



HAL
open science

Half-Baked Logo Microworlds as Boundary Objects in Integrated Design

Chronis Kynigos

► **To cite this version:**

Chronis Kynigos. Half-Baked Logo Microworlds as Boundary Objects in Integrated Design. *Informat-ics in Education - International Journal*, 2007, 6, No. 2, pp.335-358. hal-00588777

HAL Id: hal-00588777

<https://hal.science/hal-00588777>

Submitted on 10 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Half-Baked Logo Microworlds as Boundary Objects in Integrated Design

Chronis KYNIGOS

*Educational Technology Lab, University of Athens and RACTI
Panepistimiopolis, Ilissia, Athens 15781, Greece
e-mail: kynigos@ppp.uoa.gr*

Received: August 2007

Abstract. The paper addresses the problem of fragmentation of the communities involved in the design of digital media for education. It draws on the experience gained at the Educational Technology Lab in the design of Logo-based microworlds with three different platforms respectively based on component computing, 3D game engines and 3D navigation with a GIS. In this paper I use the term half-baked to describe a microworld which is explicitly designed to engage its users with changing it as the main aspect of their activity. I discuss this kind of microworld as a tool for integrated design involving people with diverse expertise and/or roles to communicate. These kinds of microworlds implicitly exist within the community, but they can be explicitly designed mediated and put to use in the role of facilitators for integrated design and development to enable a growing communication amongst researchers, technicians, teachers and students. A template for presenting microworlds which was constructed through the experience with four such integrated communities is used to describe for each respective case the design principles, the affordances, the histories of development and the variety of emergent microworlds.

Key words: microworlds, malleable, integrated design, boundary objects.

Introduction

I use the term “half-baked” microworlds to describe digital media designed to facilitate communication between researchers, technicians, teachers and students as they become engaged in changing them. Microworlds have been the main Logo-based vehicles through which the key ideas of generation of meanings through communicational and constructionist activity have been mediated within the field of instructional design (Goldenberg, 1999). After 40 years, we have a lot of experience of how difficult it is to convey such meanings in education and more so, to have hopes for a progressive infusion of such practices in formal schooling (Kynigos, 2004). The demands for re-addressing epistemologies and personal pedagogies, changing roles and communicating about such changes, let alone the sheer logistics within the socio-systemic educational context, have been too big to see some serious scaling-up of Logo-like educational practices. At the same time, the production of microworlds and escorting materials has not scaled up either. It has been hard to convey the new kinds of pedagogy involved as well as the kinds

of interaction with the technology envisaged in such educational environments. The process of designing educationally principled materials for microworlds requires integrated expertise which is hard to find. Many pieces of educational software and corresponding materials at large look more like products of fragmented views of the problem, emerging from traditional curriculum designers who miss the opportunity for added educational value, from technicians in the software industry perceiving design for education as just another field of application of generalized design processes, researchers who have little understanding of the pragmatics of education, students who inadvertently perceive their role as requiring memorization and response to tests.

The microworld community has a lot to offer to the task of integrating expertise in the design of educational software and addressing innovation in educational systems. What is needed is to develop a much more articulate language to convey the idea of design for reflection and for the generation of integrated expertise. Out of the communities mentioned, only technicians and maybe researchers perceive themselves as designers. Yet, design can be an activity which is tightly linked to constructionism for learners and to reflexive practice for teachers. Communal design can generate the need to be explicit, to reflect and to express meanings through argumentation. During a collaborative activity, a community works towards a common goal, which can be an ideal object to be created, or a specific /improvable/ object (Bereiter and Scardamalia, 2003). This object is both the centre of the activity, and also functions as a communicational tool to shape a common language within the community. Cobb *et al.* (2003) extend this notion by proposing the term "boundary objects", e.g., specific objects within different communities, which are «/relatively transparent means of conveying meaning among the members of the community who created them/». They can also be the centre around which community members organize their activities and can additionally operate as tools for communication among the members of the same community, and the members of other communities. Half-baked microworlds are meant to be "improvable boundary objects" and the process of changing them can be orchestrated to enhance integration in microworld design.

In recent years, analyses of the use of digital media have given rise to theories of use such as Rabardel's cognitive ergonomics theory of instrumentation (2001), i.e., the process by which a digital artifact (the object) becomes a mental scheme for each individual as they use it thought a process which he termed 'instrumentation'. Guin and Trouche (1999) discussed this theory further suggesting that we need a different term for the process by which instrumentation involves changes made to the artifact itself – 'instrumentalisation'. However, this theory, which is proving relevant at least within the field of mathematics education, seems to adopt a perspective that instrumentation and instrumentalisation are phenomena which inevitably happen when digital media are used. A particular characteristic of many Logo-based artifacts (to use Rabardel's term) is that they have been designed in order to be changed, i.e., designed for instrumentalisation. This characteristic however, in many cases remains implicit and is not really made use of in practice.

Half-baked microworlds are pieces of software explicitly designed so that their users would want to build on them, change them or de-compose parts of them in order to construct an artifact for themselves or one designed for instrumentation by others. They are

meant to operate as starting points, as idea generators and as resources for building or decomposing pieces of software. This constructionist activity is seen as part of inquiry and argumentation leading to the generation of mathematical and scientific meaning. At the same time, half-baked microworlds can be designed for teachers to engage in pedagogical or epistemological reflection as they de-construct or re-construct the microworlds in a context of designing tools for students. Half-baked microworlds operate like diSessa's toolsets (diSessa, 1997) in that they are not built and presented as ready-made environments to be understood by the teachers and then used by students. Instead, the point is to change and customize them and thus to gain ownership of the techniques and the ideas behind microworld construction as outlined earlier.

Expressing meanings by changing the functionalities afforded by microworlds is something which has been happening through the years in the relatively small world of Logo friends, the microworlds originally designed or used by one or more of these communities. However, they have not frequently been explicitly conveyed as media to be questioned and changed in themselves, but rather as tools to explore and construct with and possibly to extend. Moreover, the questioning of microworld functionalities and interface has seldom been perceived as an activity for people from such communities to interact with each other in order to acquire some shared perception of concepts, perspectives and goals.

The Evolution of Perceptions of Microworlds

Although the term 'microworld' was born with the advent of Logo a long time ago, perceptions of its substance and its purpose and functionality have evolved quite drastically over the years. The term was originally borrowed by Seymour Papert from artificial intelligence (A.I.). Its meaning evolved through the years within the mathematics and science education community. Papert described it as a self-contained world where students can "learn to transfer habits of exploration from their personal lives to the formal domain of scientific construction" (Papert, 1980, p. 177). A more recent description of micro-worlds is that they are computational environments embedding a coherent set of scientific concepts and relations designed so that with an appropriate set of tasks and pedagogy, students can engage in exploration and construction activity rich in the generation of meaning. The first example of a microworld was 'Turtle Geometry' within the Logo programming language which generated a bulk of research on the construction of children's geometrical meanings (Laborde *et al.*, 2006). Later research however, involved learning with the use of micro-worlds embedding a much narrower set of mathematical, scientific and other concepts, escorted by more focused theories on learning and pedagogy (Noss and Hoyles 1996; Edwards 1995; Clements and Sarama 1997; Sarama and Clements 2002).

