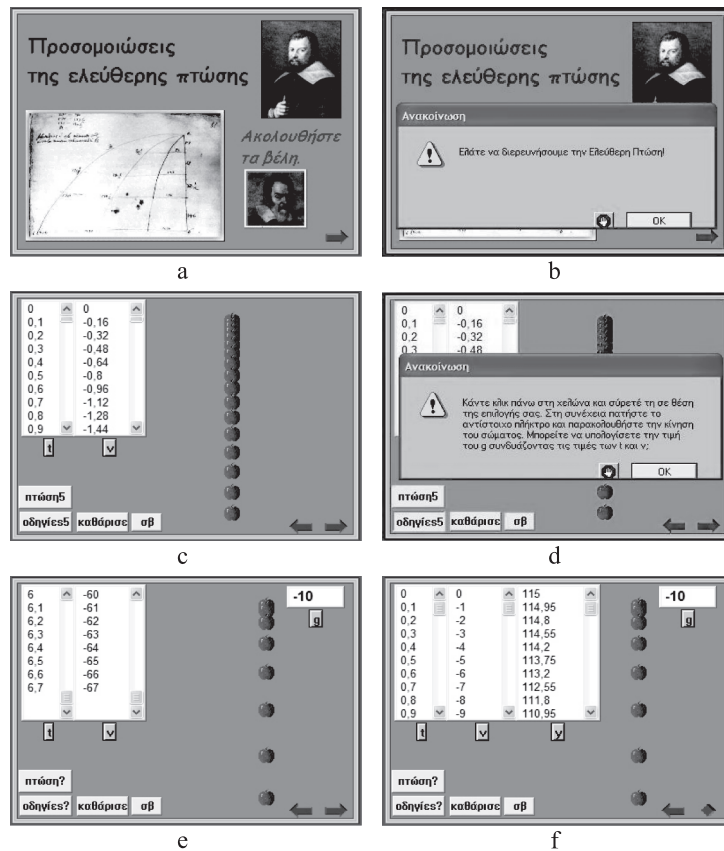


Fig. 3. Snapshots of the microworld of the students' project work (Team 8 of 2nd class).

Fig. 2f, the students added a video-experiment relating to the investigation of the free fall phenomenon (found in the installation folder of Modellus), hyperlinks to websites where relevant educational material can be found, as well as new buttons linked with procedures dealing with quizzes.

In Fig. 3, snapshots of the microworld created by another team (Team 8 of 2nd class) are presented.

In Figs. 3a and 3b we notice the introductory page enriched with images and an announcement emerging in the presentation mode of the microworld, with the message "Let's investigate the Free Fall".

In Fig. 3c, we see the turtle shaped as an apple, two different text boxes as tables to output the values of time (t) and velocity (v) as opposed to the single table used in the preconstructed microworld.

In Fig. 3d, the students gave the microworld a game dimension. As we see, an emerging announcement is calling us to observe the movement and calculate the value of g , combining the values of time (t) and velocity (v) from the text boxes.

In Figs. 3e and 3f, the students used a parametric procedure and a text box to input the value of variable g (g : acceleration of gravity).

Table 4
Skills development per team

Skills	# Team 1st class									# Team 2nd class										
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	10	11
Code modification	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use of new commands		✓		✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
New procedures development	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
New simulations development		✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
New objects import		✓			✓			✓	✓	✓	✓			✓	✓		✓	✓	✓	✓
Webpages development		✓		✓		✓		✓	✓	✓	✓	✓			✓		✓	✓	✓	✓

In Fig. 3f also, we can see three text boxes as tables to output the values of time (t), velocity (v) and position (y) (: ordinate Y).

The results of the analysis of students’ microworlds and final projects are presented in Table 4. It is evident from the checkpoints in the matrix that the 2nd class accomplished a better final result regarding the observed skills development per team compared to the 1st class.

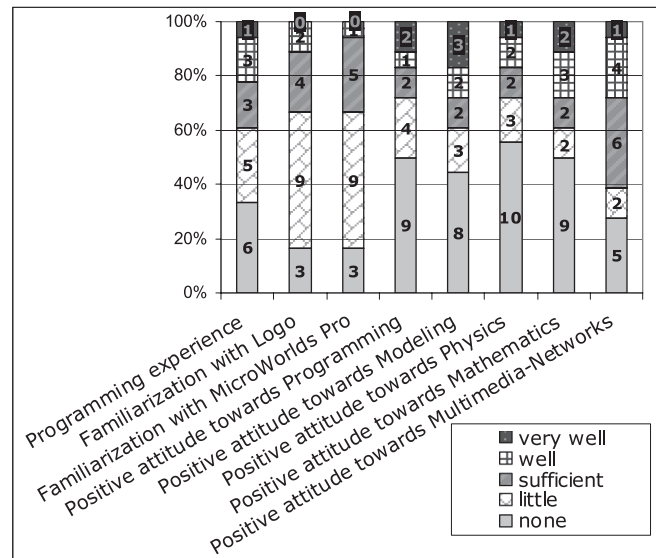
Follow-up interviews of students revealed that the experience of both instructional methods led to a reduction of the anxiety towards programming and to an increased willingness to approach programming challenges in a more positive attitude.

