



USING PROGRAMMING ENVIRONMENTS IN TEACHING GIS CONCEPTS

Imre Bornemisza, Pavel Boytchev

Abstract: When teaching Geographical Information Systems (GIS) it is important to explain the theoretical elements, but it is also necessary to illustrate the basic functions with examples. To help students it is advisable to separate the teaching of fundamentals from the practical application of complex GIS software. A simple programming language like Logo creates opportunities for acquiring knowledge in an easy way. The development of a program makes it possible to understand the most important categories. If the program contains query functions as well, students can learn more about problem solution methods.

This paper also demonstrates the power of the Logo family of languages which is delivered by different implementations. Elica Logo is used to build simple GIS tools that provide a very wide range of educational problems and lead to numerous ideas for further explorations. The nice intuitive 3D interface is an attractive approach to gain and retain student's engagement in the learning process.

Zusammenfassung: Beim Lehren von Geografische Informations-Systeme (GIS) ist es sehr wichtig die theoretische Elemente zu erklären, aber es ist auch nötig, die einfachste Funktionen durch Beispiele zu illustrieren. Um den Schülern zu helfen, handelt sich um das Lehren den Grundlagen von dem praktischen Benutzung der GIS Software. Eine einfache Programmiersprache wie Logo gibt die Möglichkeit, die nötige Kenntnisse leicht im zu erwerben. Die Entwicklung eines Programms hilft beim Verstanden der wichtigsten Kategorien. Wenn das Programm auch ein Anfrage-Funktion enthält, so können die Schülern auch Problemlösungstechniken Kennenlernen.

Dieser Artikel stellt da, die Leistung der Logo Familie unter den Programmiersprachen, die in verschiedenen Variante ausgegeben ist. Elica Logo wird einfache GIS Instrumente zu bauen benutzt, die ein sehr breiten Gebiet von Schulungsproblemen unterstützen, und die für die weitere Forschung Ideen geben. Der freundliche, intuitive 3D Oberfläche ist ein anmutiges Instrument, die beim Schulungsprozess sehr nützlich ist.

Key words: GIS, teaching, Imagine Logo, Elica

1. Introduction

When students learn GIS fundamentals, they are learning the theoretical components, however, they must use simultaneously both theory and practice. The clear understanding of the most simple functions is especially important. But working with complex software without any real understanding of the steps can be confusing. In this situation a more simple tool may be of much help to learn the fundamentals easily.

Most office software has some functions for visualization like maps, charts, etc. These functions are helpful but too restricted. To demonstrate data import, data management, reclassification, displaying and querying we need a more flexible tool. The most flexible one is a programming language, usable for all problems.

Selecting a language is not a key issue, since quite a few are used by geographers: Visual Basic (Slocum & Yoder, 1996), Python (Karssenberget al, 2007), shell script (Neteler, 2003) etc. The most important aspect of choosing the language is that students should be familiar with it.

Logo is very adapted for our goal, it is easy to use, and most students learned it at primary school (Turcsányi-Szabó, 1995). In most schools in Hungary Logo is the first learned programming language at the age of 10-12. Students, who later do not specialize in informatics or programming, will be familiar only with this language.

Logo has many variants – according to Pavel Boytchev (2008) almost 200 implementations, out of which 50 are active and under development. We found also several precedents for using Logo programming language in the field of GIS, for example NetLogo at the Old Dominion University at Norfolk (Liebert et al, 2008), Cruislet Logo at the University of Athens (Alexopoulou et al, 2007) or Elica (Boytchev, 2002). In Hungary Comenius Logo was the most popular before 2005, today Imagine is learnt by most students (Rozgonyi-Borus, <http://www.sulinet.hu>; ELTE, <http://imagine.elte.hu>). Imagine is a commercial software, but Educatio Kht. (Educatio Public Company) bought the license for schools, teachers and students speaking Hungarian, and the software can be used at home as well (Sulinet, <http://logo.sulinet.hu>). Imagine runs relatively well (except of functions setButtons and textBox) under GNU/Linux (Ubuntu 8.04) with the wine package (version 1.0). Thus Hungarian students do not have to pay neither for the operating system nor for the programming environment. Students all over the world can use other Logo variants, for example Berkeley Logo (UCBLogo – accessible for downloading at <http://www.cs.berkeley.edu/~bh/logo.html>) or Elica (<http://www.elica.net>).

