

# LOREM IPSUM LOGOS

PAVEL BOYTCHEV

Department of IT, Faculty of Mathematics and Informatics, Sofia University  
5, James Bourchier blvd, 1164 Sofia, Bulgaria  
e-mail: boytchev@fmi.uni-sofia.bg

The phrase *Lorem Ipsum* comes from the classical Latin literature and has been used by Cicero more than 2000 years ago. Since 1500's *Lorem Ipsum* was used as a dummy text in order to visualize the graphical layout through a meaningless text. The goal was to let readers concentrate on the visual representation only. The current paper focuses on the Lorem Ipsum problem in modern Logo implementations. Most of them seem to focus on the semantics of the Logo language paying less attention to the quality of the graphical output. The Turtle Graphics is typical for Logos to the extent that many people think they are two names for the same thing. Unfortunately Turtle Graphics did not change significantly since its introduction decades ago. And yet, it remains the only graphical instrumentarium of Logo. The Lorem Ipsum Logos is actually an attempt to introduce virtual reality and photo-realism to Logo users. Discussed are programs which can be treated entirely as a Lorem Ipsum text – the focus is entirely on the graphical result, and no source code is shown. Demonstrated are various techniques to achieve realism in the creation of Logo-based animations and 3D scenes. It is believed, that when users see and explore the other side of Logo, its Lorem Ipsum face, they will start to dare using Logo for solving an entirely new range of problems – problems, which match more closely to the interest of contemporary people.

**Keywords:** computer graphics, education, Logo, Elica

## 1 LOREM IPSUM

*“Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.”* This is just the beginning a text commonly used in templates and design layouts. Nowadays it is used whenever people want to put the focus on the graphical representation rather than on the meaning. When a person looks at a page with text, he or she subconsciously reads the words, thus shifting the attention to the meaning of the text. One of the easiest solutions to this problem is to use words and phrases that do not make sense to the majority of readers. There are even sites which offer Lorem Ipsum generators which can produce as long text as it is needed (Wilson, 2007).

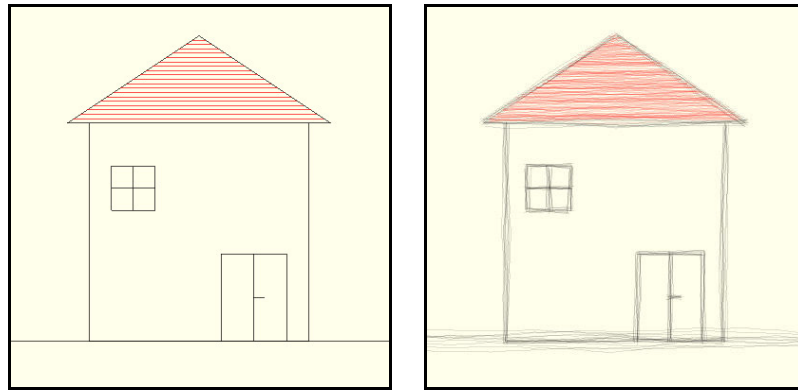
Actually, the Lorem Ipsum text originates (with modifications) from Cicero's “De finibus bonorum et malorum” (Cicero, 45 BC). Fragments of this classic Latin work were used 15 centuries later as dummy text, and since 1960s it became popular again. The benefit of using real Latin words, although ordered in a random manner, is that the distribution of characters is quite realistic.

## 2. LOREM IPSUM AND LOGO

Logo is an educational paradigm and a programming language which is both ancient and modern. The first implementation of Logo was created 40 years ago. Since then many developers had made their own implementations. As of today the number of known Logos is more than 170. The complete list can be found in the Logo Tree Project (Boytchev, 2007d).

How is Lorem Ipsum related to Logo? “Lorem” and “ipsum” are Latin words, while “logos” is a Greek one. However, the word “logos” does exist in the Latin vocabulary too although it has a different meaning.

When Lorem Ipsum is applied to Logo, it is usually done in a broad manner – it concerns how Logo implementations are designed and how Logo is used in educational environments. Traditionally Logo is taught through Turtle graphics. And traditionally most images created in this way look like old-fashioned computer drawings.



**FIGURE 1. COMPUTER-GENERATED (LEFT) AND HAND-DRAWN (RIGHT) HOUSE**

Unfortunately the lorem ipsum issue in Logo is more serious than this. When Logo is used to teach scientific or artistic skills through graphics, the main focus is on the algorithms, while the graphical realism is disregarded. The accuracy of algorithms and of reasoning is crucial for the learning process, and it naturally gathers most of the attention. The quality of the graphical output is considered merely as a side effect, which is not that essential.

Courses where graphics is important are usually Computer Graphics courses, and the Logo language is extremely rarely used in such circumstances. Educators prefer to use more main-stream industry-oriented languages like C, C++ or Java for high-quality graphics.