In the early days, microworlds were seen as strictly programmable environments and their main characteristic was the ability to construct graphical models using a programming language. The semantics and rules of the language as well as the graphical representations were integrated with the concepts and relations of the domain characterizing

the microworld. The key features of the students' activities enabled with such media were the ability to make constructions and change and extend the rules and relationships of the microworld itself. The highly editable nature of these media and their executable representations providing immediate and epistemologically succinct feedback (they can be characterized as conceptual mirrors) enabled exploration and bricolage and the kind of learning which took place was characterized by Papert and his team as a special kind of constructivist learning which they termed 'constructionism' (Kafai and Resnick, 1996). Seen in a wider context, microworlds were examples of the idea of 'deep structural access' (diSessa, 2000; Kynigos, 2004; 1995) to technologies for non-technical people, allowing for creativity, customization and personal construction of technological tools. This idea was used as an argument for the use of technologies to facilitate the growth of a technological culture rather than to inhibit creativity and expressivity with technologies.

In the field of education, with the advent of new interface technologies and the broadening of the domains over which microworlds were designed including that of science, programmability and constructionism faded in favor of dynamic manipulation of representations and higher quality graphical and iconic representations offered by multimedia and enabled by a suite of authoring systems. In science the microworld rhetoric quickly gave place to that concerning simulations. One key issue was the primary importance given to the accuracy and detail of the behaviors and relationships between objects in the respective simulation. This was based on epistemologies primarily focused on the phenomenological nature of the domain rather than on meaning construction. Another issue was the extent to which the simulation represented objects as realistically as possible or provided an abstraction from the objects' shape and form (as in Interactive possible or provided an abstraction from the objects' shape and form (e.g., as in Interactive Physics). A third was the ways in which the simulation was mediated to students and the way they themselves perceived it as a real infallible phenomenon in itself or as a human-made simulation consisting of an inevitable approximation of the objects and the simulated phenomena.

In mathematics on the other hand, the problem lies with the epistemological nature of the domain itself. Mathematics can be seen as an abstract domain of human thought, its objects being axioms, relations and mathematical proof. On the other hand it can also be seen as an applied science explaining various phenomena (economic, scientific, geometric, navigational, etc). Microworlds were seen to make mathematical representations useable and learnable and thus gave rise to some questioning about the learnability of traditional mathematical representations, such as formalization, providing alternative representations designed to facilitate meaning generation (Hoyles *et al.*, 2002).

In both cases of science and mathematics education, however, programmability and constructionism re-appeared recently with the advent of technologies allowing for programmable simulation-multimedia style tools such as E-slate microworlds (Kynigos, 2004; Penner, 2001), Boxer microworlds (diSessa, 1997), Micoworlds Pro and Imagine Logo. These kinds of microworlds constitute simulations which can be used at different levels by teachers and students alike. They can be used as phenomenologically authentic simulations providing iconic interfaces to change parameters within worlds with unchangeable rules and at the same time, as questionable worlds where rules can be changed

and tampered with to generate inquiry of what it would be like if they were different. So, the issue in question is not 'when is a microworld half-baked' and 'when is it complete'. Any respectable microworld in that sense is half-baked otherwise it becomes doubtful if it is a microworld at all. In this paper I use the term half-baked to describe a microworld which is explicitly designed to engage its users with changing it as the main aspect of their activity. Furthermore, it is necessary for this kind of microworld to be conveyed as half-baked to its users. In this paper I discuss this kind of microworld as a tool for people with diverse expertise and/or roles to communicate.

Microworlds as Tools for Pedagogical Innovation in Schools

Engagement with working with half-baked microworlds involves the assumption that the overall goal is to design a learning environment which has good chances of providing some added educational value and which at the same time will be based on the use of these digital artifacts. However, both design for innovation and use of digital technologies have posed difficult challenges for education. These challenges have changed in essence during the past 40 years but not in magnitude.

The microworld idea was first coined in the late seventies by Papert. In his argumentation he presented this kind of technology as a medium to challenge and bring about re-thinking of the educational paradigm of schooling and the epistemological approach of content domains such as mathematics. This rationale came about in the historical context of some bold attempts for curricular changes in the United States based on the problem-solving movement which focused on problem-solving methods rather than the understanding of mathematical concepts in themselves. Microworlds were seen as the ideal technological boost to the movement. Inevitably, they thus became victims of the educational innovation pendulum (Agalianos, 1997; Noss, 1992). They also became victims of the early stage in which they were created with respect to the spread of technology in the culture (Papert, 2002). Technologies were very hard to access, machines were slow and non-dependable, the internet had not arrived and people were not widely using digital technologies for something else. However, microworlds continued to be designed and used in design research initiatives providing evidence that when placed in the role of tools within a carefully designed and supported educational environment, they could greatly facilitate the generation of mathematical and scientific meanings in students. Furthermore, they appeared in contexts different to the one they originated from in countries such as Mexico, Costa Rica, Greece and others (Blikstein and Cavallo, 2002; Kynigos, 2002). Recently there seems to be growing indication of a comeback of microworlds in a different role, that of digital artifacts for argumentation and inquiry learning in a digitally rich culture, which by itself will provide an external challenge to the schooling paradigm (Papert, 2002). At the moment, however, that culture is lagging greatly in integrating the use of digital media and reciprocally in evolving with respect to the traditional schoolish research paradigm. One of the reasons is that digital media are designed and developed by fragmented communities.

Microworlds for Scientific and Mathematical Thinking with

The term ‘microworld’ has been used in the Educational Technology Lab to describe an explorable and malleable digital environment characterized by an embedded knowledge domain, acting as a coherent world, inside which a student can explore alternatives, test hypotheses, and discover facts that are true for this world. The basic characteristic of a microworld is thus that it is designed as a rule-governed environment which is accessible for manipulation and exploration by the learner. The kind of manipulation may vary according to the subject domain and the decisions of the designer. The technological affordances which we used and emphasized at the Lab were such so that a microworld:

- constitutes an exploration space;
- mediates between informal and formal;
- provides executable representations;
- offers dynamic manipulation;
- evokes interplay between private and public expression;
- generates interdependent representations;
- allows deep structural access to its functionality by means of a programming language.