2. Drawing of maps in Imagine Logo

In a student project we developed a program in Imagine Logo to draw maps. The program has query function as well. The complete program source code is accessible for downloading at <http://www.ttk.pte.hu/ami/born/imagimap> for studying. The most part of the program can be understood easily; this article describes in detail some specific functions only.

Data processing

The Logo program uses a sample dataset (DTM: Digital Terrain Model or DEM: Digital Elevation Model) that can be downloaded from the homepage of the Institute of Geodesy, Cartography and Remote Sensing, Hungary (FÖMI, <http://fish.fomi.hu>). The data file shows X, Y, Z coordinates of the Hungarian village Osztopán. The data is an ASCII file, with X, Y, Z coordinates, separated with tabs or spaces, like in the next rows:

544000	131000	158.78
544025	131000	158.76
544050	131000	158.75
544075	131000	158.58
...

The structure of the program

The program clears the text and graphic windows, and then calls the following functions:

- `sizeOfWindow`: sets the size of the graphic window.
- `readFullFile`: opens the DEM file, reads X, Y, Z values to an array and closes file.
- `minMaxOfXYZ`: calculates minimum and maximum values of the coordinates.
- `scalingParameter`, `sizeOfPen`: calculates and sets the optimal scale parameter and pen size.
- `drawingMap`: the substance of the program – draws the map.
- `lastOut`: draws end writes all the rest to do (legend, coordinate grid, coordinate values, north arrow, map title etc.).
- `buttonsToWork`: draws a button set to access more functions (redrawing, querying etc.)

Function readFullFile

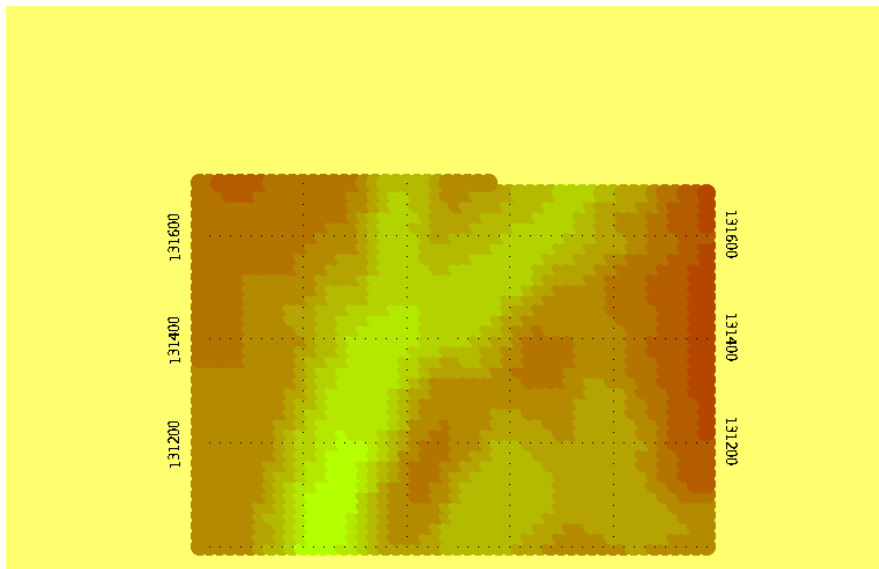
Opens the file and reads each row as a list into an array. The Replace function in Imagine Logo will add the new data to the end of the list, if the index is more than the length of the list. In this way the array becomes an increasing list of lists:

```
[ 544000 131000 158.78 ] [ 544025 131000 158.76 ] [ 544050 131000 158.75 ] ...
```

Over more than 10 000 elements array handling becomes very slow, filling the array may need more than 15 minutes.