One of the features widely announced by Logo-aficionados is the immediate feedback of user commands. More specifically, when a student types a command controlling the turtle, it reacts immediately and the result (usually a drawing) is rendered on the screen right away. This feature is said to be important for capturing students’ interest. This was true 40, 30 and maybe 20 years ago. To draw a square on the screen was quite an inspirational achievement. On the other hand, modern world overwhelms people with huge amount of streaming visual content – TV commercials, video games, movies. This artificially generated content is worked up to the level that it looks closer to us than the reality itself. As a result people have higher expectations and higher amusement thresholds. A simple drawing is no more as attractive as it was decades ago. Educators relying on such quality of graphics have to spend enormous efforts for making their classes interesting. This is still possible to happen with introductory classes and classes for beginners, but is almost impossible with University-level students (a nice exception is the three volume CSLS book (Harvey, 1997) where serious Computer Science topics are discussed in the Logo context). And this is the lorem ipsum problem of Logos – no one *loremipsumizes* Logo any more. This could be one of the reasons for Logo losing the educational battle against other programming languages. Students and teachers are not that enthusiastic in learning Logo and learning *with* Logo as they were decades ago.

Modern Logos try to overcome the problem by introducing new features like using external images, incorporating multimedia in Logo projects, supporting multiple turtles and sprite-like methods for simple animations. Educators apply various techniques for teaching Computer Science, including music (Neuwirth, 2006, pp.15-28), interactive interfaces (Kalas,

2006, pp.13-24), robotics (Kerebayashi, S., Kamada, T., and Kanemune, S. 2006, pp.138-149) and art (Sendova, 2006, pp.71-82). However, few of the Logos could manage user-created 3D content in real time. One of these Logos is Elica (Boychev, 2007a). It is an OOP-based Logo implementation which supported 3D graphics and animation (Boychev, 2006, pp.351-360). It is suitable for building models and simulations by students (Boychev, 2005, pp.66-75).

### 3. DETURTLED LOGO

Elica's approach to computer graphics in Logo is based on the total deturtle (i.e. removing the turtle from Logo) followed by enturtle (i.e. adding back Turtle graphics). The basic graphical primitives in a deturtled Logo correspond to the basic objects in the analytical geometry – squares, circles, cubes, spheres, etc

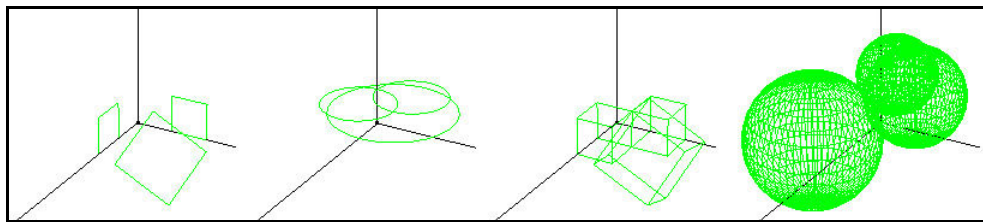


FIGURE 2. SOME OF THE PRIMITIVE 3D OBJECTS IN ELICA

These objects can be transformed by various methods of translation, scaling and rotations. Thus it is possible to obtain the complete variety of derivate objects in a way that reuses knowledge learned in other subjects (mainly Geometry). Figure 3 demonstrates cubes spread randomly on the surface of a sphere. Cubes have the default size and orientation, only their position is modified. The snapshot on the right has cubes with modified sizes and orientations. The orientations are aligned to the normal vectors of the sphere's surface, and the cubes are extended (scaled) along the corresponding normals.

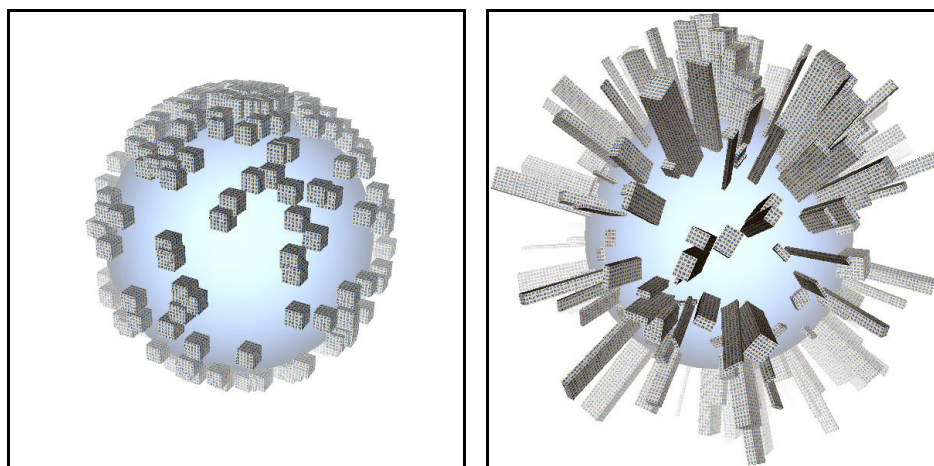
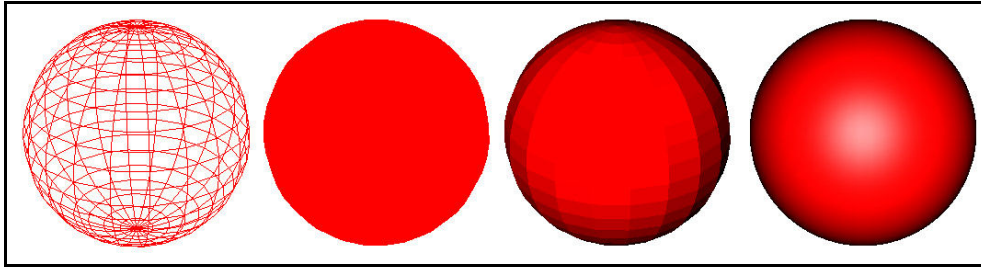