Microworlds constructed in this way provide powerful sites for inquiry based learning. They offer to the student opportunities for construction, exploration, observation, reflection and collaboration (see, for instance, Noss and Hoyles, 1996). When, on the other hand, we consider how microworlds are actually used and function as learning environments, we may describe them as a digital environment where the learner can

- manipulate the objects dynamically in order to discover their properties and understand the laws which dominate their behavior;
- make hypotheses and use the feedback from the environment to test them, in order to correct his or her own ideas about the domain modeled by tinkering with and debugging the microworld and
- use programming as a rigorous language to understand and change properties and behaviors of the objects.

Within learning environments based on the use of microworlds the learner is not told about a mathematical or scientific concept; on the contrary he/she is supposed to generate meanings about the concept through interaction with the microworld by playing a game or solving a problem.

Although there seems to be a distinction between programmable-constructionist and simulation based microworlds, the ETL group has adopted an integrated approach providing the first type of affordances to simulation-like tools. The emphasis was to design for the potential for users to change the rules and the laws that dominate the microworlds themselves. In that way, when the didactical design and time constraints allow, students can question the rules and the concepts underlying those rules, e.g., by asking ‘what if’ questions and being able to test them. This notion is hard to convey to technicians and teachers who perceive programming to be a black-box infrastructure to some GUI and digital artifacts to be valid unquestionable pieces of curriculum material respectively.

Microworlds as Tools for Integrated Design

The ETL group has been engaged over the years in the collaborative design of microworlds to be used as constructionist tools for inquiry and argumentation. This collaboration has involved all four communities of researchers, technicians, teachers and students. This is very hard to do, especially amongst actors with different expertise, role and views about the task working in a variety of contexts.

Microworlds have been perceived by the community at large as tools designed by researchers for radical innovation to be implemented in small scale situations. The era when this approach was valid and seductive to some communities outside our own seems to be ending. There is now demand for large scale initiatives and accreditation of new efforts before they have had the chance to become infused in educational curricula. The ideas behind the microworld culture are proving hard to grasp and accept not only by school systems but also by other stakeholders in education such as new computer science and telecommunication communities. This poses new challenges to the microworld community of finding methods and avenues for communicating these ideas to a wider audience in a language which is widely understood.

This kind of language can only emerge from experience of collaborating with such communities, it is not something which can be defined at a theoretical level. The process will require being explicit about what happens when collaborative design and implementation is taking place. In this paper, we have an example of such an effort where different kinds of communities tried to find a common language to describe microworlds. This language may seem well understood and not very consequential to our own community. What's important however is that it seemed to be understood by others. Half-baked microworlds are tools designed to facilitate the production of such a language. They are *improvable boundary objects* in that they afford changes which operate as an ice-breaker between different communities. In discussing and implementing such changes people with different expertise, role and background negotiate and make themselves explicit. The results don't really matter, they can be one jointly agreed upon microworld or each community can go their own way and present alternatives. The importance is that in doing so, there is better understanding between them which often leads to the creation of hybrid actors (Kynigos, 2002), i.e., people with one expertise who understand enough about another so as to make sense and use it for joint design. In the next section, I describe our own platforms for developing microworlds which were designed to address wider communities than our own. Subsequently, I provide an analytical description of a half baked microworld explaining how it was taken up and reformed through integrated design.

Some Alternative Platforms for Developing Logo-Based Microworlds

The microworlds that have been developed at ETL have so far been based on three platforms for microworld development: e-Slate, MachineLab and Cruiser. The main characteristics of the platforms are presented below.

The E-Slate Platform

E-Slate is an authoring environment which looks like a construction kit for creating exploratory software. The ‘bricks’ of this kit are software components of a large technical and functional variety ranging from a relational data-base to a slider or to a Newtonian simulator scene. The ‘glue’ to put bricks together comes in the form of pre-fabricated links which look like puzzle bits for users to stick together. However, there is a Logo language which plays the role of programmable ways of connecting components and of determining various functionalities within these. Our aim was to meet non-technical people half-way, i.e., to provide them with interesting black-box functionalities but also with the potential for them to easily create component configurations or custom ways in which they communicate with each other. In this way non-experts can become software developers without being involved with data-models, algorithms, and low-level programming. Software construction resembles assembling a “kit” rather than “writing a program”.

The MachineLab Platform

Machine-Lab is an authoring environment and test-bed for the creation and exploration of interactive virtual reality simulations (VR-microworlds). The representational infrastructure provided by Machine-Lab concerns controlling and measuring the behaviors of objects and entities in virtual environments simulating real three-dimensional phenomena and spaces. The design of MachineLab includes various kinds of representation of three dimensional worlds and objects enhanced by interactive and dynamic display affordances for the user.

MachineLab is implemented over the 3impact game engine (www.3impact.com) which wraps the 3D graphics of the Direct3D engine and the distributed/multi-user capabilities of the DirectPlay engine, together with a 3D sound engine and the Open Dynamics Engine (www.ode.org), which is a Rigid-Body Simulation engine with realistic physics support for ‘collidable’ objects and articulated bodies (rag-dolls / humanoids). There is a Logo language available to develop microworlds providing multiple 3D external representations of some aspect of the physical world. The platform offers the possibility of multiple observations from different point of reference (cameras). The user may adjust a virtual camera on any object, moving or still, and observe the phenomena using multiple system of reference.

The Cruiser Platform

Cruiser is a geographically oriented information system created by a spin-off company called ‘Talent S.A.’. It resembles GoogleEarth-like technology providing geo-coded data on 3D maps. The main representation of its interface is a geographical and/or cartographical three-dimensional space, supporting dynamic navigation. However it is particularly interesting for the development of educational/mathematical games since it supports:

- a higher level of personalization features, such as custom channels with structured designer-defined access rather than one global map-for-all (a ‘your data’ kind of Google-Earth);
- educationally rich capabilities such as the mediation of geographical data, including location signals in real-time.

Originally it was not meant to be a platform for Logo-like microworlds. However, a Logo language was developed and ported onto Cruiser. In this paper I describe how this provided the possibility to create a microworld for programming airplane avatar flights which we called 'Cruislet'.

Some Negotiated Descriptions of Microworlds

At the ETL we have been engaged in a number of multi-organisational projects where the development of microworlds has been a significant part of the project activity. Typically, their design and development has been the result of collaborative activity between teams with different expertise corresponding to the four communities mentioned above. Each community inevitably had its own agenda both with respect to the point of taking part in such activity and with respect to what they could do with the products. These agendas had to be communicated amongst teams and objectives had to be articulated at the level of abstraction which satisfied all parties (see, Kynigos, 2002). Also, putting together all these kinds of expertise proved a very difficult labor intensive task involving the gradual growth of a common language and understanding of what each party could bring to the joint work.