Function drawingMap

The colouring of the map is similar to the original relief maps. We can realize this feature with the RGB coding. The function draws the map row by row (Picture 1).



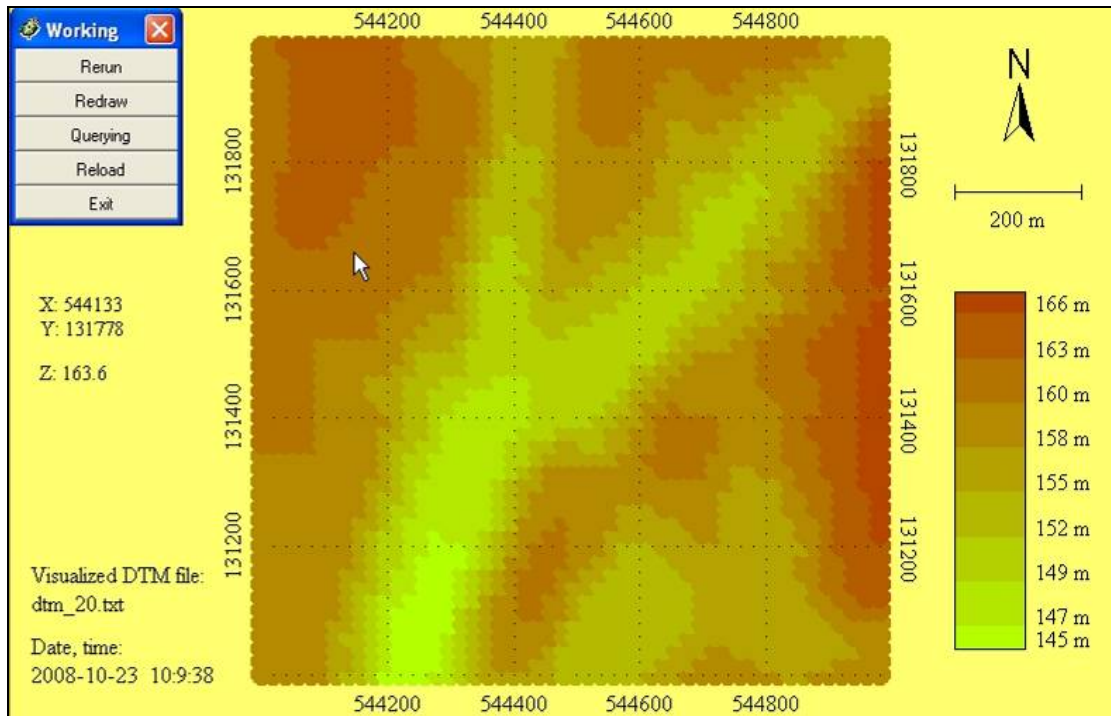
Picture 1. *The drawing of the map row by row*

Event onMouseMove

This event writes the coordinates of the mouse in real time. Using this event we can transform the mouse coordinates in Logo into real coordinates on the map. For the process we need the scaling parameter used when drawing the map.

Querying Z coordinate of selected point

Using this function we are coming to a typical problem: how to find elevation data of a point, which is not in the database. The solution is a simple method of interpolation. After each mouse click the function searches for the points nearest to the mouse pointer. If the searching distance parameter is greater than the distance of two neighbouring points, the function finds more than one point. Then the average of the found points Z values are calculated. If the function does not find any points, it prints “No data”, meaning that the mouse is very likely out of the map region. The function shows the result on the map (Picture 2).