FIGURE 3. CHANGING ORIENTATION AND SCALE

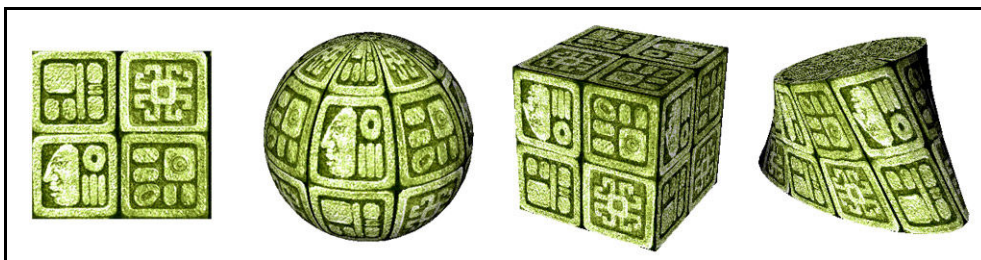
This example illustrates that there is no need to support common deformations of objects as standalone objects. An ellipse is a circle scaled along one of the axes, and all sorts of rectangles originate from the only one unit-size square.



**FIGURE 4. WIRE-FRAMED, SOLID, LIGHTED AND POLISHED SPHERE**

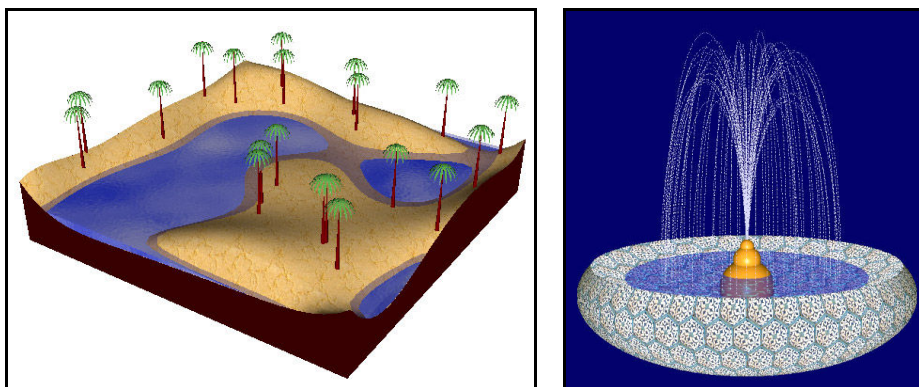
By having a 3D object transformed in a desired way, it is possible to make its appearance more natural by applying one of the many visual effects. Elica treats all objects in a consistent way, so visual effects can be applied to all objects. Figure 4 shows a wire-framed sphere which is made solid, then lit by a single light and finally smoothed.

Visual effects include features which are widely used in computer games and 3D scene rendering. To maintain better performance tiny details on the objects' surfaces are implemented through textures. Textures are images which are mapped onto the objects. These images are automatically curved depending on the surface.



**FIGURE 5. A TEXTURE (LEFT) MAPPED ONTO A SPHERE, A CUBE AND TWISTED CYLINDER.**

Figure 5 illustrates a square image (the left-most one) which is applied onto various objects – a sphere, a cube and a twisted cylinder with oval base. The method of projecting complies with the surface. If needed users may control the scale factor of the texture as well as its spin factor. Textures are applied in a variety of situations. They represent relatively cheap way to build realistic surfaces, like the dunes of the beach in figure 6; and the tiles and water surface of the fountain.



**FIGURE 6. APPLICATION OF TEXTURES IN MORE COMPLEX 3D SCENES.**

The meaning of ‘cheap’ here is that students do not need to be artists or graphical designers in order to use the appropriate texture. There are many sites on the net which provide sets of free textures. Sets are often grouped in topics – Nature, water, metal, marble, bricks, rocks and many others. Depending on the scene the students need just to select a texture and apply it. If it is needed they can change the texture with another one until they find the most suitable one. This easiness is one of the reasons for textures to be the favorite tool in generating 3D images and scenes.