The results from the structured part of the interviews are presented for the 1st class (18 students) in Fig. 4 regarding a) the prior experience and b) the acquired experience after the lessons. The corresponding results for the 2nd class (23 students) are presented in Fig. 5 regarding a) the prior experience and b) the acquired experience after the lessons.

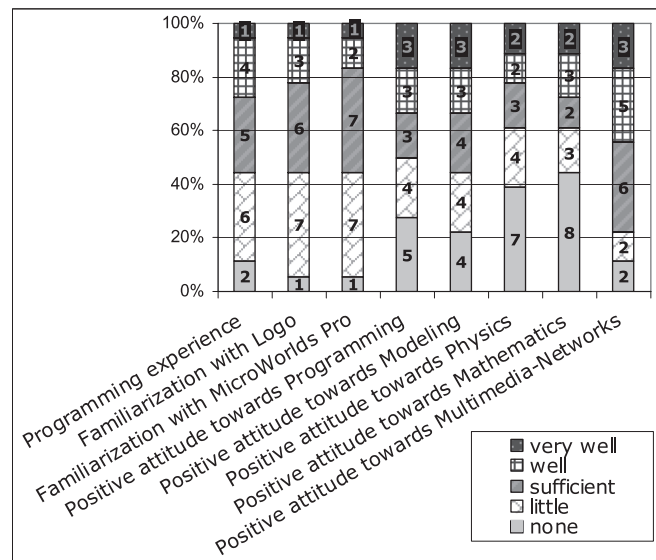
To measure the students’ responses to the items – regarding a) the prior experience and b) the acquired experience after the lessons – of the structured part of the interviews post-evaluation, a five-point Likert scale from 1 to 5 was used, where 1 was coded as the lowest and 5 as the highest (1 = “None”; 2 = “Little”; 3 = “Sufficient”; 4 = “Well”; 5 = “Very well”). According to students’ answers the teacher as the interviewer placed a cross “X” in the appropriate box of a matrix using the above scale.

These results indicate that the majority of students of both classes stated that they developed an increased programming experience, familiarization with Logo and MicroWorlds Pro, as well as an increased positive attitude towards programming, modeling, Physics, Mathematics and Multimedia-Networks.

The general interview results show that the students in both classes were satisfied with their participation in the course, with the knowledge they acquired as well as the artifacts they constructed. They also appeared to enjoy the lessons, and they admitted that



a



b

Fig. 4. Results from the structured part of the interviews for the 1st class (18 students) regarding a) the prior experience and b) the acquired experience after the lessons.

they have found the teaching approach and material “very intriguing”, “effective”, “really scaffolding”, “helpful to learn how to learn”, and “focus on construction”.

Among the special features of MicroWorlds Pro that contribute to the creation of an effective learning environment we should mention the following: The options Duplicate Page and New Page from the Pages Menu of the environment have proved to be partic-