We designed and put to use all three platforms as facilitators for the integrated design process. We've had the most experience by far with E-slate which is being developed since the mid-nineties. However, both MachineLab and Cruislet allow the creation of a variety of 3D microworlds based on the use of Logo programming. The use of MachineLab involves the creation or use of 3D assets and the provision of programmable properties and behaviors to each one. This is not necessarily so with MaLT, the MachineLab-based 3D Turtleworlds software which is focused on only one such asset, the turtle. The use of Cruislet takes the underlying 3D geographical representation as a black-box infrastructure. Microworlds so far have consisted of different configurations of airplane avatars and their individual or dependent behaviors. All three platforms are based on the idea of 'black and white box' design (Kynigos, 2004). That is, instead of having a generic platform where everything can be developed through the use of a generic language (as is the case in Boxer or Logo based environments), there is a combined one, where some selected functionalities or objects are given as black box entities. This enables the user to start design and development with some technically complicated objects as a given (components in E-slate, 3D assets in Machinelab, 3D navigation in a GIS in Cruislet). It is not as conceptually powerful as generic designs but it meets users half way since it allows them to create impressive software rather quickly.

Over the years we have tried different models of collaboration during the integrated design process. One model is 'the two big machines' where a large group of developers work with a group of researchers linked with a group of teachers. The interaction between the two groups is done through one or two hybrid actors. This model has the advantage that a lot of effective work gets done, but the problem that real integration can be rather thin (diSessa, 2004). Another model is that of a researcher, a teacher and a developer

working closely together. Much better integrated work is achieved but development has to begin at a much higher level and developers need to have the motivation to delve into the specifics of the domain and the functionalities of objects and relationships. Conversely, educational researchers need motivation to create the microworld and find some added value for its use.

Half-baked microworlds have either been designed as such from scratch by the researchers at ETL or have emerged in the process of microworld development. That is, microworlds which began as projects to construct ready-for-use digital media turned into objects for re-negotiation and design of new microworlds. In the next session, I describe four microworlds, two of the former kind and another two of the latter. The focus here is on the way they are described for which I'm using a common template. This template emerged as a reasonable way to describe microworlds so that they are understood by all communities concerned and was a result of many discussions aiming to make some sense between the groups. I've added two fields at the end, 'procedure of integrated development' and 'emergent uses of the microworld' to describe the context and the itinerary of design which began from the generic half-baked microworld.

The Juggler

Represented Phenomenon

The subject domain for this microworld is velocity and acceleration in a Newtonian space. A 'Juggler' with two bats tries to keep a green and a red ball in the air. Only one bat can be moved at a time. The initial velocities and field force are set by means of dynamically manipulating respective vector representations. Once the game starts, the velocity vectors become measures of velocity and thus change continually. The field force vector of course does not change once the game has started. The force can be set to the magnitude and direction of gravitational pull, but the student is free to set its direction and length to any parameters.

Level of Abstraction

The representation of the entities is abstract, in the sense that the objects are represented as geometric figures. On the other hand since the shape of the actual objects, balls and bats, is very close to the specific figures used for the representation, we may say that besides its abstraction, the representation has some realistic characteristics. The representation of the velocities and field force however is abstract and in the form of a vector which is manipulable prior to starting the game. Also, some specific conventions made with respect to the simulated physics (such as the collision rules and the angle of the bats) give a sense of abstraction to the game.

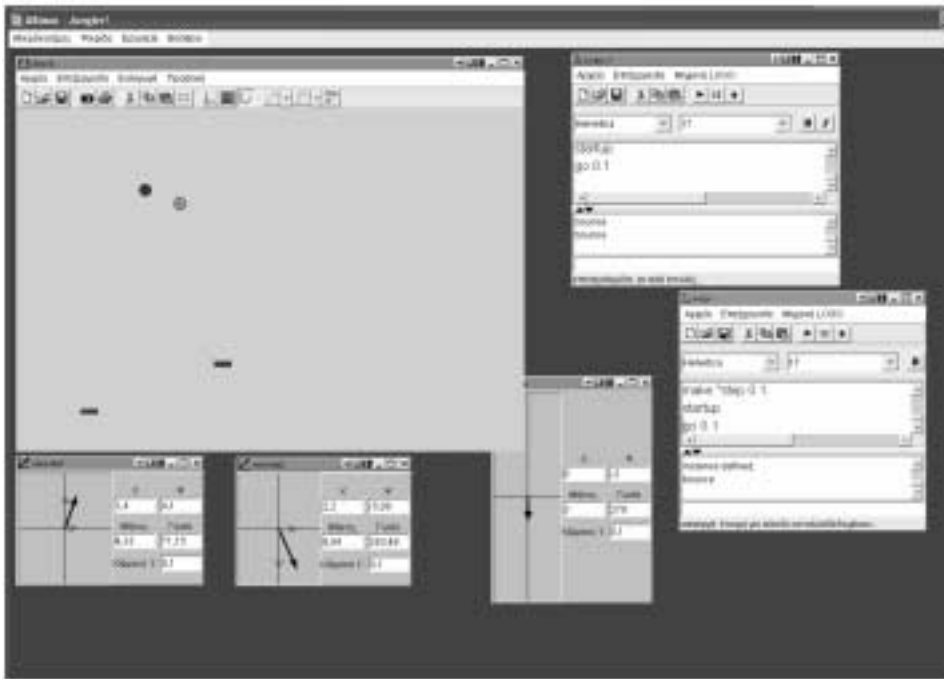


Fig. 1. The 'Juggler' microworld.

Concepts Embedded

Concepts at the kernel of the microworld are velocity, acceleration, gravity, collision. The microworld's 'conceptual field' (Vergaud, 1987) however, also consists of functional relations (time vs point on plane), change and rate of change, vectors and vectorial entities, angles (in collision rules).

User Interface

Description of the Scene

The scene contains the two balls and bats. It is possible through the programming interface to add or take away balls and bats. There are the three manipulable vector representations. Also there are two instances of the Logo code, one for each ball, and the game starts by executing the start commands.

Manipulation

The velocity vectors operate as controls before the game starts and as measures after it has begun. The field force vector operates in the same way, however, as a measure it remains constant. Each bat needs to be selected and then can be dragged anywhere on the field before or during the game. Only one bat at a time can be dragged.

Programmability

This microworld is designed for questioning the respective concepts through changes in the code of the simulation which for each ball is about one page long. The students can make changes to the velocity and field force equations, can change the rules of collision and the properties of the balls and the bats (size, mass, shape, color etc). They can also add game features like scoring, ten best scores etc.

Feedback from the Computer

When Changing the Parameters

Parameters can be changed either by changing values in the programming code or by manipulating the vectors before the game starts. Feedback is given by means of the behavior of the bats and balls during the game. The user needs to act in time to keep the balls in the air. In doing so, she/he develops strategies to anticipate the balls' trajectory based on the concepts and relations at hand.