#### 4. ENTURTLE LOGO

Elica’s graphical library provides basic and advanced functionality for creating graphics from scratch. Elica is a Logo implementation, and as such it is expected that it also supports Turtle graphics.

But Elica Logo is already deturtled, so to have Turtle graphics it is needed to enturtle it. In Elica this is done with a user-defined library. Because the turtle is not embedded in the system, it is possible to create various modifications of it, including the well-known traditional turtle.

Figure 7 shows the genealogical tree implemented in Elica. The root of all sorts of turtles is the *universal turtle* which provides the core functionality of all successors. There are several evolutionary braches springing out from the universal turtle. The *traditional turtle* is a direct successor, which triangular form could be replaced by user-define 3D shapes – *3D-shaped turtle* and *car turtle*. The traditional turtle and its derivates live all in 2D space.

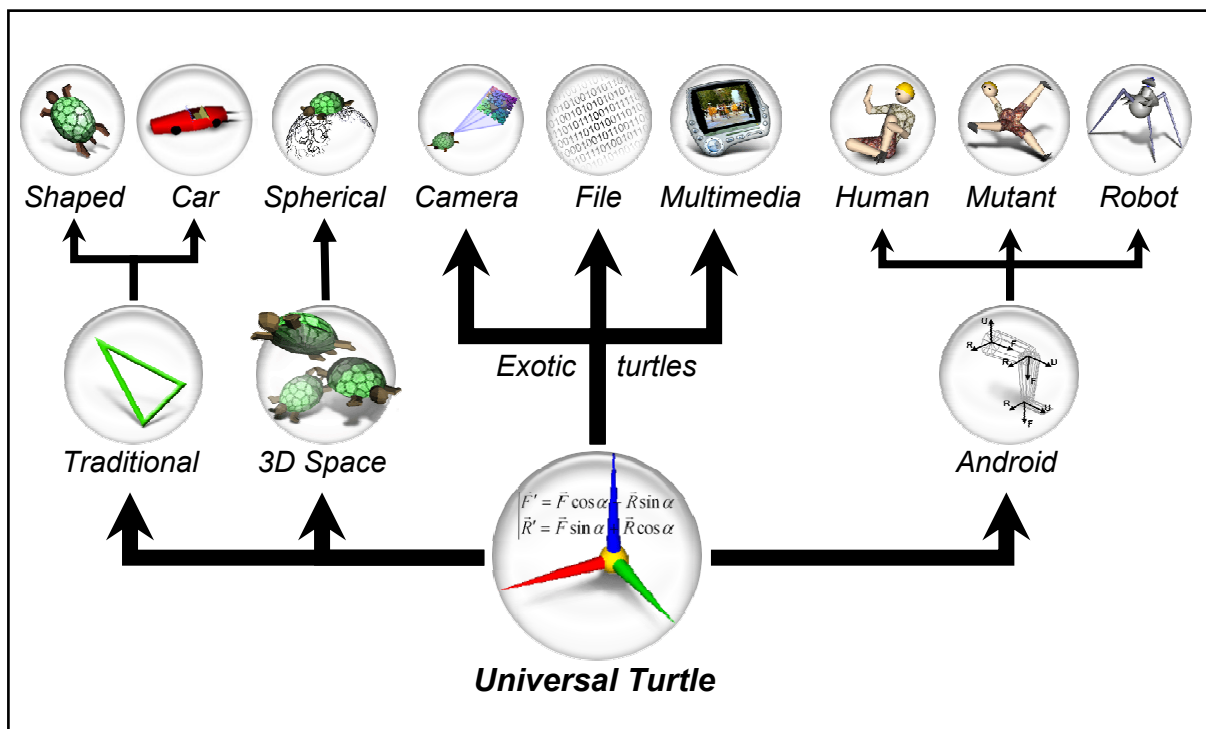


FIGURE 7. HIERARCHY OF TURTLES.

Another branch is for turtles which live in 3D space – the *3D turtle* and the *spherical turtle*. There is a set of *exotic turtles*; these are turtles which are not used to draw anything, but to bring turtle features onto other domains. The *camera turtle* is a turtle which determines

the viewpoint of the user in the 3D space and can be used to walk through it. The *file turtle* gives a turtle-like access to a file. Going forward and backward is projected as reading forward and backward from the file. A similar implementation is the *multimedia turtle* which represents the ‘tape’ of a video/audio file. Going forward makes the ‘tape’ wind forward, and going backward rewinds the ‘tape’.

The last branch displayed in Figure 7 is for the *android turtle*. This is a complex system of body parts connected via joints. The turtle provides means for controlling the joints and thus defining the posture of the whole system. The android turtle is ancestor of the *human turtle*, the *mutant turtle* and the *robot turtle*.