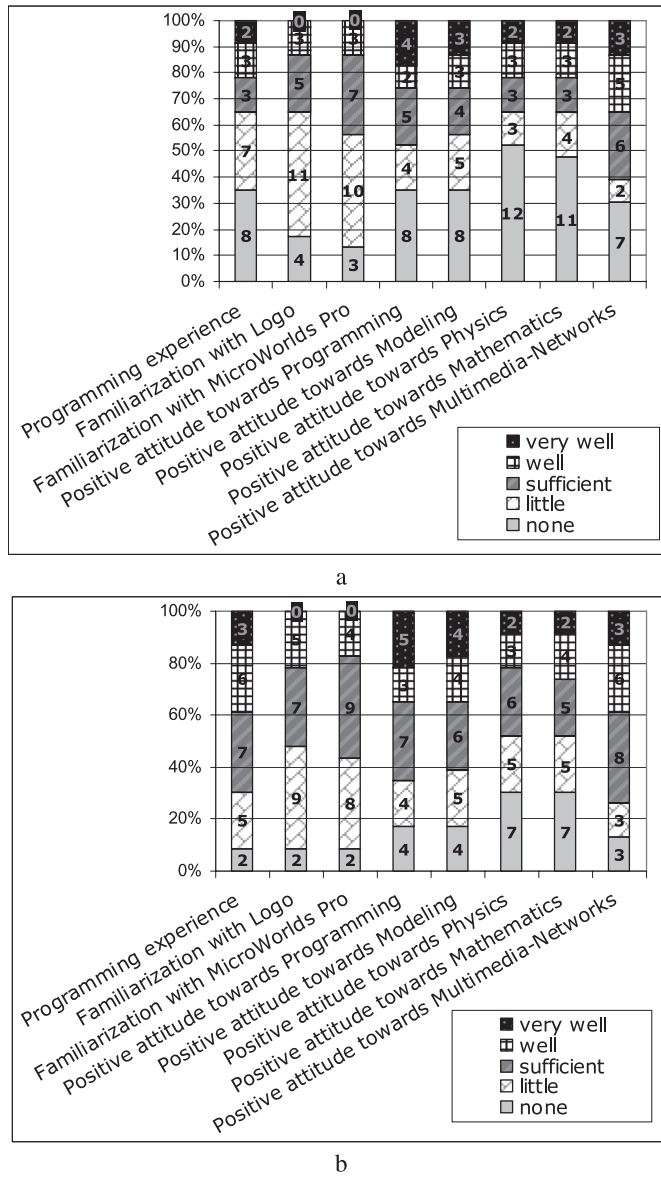


Fig. 5. Results from the structured part of the interviews for the 2nd class (23 students) regarding a) the prior experience and b) the acquired experience after the lessons.

ularly functional. In each new phase of the project, we encourage students to reproduce a selected page or/and add a new page in their microworld, in order to be able to trace back the sequential stages of their work, reflect upon them and reconsider their choices, as well as reuse the elements of the selected page and proceed to the necessary changes. In addition, the option Presentation Mode from the View Menu is functional to frame and

demonstrate completed projects, as it centers the project on the screen and sets the area around the project to black, framing the presentation by hiding the Command Center, Tab, Toolbar, MicroWorlds Menus and Status Bar. The Project Tab displays the complete state of the project including all objects, whether they are hidden or not, as well as the state variables. This has proved to be especially useful for analyzing a project, finding unused objects on a page and/or proceeding to subsequent modifications.

Below, we present a characteristic episode as an indicative example of the collaborative, exploratory learning atmosphere developed during the interaction between the teacher, the students of a group (Team 4 of 1st class) and the computing environment.

4.1. *Characteristic Episode in a Group of Students*

This episode took place in the 1st phase of the construction simulation activities, while the students analyze the code in order to grasp the commands that form the procedure defining the initial conditions of the fall, and particularly of the turtle's initial position ordinate. In order to be more tangible, it should be noted that in Greek version of MicroWorlds Pro, the first turtle that comes in the scene is named by default $\chi 1$ (where χ : the initial letter of Greek word "χελώνα" that means turtle in English) instead of t1 (where t: the initial letter of English word "turtle" in English version of the environment). This often leads Greek students to confusion as χ is considered the abscissa.

T: Try out the command: χ , show last pos;

S2 enters in the Command Center the command: $\chi 1$, show last pos and executes the command by pressing enter. While the command is executed, an error message appears in the Command Center: I don't know anything about $\chi 1$;

S1 (addressing to S2): Oops! What does it say there?

S2 (addressing to S1): Don't you know how to read?

S2 (addressing to the teacher): On the screen there is a message saying: I don't know anything about $\chi 1$;

S1 (addressing to the teacher): What is $\chi 1$;

T: Think. . . What do we call $\chi 1$;

S1: Umm. . . don't we usually call $\chi 1$ the turtle's ordinate?

S2: What turtle's?

S1: Oh! No! We didn't put a turtle! $\chi 1$ is the turtle's name;

S2: Let's put one. Let's see now. . .

S2 inserts a turtle in the page and executes the command again: $\chi 1$, show last pos in the Command Center;

S1: 63! This must be the abscissa!

S2: The abscissa or the ordinate?

S1: Oh! It's the ordinate that we're looking for? Then it must be the ordinate!

S1: I don't know. . . I found it! Let's try out the command: show pos;

S1 enters in the Command Center the command: x1, show pos and executes the command. While the command is executed, an error message appears: I don't know anything about $\chi 1$, show;

S2: You didn't put in a space, stupid!

S1: Oh, ok!

S1 corrects in the Command Center the command: x1, show pos and executes it again. On the next line, right under the command in the Control Center, appear the numbers: 30 63;

S2: 30 the abscissa and 63 the ordinate;

S1: Are you sure?

S2: I'm positive! Try the command: show first pos and you'll see.

As we can see from the above, the students explore the coordinates of the turtle's position, they formulate hypotheses, check whether their hypotheses are correct, affirm or re-examine their presumptions and correct themselves.