As Result from the Experimentation

The results of the experimentation are open and related to the ways in which the game is setup either through the programming code or through the manipulation of the vectors and bats before the game. In a sense there is continuous results given by the game since it only stops when the user presses the stop button on the programming components. There is no 'success-failure' element to the game, when the bat misses the ball, the latter continues to travel 'out of site' and its velocity can only be seen by means of the numerical measure on the vector component. The user can stop the experiment and start again.

Designed Learning Process

The simulation provides the learner with a tool for investigating and questioning the scientific laws and underlying mathematical relationships in Newtonian space. When the game is setup so that these laws apply, the user can play the game and get a feel for the behavior of a particle (ball) in motion in Newtonian space. By using the vector representations they can at the same time get a feel for the essence of vectors as mathematical objects and as measures of certain entities such as velocity. However, and perhaps most importantly, the user can question Newtonian laws and make up their own ones by changing the rules of the game. Conversely, the game can be given to students with non-Newtonian rules and they can be given the task to convert it.

Didactical Design

The microworld is designed as a game to be played, questioned, changed and designed for others to play. There is a variety of didactical designs which can be applied, from task setting to more informal projects of devising games and making rules explicit. The key to the whole process is frequent reflection, argumentation and articulation of generated meanings, rules and hypotheses.

Type of Representations Used

The Juggler microworld combines phenomenological representations on the Scene with mathematical representations through the vector interface and programming code. The game can be played exclusively by means of manipulating the vectors or by an integrated approach including changes in the code of the simulation.

Procedure of Development

The Juggler was developed by researchers as a half-baked microworld. One emergent microworld, the Balloon (see below), was built by a science teacher as designer-in-training and a Logo literate mathematics teacher doing the code helped by a developer on demand ('SEED' project). Another (called 'Marbles') was done initially as specifications given by a team at the University of Neuchatel ('ESCALATE' project) and subsequent realization that very close co-design between the team, a Logo literate science teacher and a developer was necessary. Finally, another one was done by high-school students helped by their computer science teacher simulating a custom arcanoid game.

Emergent Microworlds

The original Juggler was used in a mathematics student dissertation to teach functions and in a case study to teach the meaning and functionality of vectors. It was also picked up by a the science teacher within the 'SEED' project who built a simulation representing relative velocities of a balloon and a particle thrown at it. The balloon microworld included graphs and sliders. The programmability and gaming features were meant to be less apparent to students. For the 'ESCALATE' project, the Neuchatel team had a specific agenda to build a simulation of a Piagetian experiment involving gravitational pull, friction and collision in a system of a marble sliding down a slope to hit another two onto an upward slope. Sliders were used for a range of parameters. Finally it was used as a starting point by high-school students in a computer club to develop an 'arcanoid' game. The students took away the Newtonian rules and focused on adding new custom features to the arcanoid such as user placement of the bricks and free movement of the bat (Kynigos *et al.*, 2006).

The Right Triangle

Represented Phenomenon

A geometrical figure is created by means of a Logo procedure with two variables :x and :a by using the Turtleworlds software. For some pairs of values the figure becomes a right triangle, for others it is just a jagged line. When it is a triangle :x stands for an internal acute angle and :a for one of its sides. The other side is created by relating :x and :a with the sin function. Students are asked to explore the kind of relationship between

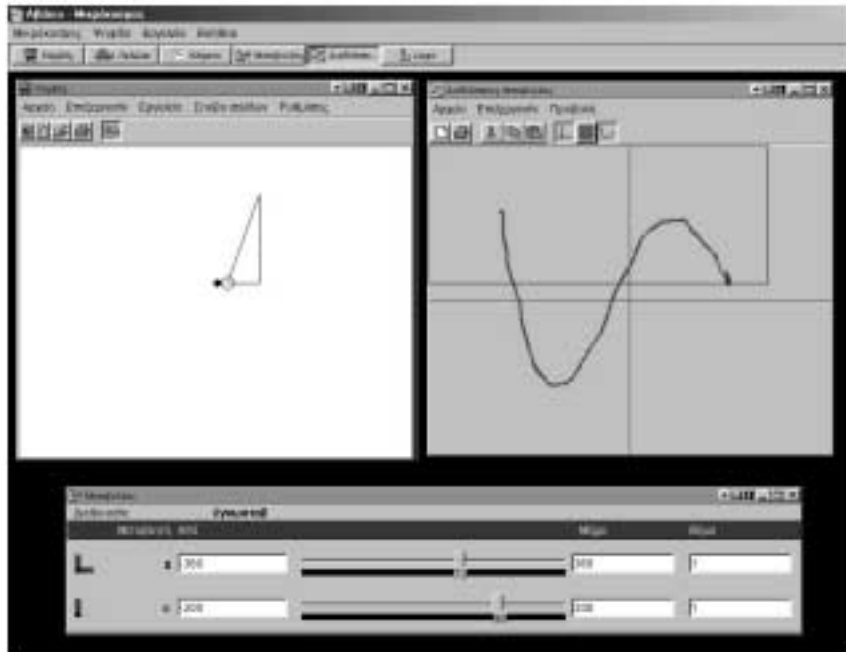


Fig. 2. The 'Right Triangle' microworld.

:a and :x so that the figure always becomes a triangle. They are encouraged to use the uni-dimensional and bi-dimensional variation tools of the software to do this.

Level of Abstraction

Within the context of mathematical epistemology, this representation is rather concrete since it provides a graphical output of an instance of the abstraction signified by the variable procedure.

Concepts Embedded

Sinusoidal functional relationships within a right triangle and of course other corollary concepts such as angle, function, variable.

User Interface

Description of the Scene

It's a Turtle Geometry setup with editor and graphics windows. On top of that however, there are a uni and a bi – dimensional variation tool. Once a variable procedure has been executed the user can click on any part of the turtle trace to activate the former tool which understands which is the underlying procedure and the number of variables. It provides a slider for each variable. When the slider is dragged the figure changes Cabri-style

in connection with the changes in variable values. The bi-dimensional tool is connected to the uni-dimensional one, provides a plane where free dragging produces co-variation between two variables and Cabri-style movement of the figure.

Manipulation

Apart from the usual Turtle Geometry functionalities, the user can drag the variation tools, edit their range and step of change. They are supposed to go back and forth from editing the code to trying out how the figure changes with the variation tools.

Programmability

This microworld is based on a program given to the students from the beginning. The students are told that the program is buggy and they are in the role of fixing it by means of experimenting what it does and trying out changes to the code.

Feedback from the Computer

When Changing the Parameters

Parameters in this case are the procedure variables. Feedback is in two forms, one is a direct forming of the figure when a procedure with specific variable values is executed. The other is in Cabri-style figural changes when either the uni- or bi- dimensional variation tools are used. There is also the usual Logo textual feedback.

As Result from the Experimentation

It's very hard to hit two values which create a triangle without the variation tool. By using the uni-dimensional one, students reach specific standalone pairs of values but it's still very difficult to make a sense of what kind of relationship there is between them. By using the bi-dimensional one, dragging the mouse on the plane leaves a sinusoidal trace once the students get the hang of moving it so that it always makes a triangle.