One of the main problems of having turtle graphics defined as a user library is that the turtle is somewhat slower than turtles in systems which have everything embedded and precompiled. However, this disadvantage is entirely overpowered by the functionality which it provides, namely, the flexibility to define custom turtles which can ‘live’ in virtually any domain, even the most turtle-unfriendly one.

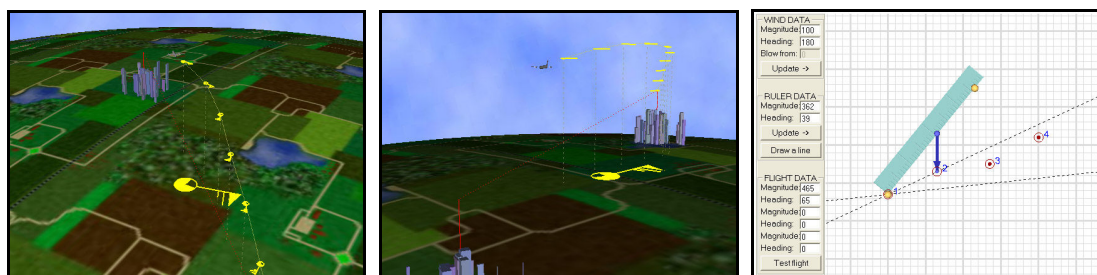
Another advantage is that Turtle graphics can be applied to a number of more advanced abstractions. The android-based turtles can be used to model complex movements. The library where these turtles are implemented contains functions to interpolate between two postures of a model and then to animate the movement. These features help the user to do with Turtle graphics things beyond any previous expectations and experiences.

#### 4. LOREM IPSUM IN THE CLASSROOM

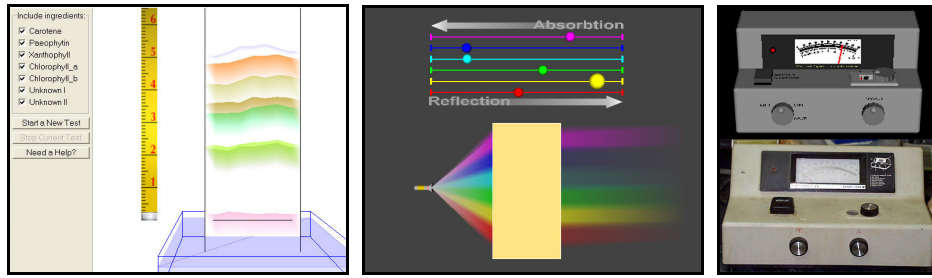
The application of Elica in schools and universities goes into two distinct streams. The first one is in the form of Elica-based applications, and the other is Elica itself used as a standalone programming environment.

For the recent 6-7 years several educational modules were created by using Elica. The first ones were requested by the Center for Innovation in Engineering and Science Education , and the School of Engineering, both at Stevens Institute of Technology, NJ, USA. The topics of the applications were rather broad. There was *Navigational Vectors* (Figure 8) used in Mathematics classes (Sayres, 2007), *Paper Chromatography* virtual lab where students could conduct experiments using paper chromatography methods; *Spectrum* lab for playing with absorption and reflection of light; and a *Virtual Spectrophotometer* device – a 3D model of a real Spectronic 20 device (Paselk, 2000) – see Figure 9.

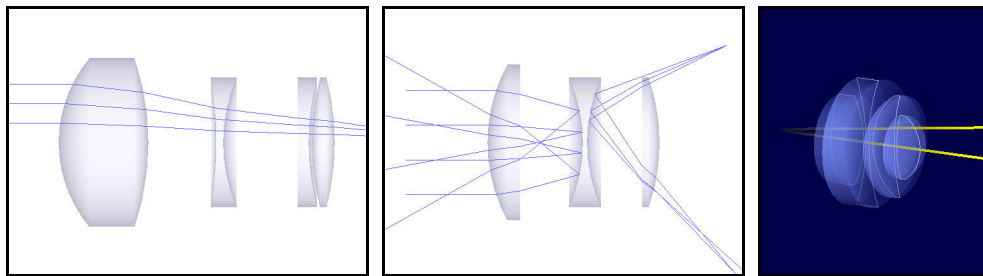
As an addition to these applications several others were made and used in various courses. The *Optics* module was used to model optical system and investigate their behavior depending on the properties of the lens, their material, and the direction of the light rays (Figure 10). The *Truss Analysis* module was used to build truss structures (towers, bridges) and to interactively determine the forces applied to individual beams (Figure 11). And finally, the *SoftLab* virtual laboratory was used to build and run lab scenarios in Thermodynamics.