In their experimentation process, students receive direct feedback from the environment; they negotiate, cooperate, criticize one another, evaluate themselves, take turns in the first and second role, suggest ways of coping with new situations and try out new commands. They become more active and their centre of interest is transferred easily, tracing new exploration paths.

The teacher urges students to experiment, observe and reflect. She doesn't answer the question posed to her directly, but poses a new question that functions as a stimulus for further reflection and exploration.

5. Conclusions

This paper presented a small-scale study investigating the use of the MicroWorlds Pro multimedia programming environment as an authoring tool for constructing models, simulations and multimedia applications with students of Senior High School.

The findings reaffirm the view that the use of open software, like MicroWorlds Pro, cultivates creativity, gives the teacher pedagogical freedom and supports the students' active engagement and learning.

The preconstructed microworld functioned as "object to think with", a good starting point and a solid ground for explorations-modifications-extensions, as a vehicle for collaboration and led to various alternative constructions of personal and social meaningful artifacts.

The proposed educational scenario and the structured series of activities characterized by a gradual increase in complexity and difficulty functioned pretty well as scaffolding during the gradual familiarization with simulation development, Logo programming and the programming environment MicroWorlds Pro.

The implementation in actual classroom circumstances and, as a matter of fact, in different classes on successive days, provided us with important feedback that led to modifications/interventions in the ergonomics, the appearance and function of the microworlds, as well as in the ameliorative reshaping of the worksheets and of the lesson plan.

6. Discussion

6.1. *Special Issues of Interest*

The use and re-use of preconstructed microworlds for the creation of new artifacts with a gradually increasing degree of complexity encourages the systematization of knowledge and bridges the gap between the simple and the more complex (Glezou and Grigoriadou, 2009b). It is difficult to provide students with solid, preconstructed models to help them create new ones, and, at the same time, so flexible as to encourage them to explore their inner functions (Simpson *et al.*, 2005).

The proposed educational scenario and the structured series of activities exploited in gradual steps and according to the acquired experience of the students could be considered especially effective in scaffolding simulation development, Logo programming and in parallel, gradual familiarization with MicroWorlds Pro; it may be adapted/extended to respond to special characteristics of users group, such as age, background and may be used in different learning contexts.

The Logo programming syntax constituted a major obstacle for some students as they were often discouraged by the strictness of the programming syntactic rules. This applies particularly to students who had never wished or had the chance to familiarize themselves with programming. Attempting to overcome this barrier we are intrigued to explore the use of different modeling paradigms, as Modellus and Scratch. Modellus (<http://modellus.fct.unl.pt/>) enables students and teachers (high school and college) to use mathematical equations and expressions to create or explore models interactively – no programming language is needed. By employing Scratch (<http://scratch.mit.edu/>), the users do not have to write programming code. Instead, they can drag and “snap together” colorful graphical building blocks, each of which represents a simple programming instruction. By snapping together different combinations of these blocks, students can create sequences that build up into simulations.

6.2. *Limitations of the Study*

The present study is a small-scale case study in which students of two classes of 3rd grade of Senior High School in “Multimedia-Networks” course participated. The sample size ($N = 18$ and $N = 23$ respectively) may not be sufficient to adequately support a generalization of the findings and associated implications for learning.

It is possible that the use of other preconstructed microworld or other series of activities and within another group of students or another course there might have other results/implications. Three factors are considered to be crucial in order to generalize the findings of the present study. First successive cycles of experimentations and evaluations of the instructional approaches in future courses. Second, the methods used for collecting research data should be assessed to ensure their quality and be completed with supplementary both quantitative and qualitative methods. Finally, the teaching strategies should be implemented in varied educational contexts.

6.3. Future Work

The implemented teaching strategies need to be further evaluated, regarding their effect on scaffolding simulation development, Logo programming -programming language and philosophy- and gradual familiarization with the programming environment. Further research is needed to help identifying the key aspects around how such teaching approaches are used to the best effect for students' engagement and learning. In addition, more research is needed for the identification of the special design characteristics of the learning activities by using preconstructed microworlds and the preconstructed microworlds themselves, which promote engagement, artifact and knowledge construction, "thinking about thinking", meaning negotiation and collaboration for different learners and learning contexts.

Our future research plans focus on exploring the use of different modeling paradigms, such as Scratch and Modellus, by implementing alternate instructional strategies which might lead to the most effective combinations to support student engagement in simulation development.

