

Exploring visible mathematics with IMAGINE

Building new mathematical cultures with a powerful computational system

Ivan Kalas and Andrej Blaho

Department of Informatics Education, Comenius University, 842 48 Bratislava, Slovakia; kalas@fmph.unia.sk; blaho@fmph.uniba.sk

Abstract: In our paper we explore how programmable pictures together with events, parallel independent processes and direct manipulation tools can be used for building powerful interactive visual elements and provide rich environments for exploring basic mathematical concepts. To visualize the concepts we use IMAGINE turtles, the shapes of which are specified by the Logo language. Thus we achieve high interactivity in the resulting microworlds. Children can easily create such objects, control them, combine, move, group, match, etc. We hope that new features of IMAGINE will inspire math teachers and developers to create new visible math educational materials.

Key words: elementary education, organising for learning, curriculum

1. INTRODUCTION

In his inaugural lecture, Noss (1997) compared the computer to a mathematical piano, saying the computer "...can make tangible a mathematical object in much the same way that a piano can make tangible (or at least audible) the meaning of F#." He also said that powerful computational systems "... offer alternative means to express mathematical relationships, novel kinds of symbolism, and innovative ways to manipulate mathematical objects: In short, the emergence of new mathematical cultures." IMAGINE (Kalas and Blaho 2000) is such computational system with ambitions to stimulate the emergence of new cultures for constructing, exploring and understanding.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35668-6_17](https://doi.org/10.1007/978-0-387-35668-6_17)

G. Marshall et al. (eds.), *Learning in School, Home and Community*

© IFIP International Federation for Information Processing 2003

IMAGINE represents new generation of Logo. Researchers of the integration of ICT into learning processes remember that Logo always defined its goal as developing problem-solving and logical-thinking skills, and allowing young learners to build their own “intellectual structures through estimation, interaction, experience and revision” (Watt 1983). Logo has offered simple tools and mechanisms: (a) one or multiple turtles, i.e., graphical objects which have certain shapes and can draw; (b) a vocabulary of simple commands such as forward, right, etc.; and (c) a mechanism for defining the user’s own procedures. The essentials allow and encourage learners to explore, create and understand.

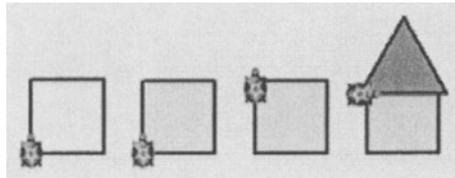


Figure 1. Traditional Logo activities

Recent developments in educational computer environments have added several new tools and techniques, as illustrated by IMAGINE. Besides traditional elements of turtle geometry as shown in Figure 1, it offers new attractive concepts for education. The new enhancements support understanding in modern exploratory mathematics, which increasingly employs creative computer environments to:

- encourage motivation in specific topics;
- explore, visualize and demonstrate relations and dependencies;
- to simulate and model;
- to act as a microworld for discovering in certain topics and as a laboratory for creating and building — e.g., a computer environment which offers a kind of smart paper with proper tools to support certain activities as shown in Figure 2, or popular environments for dynamic geometry, which offer new ways for solving traditional problems, but also completely new types of problems in the same area of explorations and new strategies how to solve them;
- as an environment for solving problems with constraints;
- for testing, etc.

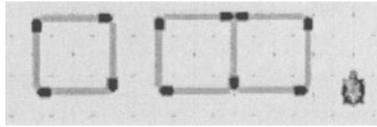


Figure 2. Smart paper with matches for exploring counts of things developed at IoE

The essential properties of IMAGINE help make mathematical objects and relations between them tangible: programmable pictures, direct manipulation tools, events, object oriented structure, and parallel independent processes, as shown in Figure 2.

1.1 Programmable pictures

Among the most interesting and innovative properties of IMAGINE is the possibility to specify shapes of turtles directly by simple Logo commands. In Kalas and Blaho (2000) we presented a broad palette of possibilities offered by this property for educational environments. All shapes specified in this way are independent of the absolute frame and thus in harmony with the traditional Logo philosophy. All programmable pictures stick to the relative frame and IMAGINE rotates them according to actual headings of turtles. Being turtles, such visual objects may easily move along the screen, change their headings, colours, sizes, speeds, etc.