Learning Process

Students are of course not given a hint about the necessary relationship. The point is for them to integrate the use of all three representations, text, figure and variation tool to experiment, explore and express some theorems in action. The trace on the bi-dimensional tool provides a hint as to the nature of the relationship. It is then up to didactical engineering and the level – expertise of the students to move on to working out the solution and expressing it by means of de-bugging the procedure so that it always creates a right triangle. The students can subsequently put the theorem to use by creating shapes and animations based on right triangle shapes.

Didactical Design

A lot depends on the experience and expertise of the students of course. The main didactical agenda could be focused on the process of doing mathematical exploration and on the support for engagement with argumentation over the concepts of sinusoidal functions periodicity, variables. The activity is open to a range of social orchestrations from one on one to collaborative pairs, large group debates, presence or distance argumentation and exchange of ideas and constructs.

Type of Representations Used

The representations are mathematical and thus address abstract concepts and relations. However, the figural representation constitutes an instance of a generalized figure and the segments are created to signify the concept of Euclidean line (i.e., without substance). The figures can be animated through the variation tools providing a sense of the ways in which they can change.

Procedure of Development

The right triangle microworld was designed in the Educational Technology Lab. It became a boundary object between researchers and teachers. Its development involved very simple Logo code only, so the code itself became the object of negotiations around engineering, constructionist mathematics and the accompanying pedagogies.

Emergent Microworlds

This microworld was used in the context of in-service teacher education courses on the use of technology for mathematics education (Kynigos, 2007). An equivalent microworld was built as a result of debate and reflection of these teachers, consisting of a procedure for an arc and its relationship to the corresponding cord of the respective circle. Furthermore, the microworld was used as a starting point of a PhD theses. One was on the concept of periodicity of trigonometric functions, where the right triangle was transformed into the shape of a clown (Kynigos and Gavrilis, 2006). The other involved a buggy simulation of an ancient quadrant in order to focus on the tangent function (Keisoglou and Kynigos, 2006).

Heron's Automata*Represented Phenomenon*

An ancient Greek temple is the central feature of the microworld. A priest uses torches to ignite Heron's technology, an underground boiler creating the energy to open the temple gates. He can use one to six torches enhancing the angle at which the gate opens. The user can either choose the number of torches or navigate around and inside the temple with six arrow and letter keys, one for each direction on the three axes of the 3D space. The user can also navigate underground to observe a representation of the boiler technology. The microworld is constructed in MachineLab.

Level of Abstraction

The representation is concrete since there are no visible mathematical representations. However, navigation of the camera in the three dimensions is done through abstract means, i.e., arrow keys plus a couple of other designated keys.

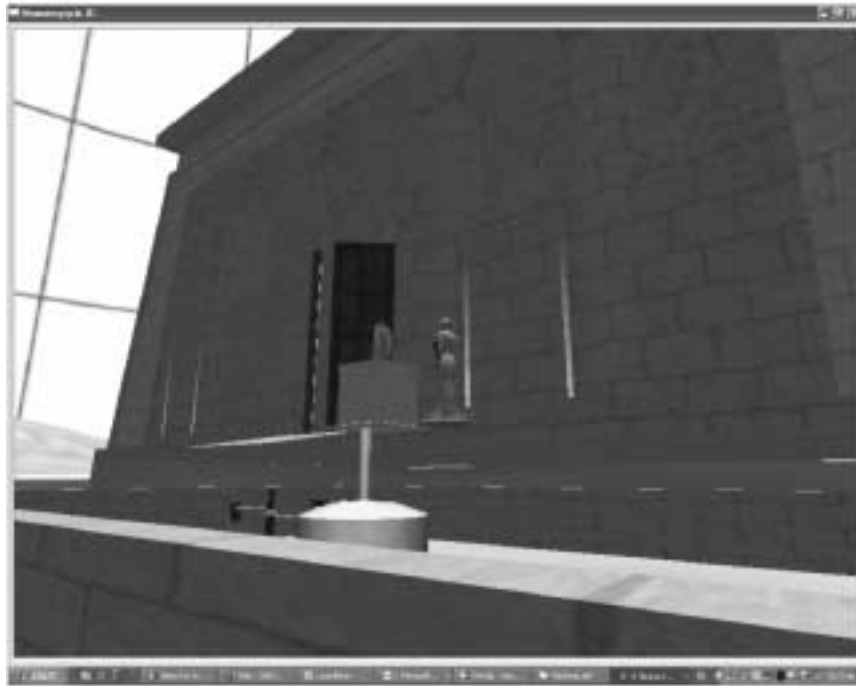


Fig. 3. 'Heron's Automata'.

Concepts Embedded

The concepts which were primarily meant to be embedded were to do with the functioning of Heron's technology and the history of the times.

User Interface

Description of the Scene

The scene consists of the temple and the priest in front of the gate. The only thing that can be put to motion out of the objects on the scene is the gate. The user can navigate below the ground to observe the technology for opening the gate.

Manipulation

Only two things can be manipulated. The positioning of the camera which can be moved in a continuous sense through the navigation keys and the number of torches used by the priest which requires a selection of an equivalent number of keys.

Programmability

This microworld was not originally meant to leave programmability open to students or teachers. Logo was used to program the basic functionalities of navigation and selection of number of torches. This was done so that a Logo literate teacher could communicate with a historian and a developer. The historian was not aware of the learning

potential of interactivity before engagement with this project. However, a study was carried out with students who used the Logo code to set the camera at specific points in space experimenting with the inputs of Cartesian coordinates.

Feedback from the Computer

When Changing the Parameters

Parameters in this case were the number of torches to open the gates. Feedback is in the form of the gates slowly opening at respectively different angles. Also, moving the camera provides a sense of navigation around and inside the temple.

As Result from the Experimentation

There is very limited scope for experiments with the number of torches. However, students unexpectedly found navigation fascinating, trying to figure out and control moving the camera.

Learning Process

The design of the learning process turned out to be an issue for debate and communication between the historian (not history educationalist) and the teacher – mathematics educator. What was important to the former was the historical accuracy of the phenomenon and the temple itself. A lot of effort was spent on getting the 3D asset right according to historical sources. The mathematician on the other hand was interested on issues of mathematical relationships between the magnitude of heat and the energy created by the boiler. In the end he felt that the ground for mathematical exploration was minimal.

Didactical Design

Again there were two visions of didactical design. One involved a revelatory approach to learning and an unquestionable role for the technology. The other a facilitator role for the teacher with students carrying out experimentation and possibly questioning the relationships between measure of heat and energy.

Type of Representations Used

The representations were in the form of a realistic simulation of a phenomenon. There were no mathematical representations apart from the underlying Logo code mainly usable with respect to the navigation of the camera.