Picture 2. The map drawn by turtle – with result of querying

3. 3D maps with Elica Logo

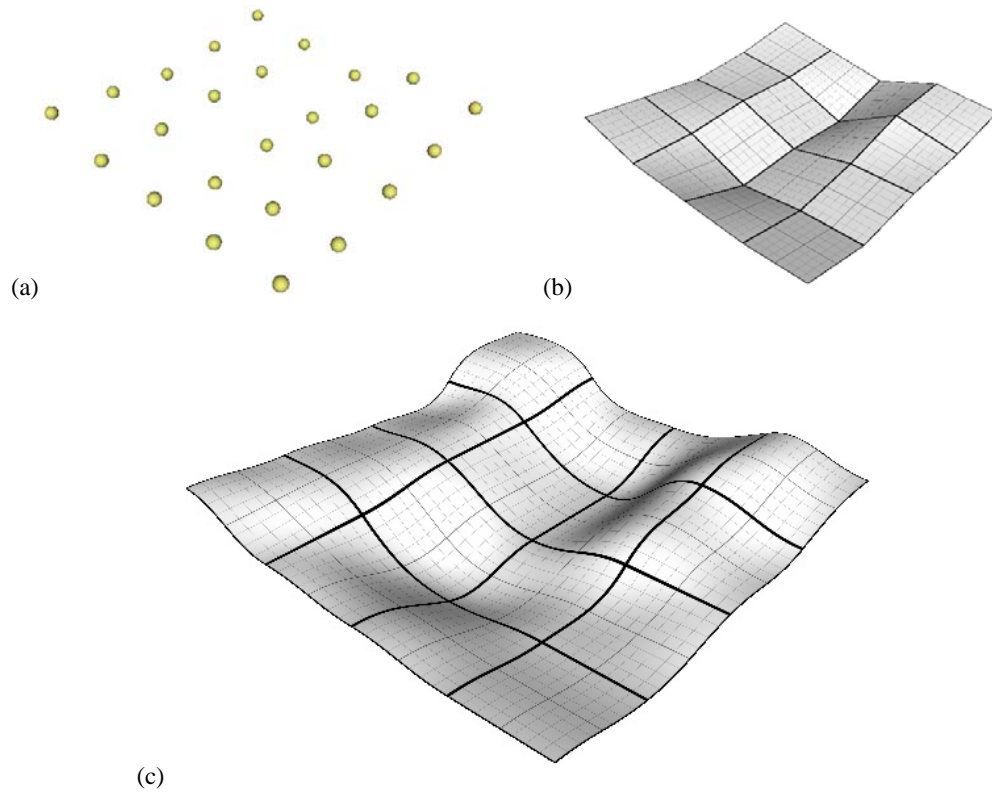
The variety of Logo implementations poses an interesting question to people who learn or use it: Which Logo is the best one? The answer is that it is not possible to define which one is the best, because each Logo is good for some types of task, and not so appropriate for others.

Studying the feedback from users working with Logo-based GIS-related software, one of the most important issues was to have three-dimensional maps including some landmarks. There are at least three Logo versions which support some kinds of 3D scene, where Elica (Boytchev, <http://www.elica.net>) and aUCBLogo (Micheler, <http://www.aucblogo.org>) are the most advanced ones. This section describes possible usage of Elica Logo to create and work with 3D maps.

The ground surface as defined in files with geographical data is represented as a set of altitudes for preselected locations. Usually these locations are the nodes of a virtual grid that spans over the terrain – Picture 3(a). Reconstruction of the actual surface cannot be done accurately without more detailed information and the most straightforward approach is to define quadrilaterals of which vertices match the preselected locations – Picture 3(b).

However, using flat patches does not produce a smooth surface. It is possible to reduce the size of the patches thus gaining some level of smoothness, but this will be at the cost of more data and reduced performance.

Another approach is to use NURBS surfaces (Farin, 1990; Piegl, 1991). NURBS stands for Non-Uniform Rational B-Spline and it is a mathematical method for defining free-form surfaces. Using such types of surfaces makes it possible to achieve smooth transition between patches (see Picture 3(c)). For a higher level of realism a texture could be mapped onto the surface and this will generate a nice 3D view of the terrain – Picture 4.



Picture 3. *Grid of dots (a), quadrilaterals (b) and NURBS surfaces (c)*



Picture 4. *A satellite photo mapped as a texture onto the NURBS surface*

NURBS surfaces are defined mathematically and students can use this definition to query the altitude of any points of the map. They are not required to make their own interpolation, the mathematical definition works for any points within the boundaries of the surface.