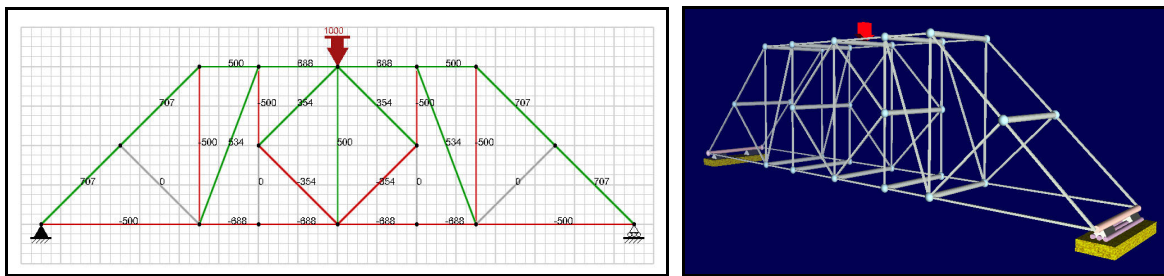
**FIGURE 8. NAVIGATIONAL VECTORS – TWO SNAPSHOT FROM THE SIMULATION MODULE AND ONE FROM THE DESIGN**



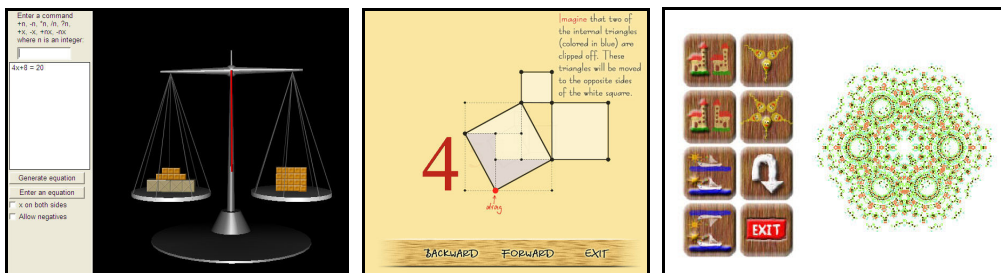
**FIGURE 9. PAPER CHROMATIGRAPHY MODULE, SPECTRUM LAB AND TWO SPECTROPHOTOMETERS – A PHOTO OF REAL SPECTRONIC 20 (DOWN) AND ITS VIRTUAL 3D MODEL (UP).**



**FIGURE 10. A ZEITES OPTICAL SYSTEM RECONSTRUCTED IN THE OPTICS MODULE, A SYSTEM WITH MIRROR SURFACES AND A 3D VIEW OF AN OPTICAL SYSTEM.**



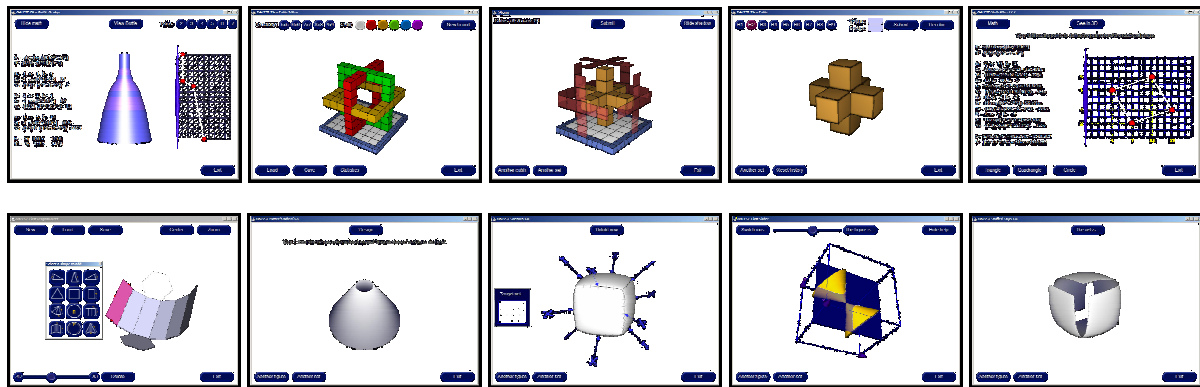
**FIGURE 11. A BRIDGE AND ITS 3D MODEL BUILT WITH TRUSS ANALYSIS MODULE**



**FIGURE 12. EQUATION BALANCE, PYTHAGORAS THEOREM AND PATTERN CONSTRUCTOR**

Except for university students, Elica-based application were successfully used with secondary-school students. *Equation Balance* is an application for learning how to solve equations, *Pithagorean Theorem* demonstrates interactively the Pythagorean theorem (Boychev, 2007c), and *Pattern Constructor* is an exploration toolbox for playing with symmetries (Figure 12).