We can use, for example, a well known and simple drawing of a thick red square as a description of the turtle's shape:

```
? setShape [setPenWidth 7
            setPenColour "red
            repeat 4 [fd 40 rt 90]]
```

After doing this, we have no problem with rotations of the square. It will be created automatically and online by IMAGINE itself. We can specify complex shapes which may evolve dynamically and express various mathematical objects.

What will we gain from the possibilities? Since we have used simple Logo commands to specify the shape of a turtle, we get three rewards in return: (a) the shape can be specified in exactly the same language, which we have used so far for drawing pictures; (b) we will get the shape we want — the shape can be modified any time (even while running the project); (c) this object is a Logo turtle with all related rewards such as: Easily move along the screen (or page) either by simple Logo commands like `fd`, `bk`, `setPos` etc., or by being dragged. It can be easily rotated by simple Logo commands `right`, `left` and `setHeading`, and its shape can contain lines of any colours and

widths, which we set by `setPenColour`, `setPenWidth` etc., as far as we specify the shape as if drawing it by traditional turtle pen.

1.2 Direct Manipulations

Every modern programming environment for education strives to provide its users with powerful tools for direct manipulations so that young learners, students and teachers can do a lot without describing their steps through instructions of the language itself. Although IMAGINE successfully lifted the ceiling of direct manipulations from SuperLogo, its previous Logo version (Blaho, Kalas and Matusova 1994; Blaho and Kalas 1998), both IMAGINE and SuperLogo keep the same rule: Whatever can be done through direct manipulations, can exactly be expressed in Logo instructions.

Direct manipulations help us in exploring visible mathematics in two distinct ways. Either they are built in more complex application to help the learner to interact with it by clicking, dragging, giving voice commands, communicating through dialog boxes, etc. or they support the developer himself/herself so that he/she can efficiently create and modify the IMAGINE application. Properties, which help in both ways, are, for example, creating clones of objects. We can right click the object – either a simple or very complex one – and copy it into Clipboard. If we then right click into any page or any pane and choose Paste from Clipboard command, we create a clone as shown in Figure 3.

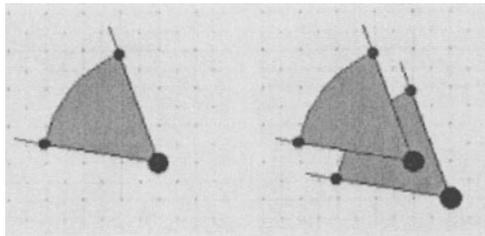


Figure 3. Clones of objects created by cut and paste through Clipboard

The page itself is an object with its own settings like background colour and background picture. If we right click the page and choose the Load background... command, the background picture can be specified, for example by a bitmap to be tiled all over the page. In this way, a useful grid of points can easily be created in the page as shown in the background of Figure 3.

1.3 Events

Each event has a name and a reaction (the body of the event) and belongs to an object. The name specifies the condition of the event, for example `onClick` – meaning that the mouse has clicked the object. The reaction of an event is a list of Logo instructions. When run by IMAGINE, it behaves in a similar manner to running a procedure with no inputs. Specifying events is a powerful way to incrementally build the behaviour of our microworlds. Although there are many standard events, the following are especially useful when developing microworlds with visualized math concepts:

- `onClick` — if an object owns such event, its reaction specifies what should happen if we click it. It may, for example, jump to another position, change its pen colour, increase its pen width, turn right by 90 degrees, etc. If a page owns such event, its reaction can colour a certain region or create a new object at that position etc.;
- `onLeftDown` — this event analyses the interaction with mouse in more detail. If you click an object, you first press the left mouse button over an object, and then you may drag and finally release the button. `onLeftDown` event specifies what should happen during the first phase of clicking the object. In the IMAGINE mosaic microworld all four sample tiles to the left have `onLeftDown` event, which specifies that a copy (a clone) of itself should be created and should stay here as far as the clicked tile will be dragged away from here and placed somewhere within the mosaic;
- `onDrag`, `onLeftUp`, `onChange`, etc.