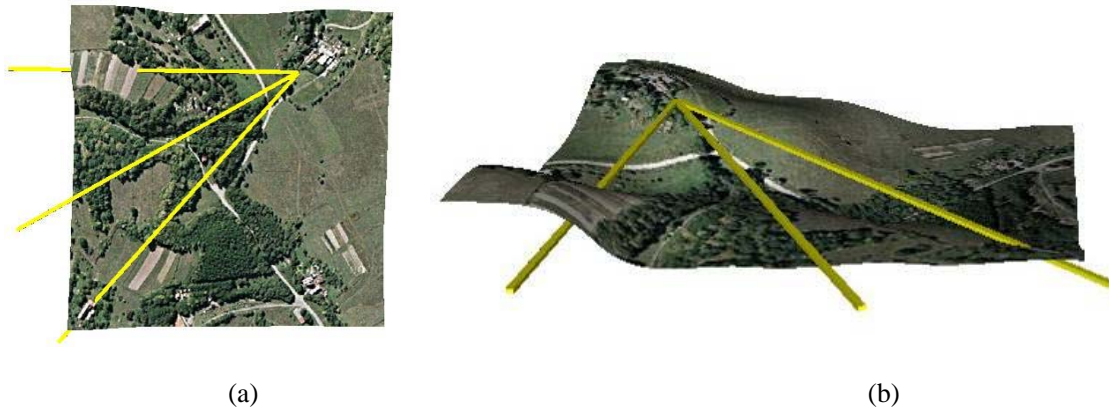
Having a 3D map of a terrain that can be freely rotated and zoomed in/out is not the only benefit of building maps in a 3D-enabled Logo dialect. Querying altitudes of points is straightforward and leads to a new world of interesting and challenging educational problems. We will describe briefly only three of them – planning bridge construction, finding the best place for a TV tower and flood analysis.

Bridge construction

The problem with bridge construction is that we want to build a bridge from one specific location to another unknown location across the valley. The actual problem is that the bridge must be horizontal and if the specific location is too high it would be hard to find the corresponding place for the other end of the bridge.

Picture 5(a) shows a top-view of three possible directions for a bridge. By examining the altitudes of various locations students can calculate that the middle bridge is not possible, because it does not reach any point from the terrain. From the other two bridges – one of them is twice shorter than the other one (and thus it is approximately much cheaper to construct). Nevertheless, which one of the two is a better one is already not a mathematical problem but more like engineering or social issue.

The second snapshot, Picture 5(b), is a visual proof that only two of the bridges are possible. Their structures poke through the surface and determine the exact location of their other ends. The middle bridge ‘hovers’ in the air without touching the ground.



Picture 5. *Visual proof of possible and impossible bridges*

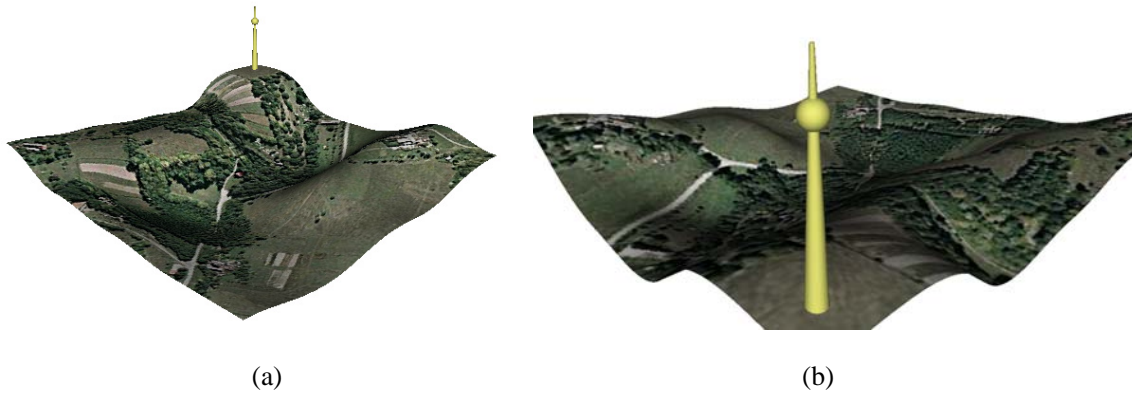
The TV tower

Another problem is that a GIS-based 3D map can help to find the location and the height of a TV tower if we know the region that must be covered by its signal. Naturally, locations are on top of hills or on small mountains.