Procedure of Development

The development involved an organization with know-how on the creation of 3D assets and on historical information. These people collaborated with a developer who created some Logo commands on top of the Delphi interface for programming the MachineLab game engine. The development team also consisted of a researcher with knowledge in

the design of Logo-like microworlds and the two mathematics teachers with knowledge of Logo and microworld pedagogies. As one can imagine communication was extremely difficult and the result seemed limited to all.

Emergent Microworlds

This first version of MachineLab was built to see whether it was a workable strategy to add a Logo programming interface to realistic game engines. Our black and white box approach was looking for means to have a rich 3D environment as a black box basis for Logo-like microworlds. However, the Automata microworld although adequate for the project at hand, made us realize that without a mechanism to create mathematical representations and combine them dynamically with the 3D simulations, we could only achieve limited interactivity which would be of interest for the potential for students to generate meanings through the use of the microworlds. We thus used funding from subsequent projects to re-build MachineLab and port our Logo language onto a Java interface for the game engine. This enabled us to create mathematical representations such as variation tools and programmable 3D objects such as a turtle or a rope-like trace for its displacements. A full turtle geometry environment can thus be integrated with a 3D virtual reality world including geometrical or any other objects.

Navigational Mathematics

Represented Phenomenon

The user is in the role of giving directions for the creation and displacement of airplane avatars on a 3D geographical information system containing maps with geo-coded information. Displacements can be carried out either through cartesian (lat, long, height) or vector (angle theta, angle phi, length r) coordinates. They can also be done declaratively by providing the name of a location. All of these ways are dynamically linked so that a change in an input from one system creates changes in the inputs of the other and vice versa.

Level of Abstraction

The geographical space and the plane avatars are pretty concrete. Displacements however are done through mathematical representations of geometrical positions of a Mercator projection of the earth's elliptical shape. These displacements are done either through an interface where the values are put in for each displacement followed by the pushing of a 'go' button or by means of a Logo language with primitives for these displacements and for addressing avatar names.



Fig. 4. The 'Cruislet' microworld.

Concepts Embedded

Orientation and spatial awareness is one area of concepts. Cartesian coordinates and vector displacements are another, leading to the notion of geometrical systems and their link to analytical algebra.

User Interface

Description of the Scene

The scene is based on 'Cruiser', a GoogleEarth-like software originally addressed to other domains such as fleet management, publishing, real-estate and public information platform for municipalities. The 'Cruislet' microworld has the mathematical interface for avatar displacements and also camera displacement in relation to the avatars. It also has the Logo language for programmable avatar trips. At the moment there are maps of Greece and Cyprus but that is changing as more geo-coded information is added onto the Cruiser platform. The usual geographical data is contained as well as the possibility for users to add their own.

Manipulation

Manipulation is indirect. It comes in the form of avatar creation and displacement through either the special user interface for inputs to displacement values or through Logo.

Programmability

A full Logo language is available with special primitives for avatar creation, addressing, coordinate, vector and geographical displacements.

*Feedback from the Computer**When Changing the Parameters*

Values to coordinates or vector angles and length are input. Pressing a 'go' button causes the change.

As Result from the Experimentation

A large variety of experiments are of course possible by means of predicting where an avatar will go when a value or a combination of values are input.

Learning Process

Users can get the avatars to make specific displacements or sequences of displacements to create a trip. These can be done in any direction in 3D space. The view of the avatar also changes with respect to the position of the camera following it. This provides a significant freedom for activities and interests. Activities can also involve more than one avatar and the user can build dependencies on avatar movements using Logo programs. All these activities can be done by choosing to use the coordinate or the vector system and by inevitably making comparisons between the two. It's like using the London underground, you get used to thinking in absolute or relative coordinates.

Didactical Design

Didactical engineering will determine when and how a teacher might intervene to help the students make sense and articulate how the two systems work and how they can be made use of to determine locations and displacements. There is a variety of social orchestrations possible here from competitive (guess a flight given by the teacher) to game-like (guess my flight) to creative (create a flight and collect information on the way).

Type of Representations Used

The representations for displacements and the Logo language are executable formal mathematics. The displacements resulting from executions are mathematical in the sense that they are strictly defined on the Mercator projection. On the other hand they are realistic since the terrain is a 3D map with a lot of multimedia information on specific locations and the possibility for inserting one's own.

Procedure of Development

Developing this microworld required the negotiation of agendas of a research team and a corporate organization. Could this microworld become the basis for a future commercial

application as an add-on to Cruiser? Could it at the same time be a platform to build freely available mathematical games? Researchers and developers from two organizations collaborated in its development. They had had years of experience in such collaborations and this made it possible for the software to be developed on schedule.

Emergent Microworlds

Cruislet is a platform rather than a microworld in the sense that it has a very broad scope for activities. The process of using it to design more specific microworlds has just begun. The first one is based on the game 'guess my flight' where an avatar's displacements are dependent on another ones. The students explore the dependency and when they discover it play a game of creating dependencies for their peers to investigate. I'm sure many more microworlds will come.

Conclusions

Four half-baked microworlds were described, the first two were designed as such, the latter two were used as half-baked in that they provoked the design of other emerging microworlds. The breadth of technologies point to that the notion of half-baked microworlds is not technology dependent, it's rather a construct to think about the design and use of malleable educational software. What is in discussion here however is the template for describing such microworlds. This may look rather normal to the Logo community. However, it was the result of negotiation between teams within and outside this community and used as a tool to communicate design ideas for learning environments. It therefore portrays the kinds of things that communities outside our own may consider important in understanding the essence and function of microworlds in educational environments. I'm sure that given another configuration of teams, we would see a different, perhaps more elaborate such template. What's important however is to engage in the process of constructing a language to communicate with broader educational communities, now that technologies are becoming widely available and used. After 40 years, integration of Logo-like ideas and technologies in wider educational communities is proving essential for their presence in education beyond a recluse club of equally ageing enthusiasts.

Acknowledgements

'ESCALATE' Enhancing SCience Appeal in Learning through Argumentative inTERaction FP6-2004-Science-and-Society-11, 020790 (2006–2008).

'ReMath' – Representing Mathematics with Digital Media FP6, IST-4, STREP 026751 (2005–2008).

'Greekworks': Interactive tools for presentation and study of Ancient Greek Technologies. Ministry of Development, General Secretariat for Research and Technology, R&D Consortia in sectors of national priority, Action 4.5.1., P1 (2003–2005)

‘SEED’ – ‘Seeding cultural change in the school system through the generation of communities engaged in integrated educational and technological innovation’, European Community, IST, School of Tomorrow, IST-2000-25214.