References

- Ackermann, E. (2003). Hidden drivers of pedagogic transactions: Teachers as clinicians and designers. In: Cnotinfor (Ed.), *Proceedings of Eurologo 2003*. Porto, Lda.
- Brouwer, N., Muller, G., Rietdijk, H. (2007). Educational designing with MicroWorlds. *Journal of Technology and Teacher Education*, 15(4), 439–462.
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A. Miller, P. (1997). Minilanguages: A way to learn programming principles. *Education and Information Technologies*, 2(1), 65–83.
- Clements, D.H., Meredith, J.S. (1993). Research on Logo: Effects and efficacy. *Journal of Computing in Childhood Education*, 4, 263–290.
- Dagiene, V. (2003). A set of Logo problems for learning algorithms. In: Cnotinfor (Ed.), *Proceedings of Eurologo 2003*, Lda. Porto, pp. 168–177.
- Dapontes, N. (2005a). How to program a free fall having Galileo's law as a starting point? Retrieved August 20, 2006 from http://www.dapontes.gr/index.php?option=com_content&task=view&id=163&Itemid=49 (in Greek).
- Dapontes, N. (2005b). How to program a stroboscopic representation of the free fall? Retrieved August 20, 2006 from http://www.dapontes.gr/index.php?option=com_content&task=view&id=164&Itemid=49 (in Greek).
- Dapontes, N., Ioannou, Σ., Mastroiannis, I., Tzimopoulos, N., Tsovolas, Σ., Alpas, A. (2003). *The Teacher as a Creator. Ideas on How to Teach MicroWorlds Pro in Kindergarten and Primary School*. Kastaniotis Publications, Athens (in Greek).
- De Jong, T., van Joolingen, W.R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research*, 68(2), 179–201.
- DePasquale, P. (2002). Subsetting language elements in novice programming environments. In: Edwards, S. (Ed.), *Proceedings of the RESOLVE Workshop 2002*. Columbus, OH, pp. 108–111.
- Dimitracopoulou, A., Komis, V. (2005). Design principles for the support of modelling and collaboration in a technology-based learning environment. *Int. J. Cont. Engineering Education and Lifelong Learning*, 15(1/2), 30–55.
- DiSessa, A. (1995). Epistemology and systems design. In: diSessa, A., Hoyles, C. (Eds.), *Computers and Exploratory Learning*, Springer Verlag, pp. 15–29.
- DiSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy*. MIT Press, Cambridge, MA.
- DuBoulay, B. (1989). Some difficulties of learning to program. In: Soloway, E., Spohrer, J.C. (Eds.), *Studying the Novice Programmer*. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 283–299.

- Ebrahimi, A. (1994). Novice programmer errors: Language constructs and plan composition. *International Journal of Human-Computer Studies*, 41, 457–480.
- Forbus, K., Carney, K., Harris, R., Sherin, B. (2001). A qualitative modeling environment for middle-school students: A progress report. Retrieved September 20, 2007 from http://www.qrg.northwestern.edu/projects/NSF/Vmodel/papers/Vmodel_QR01_Final.PDF
- Glezou, K., Grigoriadou, M. (2007). A novel didactical approach of the decision structure for novice programmers. In: Kalas, I. (Ed.), *Proceedings of 11th European Logo Conference*. Bratislava, Slovenia.
- Glezou, K., Grigoriadou, M. (2008). Simulation development by students: An alternative cross-thematic didactical approach. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008*. AACE, Chesapeake, VA, pp. 4108–4117.
- Glezou, K., Grigoriadou, M. (2009a). Supporting student engagement in simulation development. In: O'Malley, C., Suthers, D., Reimman, P., Dimitracopoulou, A. (Eds.), *Proceedings of 8th International Conference on Computer Supported Collaborative Learning CSCL2009: Computer Supported Collaborative Learning Practices*. Rhodes, pp. 414–418.
- Glezou, K., Grigoriadou, M. (2009b). Design principles of training material for introductory courses to programming and Logo by using preconstructed microworlds. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009*. AACE, Chesapeake, VA, pp. 1606–1614.
- Grigoriadou, M., Gogoulou, A., Gouli, E. (2002). Alternative teaching approaches in introductory programming lessons: Teaching suggestions. In: Dimitrakopoulou, A. (Ed.), *Proceedings of the 3rd Pan-Hellenic Conference "The Information and Communication Technologies in Education"*, Vol. A'. Rhodes, Greece, pp. 239–248 (in Greek).
- Harel, I., Papert, S. (1991). *Constructionism: Research Reports & Essays, 1985–1990 by the Epistemology & Learning Research Group*. Ablex Publishing Corporation, Norwood, US.
- Hoyle, C., Noss, R., Adamson, R. (2002). Rethinking the microworld idea. *Journal of Educational Computing Research*, 27(1/2), 29–53.
- Kafai, Y., Resnick, M. (Eds.) (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Lawrence Erlbaum Associates, Mahwah, NJ.
- Kalas, I. (2006). Discovering informatics fundamentals through interactive interfaces for learning. In: Mittermeir, R.T. (Ed.), *ISSEP 2006*. LNCS, Vol. 4226, pp. 13–24.
- Komis, V. (2005). *Introduction in Didactics of Informatics*. Kleidarithmos Publications, Athens (in Greek).
- Komis, V., Dimitrakopoulou, A., Politis, P. (1999). Activities in the field of analogy using educational modeling software. In: *Two-day Meeting: "Informatics in Education"*, Computer Instructors Association of Hepirus, Pedagogical Institute, University of Ioannina's Pedagogical Dpt of Civil Education, Ioannina, May 1999, pp. 125–138 (in Greek).
- Kynigos, C. (2006). Half-baked Logo microworlds as boundary objects in integrated design. In: Kalas, I. (Ed.), *Proceedings of 11th European Logo Conference*. Bratislava.
- LCSI, <http://www.microworlds.com/>
- Louca, L., Constantinou, C. (2002). The use of computer-based microworlds for developing modeling skills in physical science: An example from light. *International Journal of Science Education*. Retrieved October 20, 2007 from <http://www.stagecast.com/pdf/research/Modeling.pdf>
- Louca, L., Druin, A., Hammer, D., Dreher, D. (2003). Students' collaborative use of computer-based programming tools in science: A descriptive study. In: Wasson, B., Ludvigsen, St., Hoppe, U. (Eds.), *Designing for change in Networked Learning Environments: Proceedings of the CSCL 2003*. Kluwer Academic Publishers, The Netherlands, pp. 109–118.
- Modellus. <http://modellus.fct.unl.pt/>
- Pane, J., Myers, B. (1996). *Usability Issues in the Design of Novice Programming Systems*. Technical Report (CMU-CS-96-132), School of Computer Science, Carnegie Mellon University.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.
- Resnick, M., Kafai, Y., Maeda, J. et al. (2003). A networked, media-rich programming environment to enhance technological fluency at after-school centers in economically-disadvantaged communities. Retrieved October 10, 2008 from <http://www.media.mit.edu/~mres/papers/scratch.pdf>
- Rieber, L.P. (2004). Microworlds. In: Jonassen, D. (Ed.), *Handbook of Research for Educational Communications and Technology*, 2nd edition. Lawrence Erlbaum Associates, Mahwah, NJ, pp. 583–603.
- Schwartz, J.L. (2007). Models, simulations, and exploratory environments: A tentative taxonomy. In: Lesh, R.A., Hamilton, E., Kaput, J.J. (Eds.), *Foundations for the Future in Mathematics Education*. Lawrence