Picture 6(a) represents one possible location of the tower. The second snapshot (Picture 6(b)) is a view of the valley from the top of the tower. It is obvious that this place for a tower is good, because it provides direct view to the entire valley.

Solving this problem is in some way the opposite of the problem with bridges. In the case with the TV tower we want our imaginary lines to reach every point of the valley without crossing any intermediate peaks.

However, the actual planning for bridges and TV towers is much more complex, of course. Within the GIS domain we ignore issues like underground geological structure, ecological domains, social effect, etc.



Picture 6. A tower as seen from the valley (a) and the valley as seen from the tower (b)

Flood analysis

Using GIS software in education would be much more beneficial if it allowed students to explore various problems. The previous two problems of finding best locations for bridges and TV towers are examples of one class of problems. Students could also try to build roads, plan locations of new residential areas, and locate the best place for agricultural areas, etc.

There are several other classes of problems that can be explored. For example, we will show a visual approach to flood analysis, namely, depending on the water level to find which regions will be affected. Flooding can be caused by many phenomena – excessive rains, ice melting in the spring or broken barrage of a reservoir.

Flood analysis can span to a variety of sub-problems, like:

- What residential areas will be flooded (this information defines evacuation plans for the population)
- What roads will be flooded (this information helps to determine alternative routes for transportation and commuting)
- What plants will be flooded (this information is used to find out possible shortage of food, electricity, etc.)

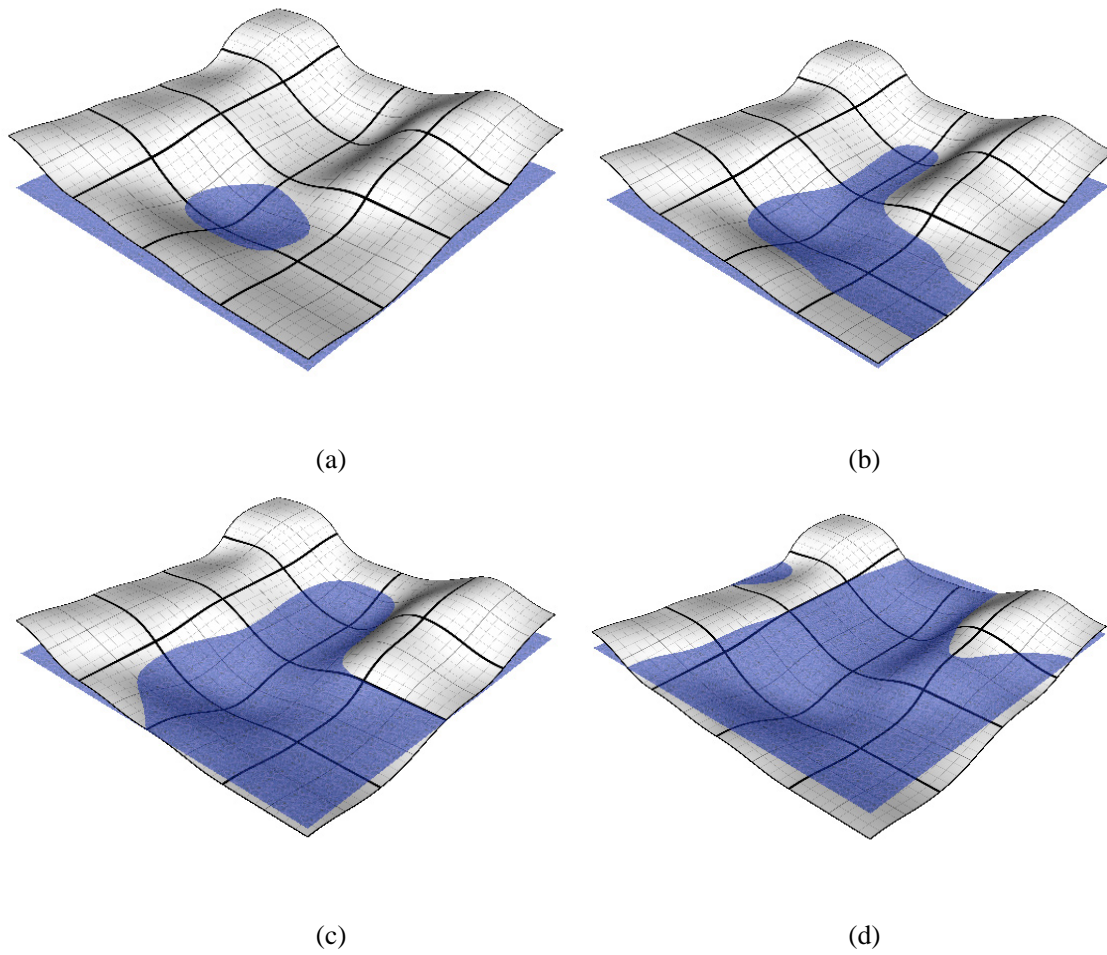
One of the greatest benefits of open-source Logo programs is that users are not only allowed to modify the code, but they are also encouraged to do so. The images of bridges and TV towers in the previous two cases are not a part from the map – they were added subsequently to the 3D scene.