At the end of 2006 and the beginning of 2007 a new set of applications was developed within the DALEST (Developing Active Learning Environment for Stereometry) Project. The project was co-funded by the European Union. The set contained 10 individual applications (Boychev, 2007b) covering various problems with intersections, nets, structures of cubes, folding/unfolding (Figure 13). The software was accompanied by a set of activities. Some of the applications were already tested with secondary-school students and the results were extremely encouraging (Boychev, P., Chehlarova T., and Sendova, E. 2007, pp. 109-119)



**FIGURE 13. DALEST-ELICA APPLICATIONS.**

Except for Elica-based applications, Elica is used in schools as a standalone programming environment. For example, the IT Department of the Faculty of Mathematics and Informatics at Sofia University teaches courses based on Elica – Computer Graphics, Computer Environments for Education, and Physics of Movement. Some courses are tailored for Bachelor levels, others for Master’s levels. Students are expected to create their own 3D animations. The best works are later included in the Online Elica Museum, which serves as an exhibition of the features of Logo (Figure 14).

## 5. CONCLUSION

As promised this paper discussed some techniques and shows some examples of graphics that go beyond the average style and complexity of Logo-generated images. No source code for any of the graphics is shown on purpose.

Lorem ipsum concepts about Logos do not undervalue the educational experience accumulated for the last 40 years. There are thousands of interesting problems that can be solved in text-only Logo environments, and much more in traditional Logo’s Turtle graphics. There are excellent books, textbooks and online resources where serious topics from Computer Science are covered with Logo programming. In this way Logo is a perfect choice for education. It can be used by a wide (in age and skills) range of users.

The goal of this paper is to show that it is possible to let Logo utilize new technologies in a way to create more attractive and more engaging learning environments. Access to 3D modeling and simulation is a factor which can be used to teach basic and advanced topics from Computer Science, Mathematics, Physics, and many other subjects.

Logo can still be used to create conventional drag-and-drop educational environments, but it can also be used in standalone mode where students create and manipulate 3D content in real time and all this is done programmatically. Immediate feedback, low threshold and high ceiling are still valid in this new context. As a result Logo can only benefit from inclusion of state-of-the-art functionality. Improvements in the graphics decrease the gap between the classroom and out-of-class experience. Although Logo can be now used to created models seen on TV or in video games, it does not enforce the passiveness typical for these media. Logo encourages active learning and does this in a very unostentatious way.



<b>Yin-Yang</b>  <a href="#">yiya.zip</a>	<b>RGB Pyramid</b>  <a href="#">rgbpy.zip</a>	<b>Neon Torus</b>  <a href="#">neoto.zip</a>	<b>Sheer Ghost</b>  <a href="#">sheegho.zip</a>	<b>Staring Eyes</b>  <a href="#">staeye.zip</a>	<b>Palm Beach</b>  <a href="#">pabea.zip</a>
<b>Mandelbrot</b>  <a href="#">mande.zip</a>	<b>Bounce In A Box</b>  <a href="#">boubo.zip</a>	<b>Butterfly</b>  <a href="#">butte.zip</a>	<b>Life Game</b>  <a href="#">liga.zip</a>	<b>Ortho Rings</b>  <a href="#">ori.zip</a>	<b>Rubik Cube</b>  <a href="#">rucu.zip</a>
<b>Hairy Ball</b>  <a href="#">haiba.zip</a>	<b>Pleated Star</b>  <a href="#">pleasta.zip</a>	<b>Pencil Drawing</b>  <a href="#">pedra.zip</a>	<b>Unreal Reality</b>  <a href="#">urea.zip</a>	<b>Leonardo Rings</b>  <a href="#">leori.zip</a>	<b>Hyperhyperboloid</b>  <a href="#">hyhybo.zip</a>
<b>Massive Massif</b>  <a href="#">mama.zip</a>	<b>Koch Tetraeder</b>  <a href="#">kote.zip</a>	<b>Math Shinkansen</b>  <a href="#">mashi.zip</a>	<b>Solar System</b>  <a href="#">sosy.zip</a>	<b>Analog Clock</b>  <a href="#">aclo.zip</a>	<b>Grass in the Wind</b>  <a href="#">grawi.zip</a>
<b>The Matrix Effect</b>  <a href="#">mae.zip</a>	<b>Face Caricature</b>  <a href="#">faca.zip</a>	<b>Ant's Walk</b>  <a href="#">awa.zip</a>	<b>Fun Fair</b>  <a href="#">fufai.zip</a>	<b>Water Fountain</b>  <a href="#">wafou.zip</a>	<b>Circus Dancer</b>  <a href="#">cida.zip</a>
<b>Vintage Photo</b>  <a href="#">vipho.zip</a>	<b>Morning Meadow</b>  <a href="#">momea.zip</a>	<b>Kaleidoscope</b>  <a href="#">kalei.zip</a>	<b>Water Waves</b>  <a href="#">wawa.zip</a>	<b>Hoola Hoops</b>  <a href="#">hoofoo.zip</a>	<b>Rotational Surfaces</b>  <a href="#">rosu.zip</a>
<b>Bouncing Balls</b>  <a href="#">bouba.zip</a>	<b>Earth and Moon</b>  <a href="#">eamoo.zip</a>	<b>Lernaean Pentapus</b>  <a href="#">lepe.zip</a>	<b>Kissing Spheres</b>  <a href="#">kisphe.zip</a>	<b>Differential Gears</b>  <a href="#">digea.zip</a>	<b>Curious Worm</b>  <a href="#">cuwo.zip</a>
<b>Fireworks</b>  <a href="#">fire.zip</a>	<b>Falling Bricks</b>  <a href="#">fabri.zip</a>	<b>Molecules</b>  <a href="#">mole.zip</a>	<b>Multiple Layers</b>  <a href="#">mulaye.zip</a>	<b>Transparent Cube</b>  <a href="#">tracu.zip</a>	<b>Constellations</b>  <a href="#">conste.zip</a>
<b>Circle Spline</b>  <a href="#">cispli.zip</a>	<b>Fisheye Effect</b>  <a href="#">fie.zip</a>	<b>X-Ray Plane</b>  <a href="#">xrapla.zip</a>	<b>Bug Spiral</b>  <a href="#">buspi.zip</a>	<b>Urban and Rural</b>  <a href="#">uru.zip</a>	<b>Precious Pearls</b>  <a href="#">prepe.zip</a>