- Erlbaum Associates, Mahwah, NJ, pp. 161–172.
- Scratch. <http://scratch.mit.edu/>
- Simpson, G., Hoyles, C., Noss, R. (2005). Designing a programming-based approach for modelling scientific phenomena. *Journal of Computer Assisted Learning*, 21(2), 43–158.
- Soloway, E. (1986). Learning to program = Learning to construct mechanisms and explanations. *Communications of the ACM*, 29(9), 850–858.
- Spohrer, J., Soloway, E. (1986). Alternatives to construct-based programming misconceptions. In: *Proceedings of CHI '86*, pp. 183–191.
- Teodoro, V. D. (2002). *Modellus: Learning Physics with Mathematical Modeling*. PhD Thesis. Retrieved March 07, 2009 from <http://hdl.handle.net/10362/407>
- Turcsányi-Szabó, M. (1998). Designing LOGO based microworlds for effective learning – A road to improve teacher education. In: Marshall, G., Ruohonen, M. (Eds.), *Capacity Building for IT in Education in Developing Countries*. Chapman & Hall, IFIP, pp. 163–171.
- Vosniadou, S. (2005). *Planning Learning Environments Supported by Modern Technologies*. Athens, Gutenberg Publications (in Greek).
- Vygotsky, L.S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, Massachusetts.
- Wilensky, U., Reisman, K. (2006). Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing computational theories – An embodied modeling approach. *Cognition & Instruction*, 24(2), 171–209.

K. Glezou received her BA in physics, MSc and MEd from University of Athens in 1990, 1994 and 2000 respectively. Currently, she is a teacher of informatics in Arsakeia Schools in Athens, teacher trainer, and she is working towards her PhD degree at the Department of Informatics & Telecommunications of the University of Athens. Her research lies on the area of open interactive learning environments which support exploratory and collaborative learning in the context of didactics of science and didactics of informatics. The research focuses on the development of educational scenarios and microworlds and utilization in the teaching-learning process through exploratory and collaborative activities. In 2009 she created the Greek Educational Online Social Network “Logo in Education: a community of practice and learning” (<http://logogreekworld.ning.com/>).