In a similar way flooding can be simulated by a flat surface which can be interactively moved up or down representing different levels of water.

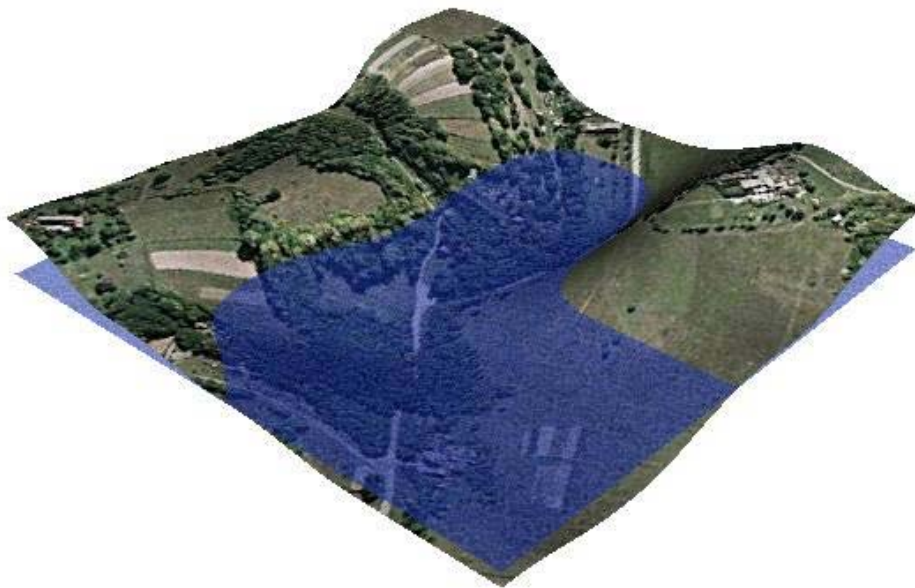
Picture 7 shows four situations at different levels of the water and Picture 8 shows a textured version of flooding. Providing a visual feedback of flooding makes a big difference to how people recognize the effect of this disaster. However, it also demonstrates the importance of the GIS information and the way it can be made useful for people.

The natural disaster scenario leads to other educational problems. Students can predict avalanche routes, find safe locations in a case of a radioactive contamination.

There are many other situations in which GIS-related software can be used to solve problems. For example, they can use it to plan mountain trips, or sight-seeing excursions. They can use it to practice their programming and mathematical skills. As it turns out the possibilities are limited only by our imagination.