1.4 Object-oriented Structure

In Blaho and Kalas (2001) we analyzed the object-oriented challenge for Logo creators and Logo educators. We noticed that for more and more Logo educators it is increasingly appealing to present object features to children. Unlike adults, children have no previous experience with any other (non-object-oriented) paradigm, so no transition is needed for them. Many activities and computer games children play have a kind of “object spirit” where you start by composing the game space out of a library of elementary parts and behaviours. We showed children how to use their object to develop problem solving. The object-oriented metaphor keeps the project more transparent and well structured, whenever:

- you need many identical objects with identical behaviours which won’t be modified while running the project. In such case, clones of a prototype or several prototypes can easily be created as shown in the angles in Figure 3;

- you need many similar objects with minor individual differences, which may evolve while running the project. In such case, it may prove useful to specify classes (families) of similar objects (instances). The complete behaviour of each object results from combining common parts with private pieces. Whenever a new instance of a class is created, it simply gets its shape, position, heading and other elements of its behaviour.

In many IMAGINE projects we make use of a full object-oriented metaphor for creating generators of objects with complex behaviour. Such generators (like a tool button for adding another match or adding a new fraction, a new circle or a new cog wheel) may help the user both in his/her exploring visible math and understanding an object approach to problem solving as well.

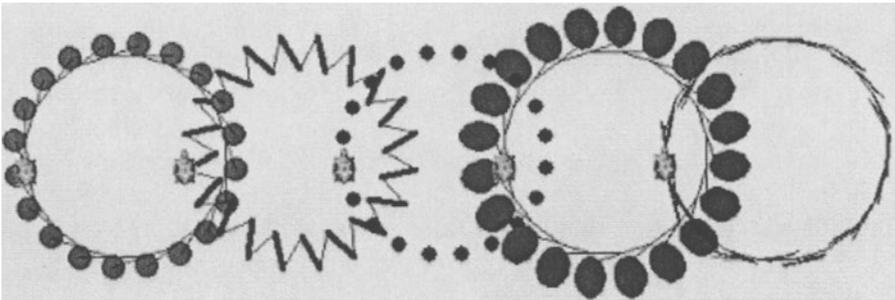


Figure 4. Five turtles above are drawing the same circle, each in its own way

Another powerful and useful object-oriented property is the possibility to make turtles interpret the same commands (even primitive ones) in different ways. Figure 4 illustrates how five turtles with the redefined `fd` command will draw a circle by `repeat 20 [fd 20 rt 18]`. The possibility to have private versions of any procedures may result in interesting effects like drawing in different symmetries, different transformations, sizes, colours, etc.

1.5 Parallel Processes

In IMAGINE we can launch several parallel processes. They run independently and can be used, for example, to:

- watch over relations and dependencies. In a dynamic geometry microworld, if we drag a point, there is probably a process (or several processes), which updates the shapes and positions of all dependant lines and circles. These changes themselves may trigger other processes to update objects;

- make objects regularly move forever, like planets rotating around a sun;
- make objects follow paths or curves described by lists of points or vectors;
- run various experiments, simulations, gadgets, dynamic models, etc.

1.6 Support From IMAGINE Data Structures

IMAGINE data structures (like lists of points $[[x\ y] \dots]$) present powerful support for developers of educational applications (for math or other purposes). They are used for representing paths, lines, outlines or borders of areas. Even more flexible are lists of vectors, which are useful for transformations and computations with paths and curves.

2. SELECTION OF ACTIVITIES IN VISUAL MATH

Now we want to offer a selection of well-known visible math activities and match them to properties and techniques of IMAGINE, which makes it possible to implement them.

2.1 Constructing mosaics and exploring shapes

Regular shapes like squares, rectangles, triangles and other polygons, tiles, sticks, matches, cubes, etc. can be easily created by using programmable pictures. They will facilitate development of activities such as:

- Mosaics for exploring areas, symmetries and similarities. Tangrams and other activities/puzzles/games for exploring geometric shapes such as searching for all triangles/squares/polygons in a picture;
- 3-D mosaics, where pieces are identical cubes. They may be used for exploring volumes, building a structure by carrying out given instructions etc., exploring and understanding space.