M. Grigoriadou received her BA in Physics from University of Athens in 1968 and her D.E.A and Doctorat from the University of Paris VII in 1972 and 1975, respectively. She is now a professor in education and language technology, and the head of the Education and Language Technology Group, Department of Informatics and Telecommunications, University of Athens. Her current research interests lie in the areas of adaptive learning environments, Web-based education, ITS, educational software, natural language processing tools, and computer science education. Maria Grigoriadou has won 7 awards, has participated as project manager and senior scientist in 15 national and European Union projects, and has 4 invited talks. She has 32 publications in international journals, 11 contributions in international book chapters, 135 papers in proceedings of international conferences, and more than 600 citations to her research work. She is a member of IEEE, AACE, IADIS, EDEN, Kaleidoscope, and LeMoRe.

Vidurinės mokyklos mokinių sudominimas imitavimo kursu

Katerina GLEZOU, Maria GRIGORIADOU

Šiame straipsnyje pristatomas nedidelio masto tyrimas naudojant MicroWorlds Pro įvairialypės terpės programavimo aplinką, kaip pradinę priemonę kuriant modelius. Tyrime dalyvavo vidurinės mokyklos vyresniųjų klasių mokiniai, buvo įgyvendintas taikomasis projektas „Laisvojo kritimo imitavimo plėtra“ – atvira ir lanksti sistema, skirta veiklai klasėje. Buvo tiriamos dvi pakaitinės mokymo strategijos: a) imitavimo konstravimas pagal scenarijų ir b) iš anksto sukonstruoto mikropasaulio panaudojimas. Buvo stebima, kaip mokiniai bendradarbiauja ir sąveikauja programavimo aplinkoje. Straipsnyje aptariamas visas procesas, nagrinėjamas mokinių įtraukimo į veiklą lygis, mokinių mokymosi panašumai ir skirtumai, nurodant tam tikras programavimo aplinkos ypatybes, kurios padeda arba sukelia sunkumų kuriant veiksmingą mokymosi aplinką. Veiksmingas konstruktyvistinis požiūris buvo projektuojamas, kuriamas ir įgyvendinamas siekiant sudominti mokinius imitavimo procesu kaip įvairialypės terpės tarpdisciplininio projektu.

– 9 boys) respectively. The students were separated in small groups of 2 per computer of their own choice. Due to the odd number of the 2nd class, there was also one team (team #9 of 2nd class) with three students. Both classes had a similar programming background and similar averages of all disciplines.

We implemented two alternate instructional strategies (one for each class): a) simulation development from scratch and b) use of a preconstructed microworld.

The preconstructed microworld “Free fall simulation development” (Fig. 1) consists of four pages characterized by a gradually increasing complexity that corresponds to the particular first three phases of the scenario:

- on page 1 of the microworld we have the construction of the free fall simulation (corresponding to Phase A of scenario);
- on page 2 we have the construction of the free fall’s stroboscopic representation (corresponding to Phase B of scenario);
- on page 3 we have the construction of the free fall’s stroboscopic representation and the exploitation of a table exhibiting the values of time and position (corresponding to Phase C of scenario (Stage 4 – 7th Step, as described in lesson plan in Section 3.2);
- on page 4 we have the construction of the free fall’s stroboscopic representation and the exploitation of a table exhibiting the values of time, position and velocity (corresponding to Phase C of scenario (Stage 4–8th and 9th Step, as described in lesson plan in Section 3.2).

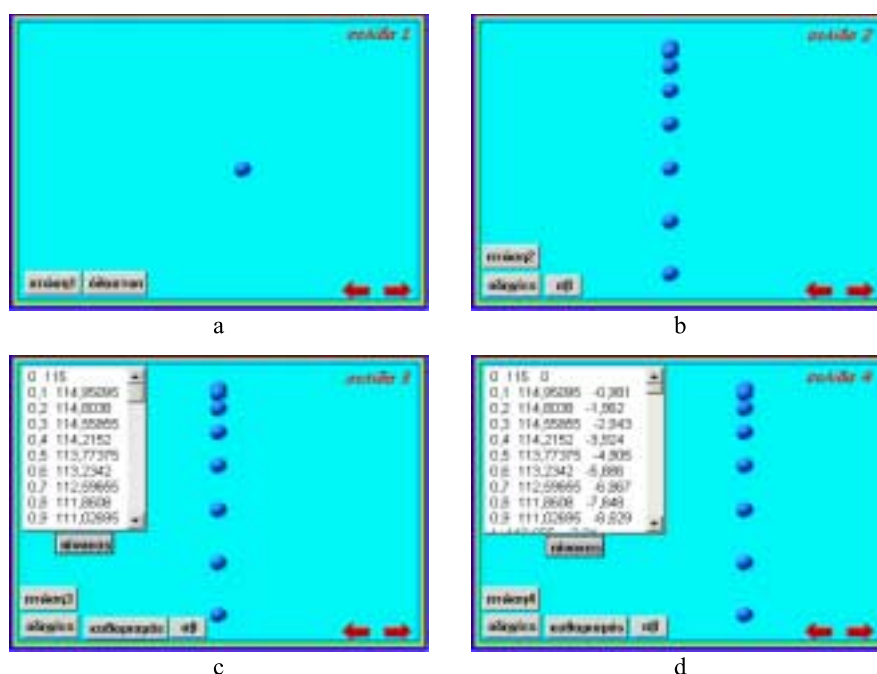


Fig. 1. Snapshots of the “Free fall simulation development” preconstructed microworld pages: a) 1st page, b) 2nd page, c) 3rd page and d) 4th page.