FIGURE 14. PROJECTS IN THE ONLINE ELICA MUSEUM.

## REFERENCES

1. Boytchev, P. (2007a). Elica Home Page. Retrieved April 29, 2007, from <http://www.elica.net>
2. Boytchev, P. (2007b). Elica DALEST Applications. Retrieved April 29, 2007, from <http://www.elica.net/site/download/dalest/dalest.html>
3. Boytchev, P. (2007c). Elica Pythagorean Theorem. Retrieved April 29, 2007, from <http://www.elica.net/site/download/pythagorean/pythagorean.html>
4. Boytchev, P. (2007d). The Logo Tree Project. Retrieved April 29, 2007, from <http://elica.net/download/papers/LogoTreeProject.pdf>
5. Boytchev, P. (2007e). Online Elica Museum. Retrieved April 29, 2007, from <http://elica.net/site/museum/museum.html>
6. Boytchev, P., Chehlarova T., and Sendova, E. (2007). Enhancing spatial imagination of young students by activities in 3D Elica applications. *Proceedings of the 36<sup>th</sup> Spring Conference of the Union of Bulgarian Mathematicians*, April 2007, pp. 109-119
7. Boytchev, P. (2006), Natural Object-oriented programming: The evolution metaphor behind Elica, *Proceedings of the 2<sup>nd</sup> ISSEP International Conference*, eds.: Dagiene V., Mittermeir R., November 2006, pp. 351-360.
8. Boytchev, P. (2005), Using Logo to model and animate, *Proceedings of the 10<sup>th</sup> Eurologo Conference*, eds.: Gregorczyk, G., Walat, A., Kranas, W., and Borowiecki, M., August 2005, pp. 66-75.
9. Cicero, M. (45 BC). DE FINIBVS BONORVM ET MALORVM. Retrieved April 29, 2007, from <http://www.thelatinlibrary.com/cicero/fin.shtml>
10. DALEST (2007). Dalest Project Web Page. Retrieved April 29, 2007, from <http://www.ucy.ac.cy/dalest>
11. Harvey, B (1997). *Computer Science Logo Style*. MIT Press.
12. Kalas, I. (2006), Discovering informatics fundamentals through interactive interfaces for learning, *ISSEP 2006, LNCS 4226*, ed.: Mittermeir R., Springer-Verlag Berlin Heidelberg, November 2006, pp. 13-24.
13. Kerebayashi, S., Kamada, T., and Kanemune, S. (2006), Learning computer programming with autonomous robots, *ISSEP 2006, LNCS 4226*, ed.: Mittermeir R., Springer-Verlag Berlin Heidelberg, November 2006, pp. 138-149.
14. Neuwirth, E. (2006), Music as a paradigm to teach Computer Science principles, *Proceedings of the 2nd ISSEP International Conference*, eds.: Dagiene V., Mittermeir R., November 2006, pp. 15-28.
15. Paselk, R. (2000). Spectronic 20. Retrieved April 29, 2007, from <http://humboldt.edu/~scimus/HSC.54-70/Descriptions/Spec20.htm>
16. Sayres, J. (2007). Navigational Vectors Project. Retrieved April 29, 2007, from <http://www.k12science.org/curriculum/vectors>
17. Sendova, E. (2006), Handling the diversity of learners' interests by putting informatics content in various contexts, *ISSEP 2006, LNCS 4226*, ed.: Mittermeir R., Springer-Verlag Berlin Heidelberg, November 2006, pp. 71-82.
18. Wilson, J. (2007). Lorem Ipsum. Retrieved April 29, 2007, from <http://www.lipsum.com>