When working with mosaics and shapes, we profit from automatic rotations of programmable pictures, an easy way to keep all pieces within a grid of regular points (either visible or hidden).

2.2 Exploring numbers, number sequences, using number rods and number lines, making use of different representations and visualizations of numbers

Programmable pictures may look like small cards with digits, numbers, signs of arithmetic operations, coins, bank notes, etc. In teaching/learning math, number rods or segments or figures of various heights are often used. In IMAGINE we can develop similar activities such as:

- Comparing, adding and sorting number rods according to their length;
- Experimenting on the number line. There may be cards with numbers attached to the line or certain animated characters, which can make steps to the left or right to visualize results of arithmetic operations.

When developing such activities in IMAGINE, we gain from direct manipulations like dragging and multiplying objects by cloning.

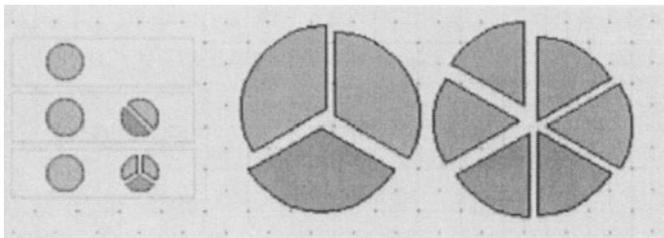


Figure 5. Playing with fractions: Dividing, comparing and adding

2.3 Exploring fractions

Figure 5 illustrates visual interactive environment for constructing, comparing and adding fractions.

2.4 Angles

Figure 3 illustrates an interactive environment for exploring angles. Each angle object is highly flexible. It can be dragged, rotated, increased, decreased or resized. It illustrates how powerful are objects (turtles) with programmable shapes and bright events.

2.5 Dynamic geometry

Probably the simplest shapes to create in Logo are points, segments, lines, circles, etc. And yet they present a surprisingly powerful tool and wide room for interesting activities. Dynamic geometry offers a wide palette of problems to be solved from very simple ones to rather complex ones such as:

- Exploring various kinds of symmetries, transformations and similarities;
- exploring distances, perimeters and lengths of borders of various shapes;
- solving simple planar constructions using points, lines and circles.

When such activities are being developed in IMAGINE, we deeply rely on events like `onDrag`, which repeatedly triggers recalculation and redrawing of all dependant objects. The object-oriented metaphor helps to keep the project transparent (with classes of objects like `Circle`, `Point`, `Line`, etc.).

2.6 Experimenting with combinations, statistics and probability

Learners are often expected to construct all possible combinations and permutations, and only later they encounter corresponding mathematical formulas. In IMAGINE any turtle can be assigned any bitmap as its shape. Combining this option with programmable shapes, we can build various basic elements for activities with combinations and permutations.

When exploring elementary statistics and probability in IMAGINE, we extensively make use of repeated computations and parallel processes for running simulations and experiments. While running them, the project records certain values and visualizes them instantly in one or another way.

2.7 Solving and constructing puzzles and labyrinths

Thanks to programmable shapes we can build pieces of certain mechanisms. The task is to combine several pieces into a functional unit. An example is the cogwheel microworld, where the final behaviour depends on the number of cogs of each wheel and the way we combine them. Programmable shapes and parallel processes are very helpful here.

There are many famous and popular puzzles in mathematics in which the task is either to move pieces respecting certain rules (e.g., game 15) or join pieces together so that they form certain shape (e.g., Pentamino). In this group of activities we should also mention mazes and labyrinths (2-D or 3-D) in which we usually have to find the shortest path or all possible paths, etc. When developing the activities in IMAGINE, we profit from the ease with which we rotate programmable shapes of objects and drag them.

2.8 Measuring, comparing and estimating

Another rich pool of activities originates from comparing, sorting, measuring, arranging and rearranging. Children can:

- sort pieces or any objects into groups of related things (based on their shapes, colours, size, meaning or functionality);
- explore and compare graphs of mathematical functions;
- experiment with a traditional watch with hands (adding or subtracting hours, halves and quarters of hours, minutes, study angles between hands, etc.).