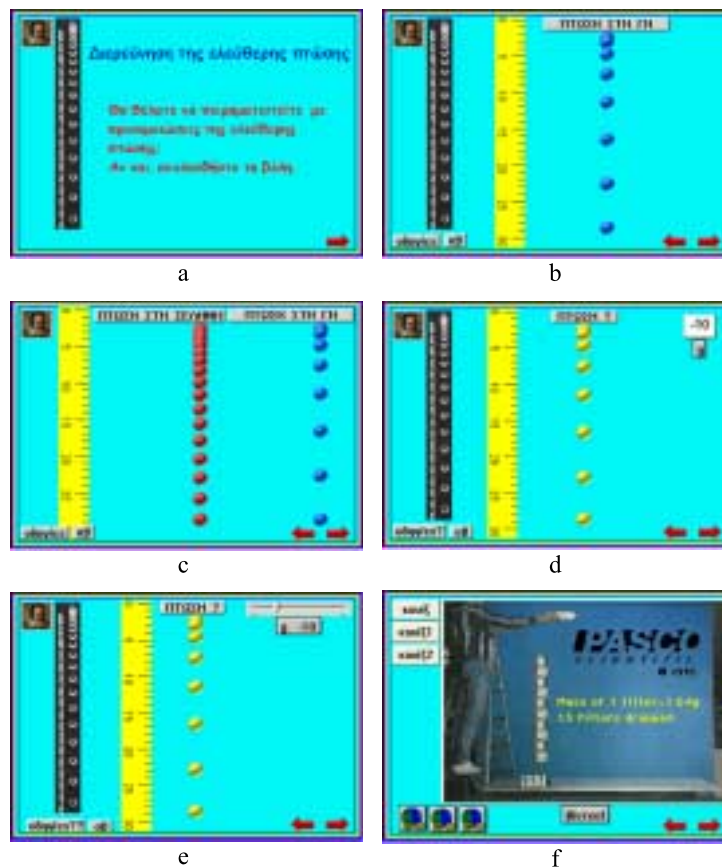


Fig. 2. Snapshots of the microworld of the students' project work (Team 3 of 2nd class).

notice enriched multiple pages with additional material that was found on the Internet and in the installation folders of softwares available (like Modellus: another computer tool for modeling and experimentation).

In Fig. 2a the students added an introductory page, calling to “Free fall investigation” and enriched with images (like an image of Galileo, an image of Free fall stroboscopic representation).

In Fig. 2b the students defined a new procedure as “Fall_on_earth”, renamed properly the corresponding button and added one turtle shaped as a ruler to measure the distance between the ball's stamps.

In Fig. 2c we see a parallel stroboscopic representation of the free fall on the earth and the moon (by clicking buttons corresponding to appropriate procedures defined as “Fall_on_earth”, “Fall_on_moon”).

In Figs. 2d and 2e, the students proceeded to the parameterization procedure. They suitably defined the parametric procedure as “Fall_?” and then used a text box (Fig. 2d) and a slider (Fig. 2e) to input the value of variable g (where g : acceleration of gravity). In



Fig. 3. Snapshots of the microworld of the students' project work (Team 8 of 2nd class).

Fig. 2f, the students added a video-experiment relating to the investigation of the free fall phenomenon (found in the installation folder of Modellus), hyperlinks to websites where relevant educational material can be found, as well as new buttons linked with procedures dealing with quizzes.

In Fig. 3, snapshots of the microworld created by another team (Team 8 of 2nd class) are presented.

In Figs. 3a and 3b we notice the introductory page enriched with images and an announcement emerging in the presentation mode of the microworld, with the message "Let's investigate the Free Fall".

In Fig. 3c, we see the turtle shaped as an apple, two different text boxes as tables to output the values of time (t) and velocity (v) as opposed to the single table used in the preconstructed microworld.

In Fig. 3d, the students gave the microworld a game dimension. As we see, an emerging announcement is calling us to observe the movement and calculate the value of g , combining the values of time (t) and velocity (v) from the text boxes.

In Figs. 3e and 3f, the students used a parametric procedure and a text box to input the value of variable g (g : acceleration of gravity).