‘Cruiser’, Talent S.A. (<http://www.talent.gr>)

References

- Agalianos, A.S. (1997). *A Cultural Studies Analysis of Logo in Education*, unpublished doctoral thesis, Institute of Education, Policy Studies & Mathematical Sciences, London.
- Bereiter, C., M. Scardamalia (2003). Learning to work creatively with knowledge. In E. De Corte, L. Verschaffel, N. Entwistle, J. van Merri (Eds.), *Unraveling Basic Components and Dimensions of Powerful Learning Environments. EARLI Advances in Learning and Instruction Series*. Retrieved from <http://iikit.org/fulltext/inresslearning.pdf>
- Blikstein, P., D. Cavallo (2002). Technology as a Trojan horse in school environments: the emergence of the learning atmosphere (II). In *Proceedings of the Interactive Computer Aided Learning International Workshop*, Carinthia Technology Institute, Villach, Austria, pp. 1–22.
- Clements, D., J. Sarama (1997). Research on Logo: a decade of progress. *Computers in the Schools*, **14**(1-2), 9–46.
- Cobb, P., J. Confrey, A. diSessa, R. Lehrer, L. Schauble (2003). Design experiments in educational research. *Educational Researcher*, **32**(1), 9–13.
- diSessa, A. (1997). Open toolsets: new ends and new means in learning mathematics and science with computers. In E. Pehkonen (Ed.), *Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education*, 1. Lahti, Finland, pp. 47–62.
- diSessa, A. (2000). *Changing Minds, Computers, Learning and Literacy*. Cambridge, MA: MIT Press.
- diSessa, A.A., F.S. Azevedo, O. Parnafes (2004). Issues in component computing: A synthetic review. *Interactive Learning Environments*, **12**, 109–159.
- Edwards, L.D. (1995). Microworlds as representations. In A. diSessa, C. Hoyles, R. Noss (Eds.), *Computers and Exploratory Learning*, Berlin/Heidelberg: Springer-Verlag, pp. 127–154.
- Goldenberg, P. (1999). Principles, art and craft in curriculum design: the case of connected geometry. *International Journal of Computers for Mathematical Learning*, **4**, 191–224.
- Hoyles, C., R. Noss, R. Adamson (2002). Rethinking the microworld idea. *Journal of Educational Computing Research*, Special issue on *Microworlds in Mathematics Education*, **27**(1-2), 29–53.
- Kafai, Y., M. Resnick (Eds.) (1996). *Constructionism in Practice*. Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey.
- Kynigos, C. (1995). Programming as a means of expressing and exploring ideas in a directive educational system: three case studies. In A. diSessa, C. Hoyles, R. Noss (Eds.), *Computers and Exploratory Learning*. Springer Verlag NATO ASI Series, pp. 399–420.
- Kynigos, C. (2002). Generating cultures for mathematical microworld development in a multi-organisational context. *Journal of Educational Computing Research*, **27**(1-2), 185–211.
- Kynigos, C. (2004). Black and white box approach to user empowerment with component computing. *Interactive Learning Environments*, **12**(1-2), 27–71.
- Kynigos, C. (2007) Half-baked microworlds in use in challenging teacher educators’ knowing. *international Journal of Computers for Mathematical Learning* (in press).
- Kynigos, C., S. Gavrilis (2006). Constructing a sinusoidal periodic covariation. In J. Novotna, H. Moraova, M. Kratka, M. Stehlikova (Eds.), *Proceedings of the 30th Conference of the International Group for the Psychology of Education*, **4**, 9–16. Charles University, Faculty of Education, Prague.
- Kynigos, C., N. Yiannoutsou, E. Alexopoulou, C. Kontogiannis (2006). A half baked Juggler game in use. In *Proceedings of the 1st World Conference for Fun 'n Games*. Preston, England, pp. 13–19.
- Laborde, C., C. Kynigos, K. Hollebrands, R. Strasser (2006). Teaching and learning geometry with technology. In A. Gutiérrez, P. Boero (Eds.), *Handbook of Research on the Psychology of Mathematics Education: Past, Present and Future*, Sense Publishers, pp. 275–304.
- Noss, R., C. Hoyles (1996). *Windows on Mathematical Meanings*. Kluwer Academic Publishers.

- Noss, R. (1992). The social shaping of computing in mathematics education. In D. Pimm and E. Love (Eds.), *The Teaching and Learning of School Mathematics*. Hodder & Stoughton.
- Papert, S. (2002). The turtle's long slow trip: macro-educological perspectives on microworlds. *Journal of Educational Computing Research*, **27**(1), 7–28.
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. New York, New York: Basic Books.
- Penner, D. (2001). Cognition, computers and synthetic science: building knowledge and meaning through modelling. In W. Secade (Ed.), *Review of Research in Education*. Washington, American Educational Research Association, pp. 1–36.
- Sarama, J., D. Clements (2002). Design of microworlds in mathematics and science education. *Journal of Educational Computing Research*, **27**(1), 1–3.
- Vergnaud, G. (1991). La Théorie des Champs Conceptuels. *Recherches en didactique des mathématiques*, **10**(2-3) 133–169.

C. Kynigos is an associate professor at the University of Athens and director of the Educational Technology Lab at the Faculty of Philosophy, Pedagogy and Psychology, School of Philosophy. He teaches courses in educational technology and mathematics education at undergraduate and postgraduate level. In the past 15 years, Chronis Kynigos has designed three logo-based platforms and employed them in research involving a) aspects of designing and generating socio-constructivist learning environments in the classroom (emphasis on mathematics), b) design and implementation of innovative teacher education methods and c) the design and implementation of methods to infuse innovation in the educational system. These are: E-slate, a component kit to construct microworlds, MachineLab, a programmable 3D simulator and Cruislet, a 3D navigation system over a GIS. A well known E-slate construct is Turtlerworlds, integrating symbolic with dynamic representations for mathematics. Chronis is director of the Educational Technology Lab, University of Athens, School of Philosophy, <http://etl.ppp.uoa.gr>. He's currently leading a European project on representing mathematics with digital media (<http://remath.cti.gr>).

LOGO mikropasauliai – tai objektų ribos integruotame dizaine

Chronis KYNIGOS

Šiame straipsnyje aprašoma bendruomenių, įtrauktų į skaitmeninių medijų kūrimo mokslui, skilimo problema. Straipsnis paremtas patirtimi, įgyta kuriant Mokslinės technologijos laboratorijoje Logo mikropasaulius trim skirtingom platformom. Tyrimai atlikti apdorojant komponentus kompiuteriu, naudojantis erdvinėmis žaidimo mašinomis ir erdvinėmis GIS navigacijomis. Šie mikropasauliai sukurti suprasti, kaip Logo gali padėti integruotam dizainui. Tokie mikropasauliai netiesiogiai egzistuoja ir be bendrijų, bet jie gali būti naudojami kaip integruoto dizaino pagalbininkai plėtojant bendradarbiavimą tarp tyrėjų, specialistų, mokytojų ir mokinių.