Beside programmable shapes, parallel processes are broadly used when developing such activities in IMAGINE.

3. CONSIDERATIONS

Obviously, mathematical objects can be visualized and explored in different environments such as Visual Basic, Delphi, etc. However IMAGINE offers properties remarkably helpful for educational purposes:

- A rich environment for wide spectrum of users from learners of preschool age to the creators of educational materials and professional educational applications is provided. Very young children are already able to play with turtles, the very same turtles which are used to visualize complex math concepts.
- Turtle geometry offers well appreciated and elegant means to cope with graphical information.
- IMAGINE (Logo) projects are always open for further development and modifications.
- It is possible to assign either simple or very complex behaviour to its predefined actor/multiple actors (i.e., turtles). Also, we can assign simple or complex visualizations to turtles, still or animated.
- IMAGINE offers many modern programming features such as parallel processes, objects, events, communication through Internet, multimedia, etc.
- Logo itself has a long tradition in education.

When developing IMAGINE itself in previous years, its authors were inspired by Noss' (1997) vision of powerful computational systems which offer novel kinds of symbolism and innovative ways to manipulate mathematical objects. We hope that the result will encourage math teachers and developers to create new visible math educational materials.

REFERENCES

- Blaho, A. and Kalas, I. (1998) Young students and future teachers as passengers on the Logo engine. In D. Tinsley and D. Johnson (eds.) *Information and Communications Technologies in School Mathematics. Proceedings of the IFIP WG3.1 Conference*. Grenoble. London: Chapman and Hall.
- Blaho, A. and Kalas, I. (2001) Object Metaphor Helps Create Simple Logo Projects. *Proceedings of the 8th EuroLogo Conference*. Linz. Wien: Osterreichische Computer Gesellschaft.
- Blaho, A., Kalas, I. and Matusova, M. (1994) Environment for Environments: New Metaphor for Logo. In D. Benzie and J. Wright (eds.) *Exploring a New Partnership: Children, Teachers and Technology. Proceedings of the IFIP WG3.5 Conference*. Philadelphia, PA. Amsterdam: North-Holland.
- Kalas, I. and Blaho, A. (2000) IMAGINE... New generation of Logo: Programmable pictures. In D. Benzie and D. Passey (eds.) *Educational Uses of Information and Communication Technologies. Proceedings of the IFIP World Computer Congress*. Beijing, China. 427 – 430.
- Noss, R. (1997) New Cultures, New Numeracies. An Inaugural Lecture. Institute of Education, University of London.
- Watt, D. (1983) *Learning with Logo*. New York: McGraw-Hill.

BIOGRAPHIES

Ivan Kalas is an associate professor at Comenius University, Bratislava. He is a member of the steering committee of the national Slovak InfoAge project, which has already integrated more than 800 schools in Slovakia. Ivan is the head of the Department of Informatics Education, which is responsible for preservice education of teachers. He is author or co-author of several books and textbooks. Ivan represents Slovakia in IFIP TC3 Committee for education. He has presented papers at dozens of international conferences and workshops. In 2000 he worked as visiting researcher at the Institute of Education in London. He is a member of several international editorial boards and a member of program committees of several national, European and international conferences. He has read several invited lectures in UK, Brazil, USA, Hungary, Poland, Bulgaria, Czech republic and Portugal. He is a member of IFIP's Working Group 3.5.

Andrej Blaho is a senior lecturer at Comenius University where he gives lectures on introductory and advanced programming. He has taken part in several major national and international projects developing interactive educational computer environments for children and students. He is the author or co-author of several books and textbooks. His books on problem solving received the Publisher's Awards in 1991 and in 1999. He is a member of IFIP's Working Group 3.5.

Ivan and Andrej wrote *SuperLogo: Learning by Developing*, which was published by Logotron, UK in 1998 and has been translated into two other languages. They developed several educational software environments, which are being used in more than 20 countries. The tools aim at developing creativity, logical and algorithmic thinking, communication and co-operation. Together with Peter Tomcsanyi they developed SuperLogo and, in 2001, together with Lubo Salanci, they released IMAGINE, a new generation computer tool for education.