Picture 7. *Flooded regions at different water levels*



Picture 8. *Flooded regions shown on a textured surface*

4. Conclusion

The most notable power of using Logo-based programming environments is that students have access to the code that creates and manipulates the maps. This is a challenging opportunity for them to see how map processing is implemented and is an inspiration for them to modify the code and apply their own ideas and algorithms.

The presented methods and tools demonstrate that Logo is very adapted for teaching GIS. The complex GIS software is often too difficult for teaching fundamentals. Due to their flexible nature programming languages like Logo give the possibility to resolve almost any kind of geometrical problem. After developing several small GIS-related programs students become confident in their practical skills and knowledge. The aim of this article was to show the first steps.

Literature

- [1] Alexopoulou, E., Kynigos, C., Markopoulos, C. (2007), Changing a half-baked 3D navigational game, *EuroLogo conference*, Bratislava.
<http://www.di.unito.it/~barbara/MicRobot/AttiEuroLogo2007/proceedings/P-Alexopoulou.pdf> [February 2009]
- [2] Boytchev, P. Elica Home Page, <http://www.elica.net> [February 2009]
- [3] Boytchev, P. (2002), North Pole Adventures. *International Journal of Computers for Mathematical Learning*.
<http://www.springerlink.com/content/t7j4v2853244q45w> [February 2009]
- [4] Boytchev, P. (2008), Logo Tree Project.
<http://www.elica.net/download/papers/LogoTreeProject.pdf> [February 2009]
- [5] Farin, G. (1990) *Curves and Surfaces for Computer-Aided Geometric Design*, Academic Press Professional, San Diego
- [6] FÖMI Földmérési és Távérzékelési Intézet, FÖMI – Letöltés (Institute of Geodesy, Cartography and Remote Sensing – Downloading).
http://fish.fomi.hu/letoltes/nyilvanos/adatszolgaltatasmintak/dtm_5x5m_ASCII.zip [February 2009]
- [7] Karszenberg, D., de Jong, K. & van der Kwast, J. (2007), Modelling landscape dynamics with Python. *International Journal of Geographical Information Science*, 21, 5, pp. 483-495.
- [8] ELTE, TTK, Imagine Logo. <http://imagine.elte.hu> [February 2009]
- [9] Liebert, K., Earnest, D. C. & Tolk, A. (2008), Using GIS Vector Data to Build Virtual Environments For Agent Based Models,
<http://delivery.acm.org/10.1145/1410000/1400556/p45-liebert.pdf> [February 2009]
- [10] Micheler, A., aUCBLogo's Home,
<http://www.aucblogo.org> [February 2009 Febr]
- [11] Neteler, M. (2003), *Open source GIS, a GRASS GIS approach*, Springer
- [12] Piegl, L. A. (1991), On NURBS: A Survey, *IEEE Computer Graphics and Applications*, 11, 1, pp. 55-71.
- [13] Rozgonyi-Borus F., Temetni jöttünk, nem dicsérni! Comenius LOGO és Imagine LOGO közti különbségek. (We are coming to bury, not to praise! Differences between Comenius Logo and Imagine Logo).
<http://www.sulinet.hu/tart/ncikk/Rad/0/31493/index.html> [February 2009]
- [14] Slocum, T. A. & Yoder, S. C. (1996), Using Visual Basic to Teach Programming for Geographers. *Journal of Geography*, 95, 5, pp. 194-199.
- [15] Sulinet, Imagine országlicence (Country License).

<http://logo.sulinet.hu/license.html> [February 2009]

- [16] Turcsányi-Szabó M. (1995), Q: WHY Logo? A: To understand HOW?. Proc. *5th Eurologo Conference*, Birmingham

Authors

Imre Bornemisza, University of Pécs, Faculty of Sciences, Institute of Mathematics and Informatics, Pécs, Hungary, e-mail: born@gamma.ttk.pte.hu

Pavel Boytchev, University of Sofia, Faculty of Mathematics and Informatics, Dept. of Information Technologies, Sofia, Bulgaria, e-mail: boytchev@fmi.uni-sofia.bg