Exploratory Learning

Ivan Kalas, Daniela Lehotska

Department of Informatics Education
Faculty of Mathematics, Physics and Informatics
Comenius University
842 48 Bratislava, Slovak Republic
kalas@fmph.uniba.sk
lehotska@fmph.uniba.sk

Abstract: In our department we have a long history of developing interfaces for learning, which are framed by constructivist and constructionist theories of learning. Thus we try to create effective opportunities for communication, exploratory and collaborative learning and working in common learning spaces. We build this specialization upon software platforms like SuperLogo and Imagine Logo (software environments being used in dozens of countries and thousands of schools for educational purposes), which have been creating since 90-ties with the aim to provide children, students, teachers and (professional) educational software developers with intuitive yet powerful means to discover, communicate and cooperate. In our software platforms we are making use of the combination of several modern and innovative features like parallel processes, object-oriented paradigm, events, tools for straightforward on-line communication through the Internet, and programmable shapes – the possibility to specify the actors' (i.e. turtles') shapes by simple Logo commands.

During recent years we have been exploring interesting opportunities emerged from these features, which allow us to create visual interactive objects in many diverse and inventive ways. Actors with shapes specified through programmable shapes may change in accordance with external settings; they may become pieces of interactive "jigsaw puzzle", blocks of a building set inviting children to explore. They may continuously visualize actual states of other objects or relations among them – in such role we call them gauges and use them for increasing the children's motivation and supporting them while children explore, modify or build behaviours in their learning processes.

Gradually we have discovered that visual interactive objects might have diverse shapes and could be used in many effective and attractive ways in working with different age groups – from simple building blocks for pre-schoolers, through flexible objects for different representations of fractions, to complex items of software laboratories, in which future teachers construct their own knowledge and understanding of key informatics concepts.

We will conclude our survey of challenging means for exploratory learning by presenting a few observations obtained by us or our partners from several recent European research projects.

1 Introduction

In our department we have a long history of developing interfaces for learning, such as SuperLogo, Thomash the Clown, Imagine Logo (see [KB03]) and others, often developed or evaluated within European research projects like Playground, Match, CoLabs, Kaleidoscope 6FP NoE with partners from IoE London, Logotron Cambridge, IoE Warwick, ELTE Budapest, Cnotinfor Portugal and others. Our interfaces for learning are framed by constructivist and constructionist theories of learning. We share Piaget's understanding that the knowledge can't be transferred from one person to another. It is constructed by an individual through his/her own experience. "To understand is to discover. A student who achieves certain knowledge through free investigation and spontaneous effort will be able to retain it" [Pi73].

In development of our environments for learning we try to create effective opportunities for communication, exploratory and collaborative learning and working in common learning spaces. That is why we are interested both in cognitive oriented and socially oriented constructivism which stresses the collaboratory efforts of learners as sources of learning.

Most of all, however, we are influenced and motivated by the work of Papert who explored the theory of constructivism and went beyond it. He calls his theory constructionism as the opposite of instructionism. Papert claims, that constructivist learning happens when people are engaged in constructing a product, something external to themselves such as a sand castle, a machine, a computer program or a book. In parallel with constructing the things children construct their knowledge. In Papert's view "better learning will not come from finding better ways for the teacher to instruct, but from giving the learner better opportunities to construct". He considers computer to be the medium for development of constructionistic learning – in [Pa93] he speaks about The Knowledge Machine which would allow children a rich exploration of the world.

Constructionism and exploratory learning have been our main visions also in several recently completed research and development projects. There we have been using Imagine Logo platform for developing and studying means which would allow children to discover, communicate and cooperate. We have been rather successful in reaching these goals mainly due to effective and productive combination of several modern and innovative features of Imagine Logo – we will briefly present them in chapter 2. After that we will show that interfaces for learning can be employed in many effective and attractive ways in working with different age groups. To prove this we will put forward outcomes of several projects: the gauges of the Space Travel Games Construction Kit project [Ka06a] (in chapter 3), the interfaces for collaborative learning of the CoLabs project (in chapter 4), interesting dynamic mathematics environment of Visual Fractions (in chapter 5), and a series of interactive Imagine Logo microworlds from experimental seminar for our future teachers. We are using these environments to provide our students with tangible experience through which they construct their own knowledge and understanding of many key informatics concepts (see chapter 6). In the closing chapter 7 we present some interesting observations from the research with the interfaces for exploratory learning, conducted by our European partners or by us.

2 Imagine Logo Platform for Our Research and Development

Imagine Logo represents new generation of educational and professional programming environment, which provides users with powerful and modern techniques like object-oriented paradigm and on-line cooperation through the Internet. This approach makes Imagine Logo an appropriate learning environment not only for children, but for teachers and professional educational software developers as well. It is becoming a widely-used tool in European educational systems for prototyping and building active learning environments, educational microworlds or stand-alone applications, and for educational research as well. Beside traditional elements of turtle geometry it offers several new attractive features and concepts for education. In [BK01] and [KB03] we have explored whether and how these new enhancements support understanding in modern exploratory learning. In our research projects we make use of Imagine Logo as a tool for quick and elegant development of interactive interfaces for learning. The most important features, which allow us to do so, are:

- Events and parallel processes – beyond all doubt they represent new approach in the development of interactive interfaces for learning. In rather natural and powerful way they support what we develop – either with children or for children: in many iterative design loops we build actors or objects, we build their shapes, behaviours, interactions with the environment or between each other. Parallel processes can be considered as forces, which independently and in parallel watch over relations and dependencies among actors, make them move, make objects follow certain paths or curves, run various experiments, simulations, gadgets, dynamic models etc.
- Multiple turtles and object-oriented structure – nearly unlimited numbers of actors (that can be organized into "families" of similar objects with related or identical behaviours) help us construct scenes with dozens or hundreds of actors of any kind. These actors (or any active pieces) may be dynamically created, may react to outer circumstances, alter their reactions, become models for new clones, may cease to exist etc.
- Programmable shapes – the possibility to specify actors' shapes (or any picture, still or animated) directly by simple Logo commands. In [KB00] and [KB03] we presented broad palette of possibilities offered by this innovative feature of Imagine Logo. Note that all shapes specified in this way are independent of the absolute frame. In harmony with the traditional Logo philosophy, all programmable pictures stick to the relative frame and as such Imagine itself 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.

In an extended process during recent years we have been deeply exploring the possibility to represent the actor's (i.e. turtle's) shape through a piece of simple Logo commands. We have considered the ways how this feature – combined with event-driven programming, object-oriented structure and parallel independent processes – qualifies Imagine as

a powerful development platform for computational systems aimed at representing, exploring and manipulating visual interactive objects in diverse innovative ways.
In [KB03] we presented several simple examples of how to turn common Logo turtles into squares, arrows, cogwheels, numbers, angles or fraction parts of different units. Inside the “setShape [...]” command brackets we simply insert usual Logo commands which would otherwise draw the same shape into the background.

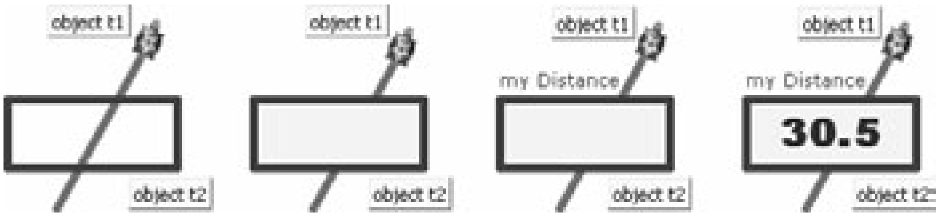
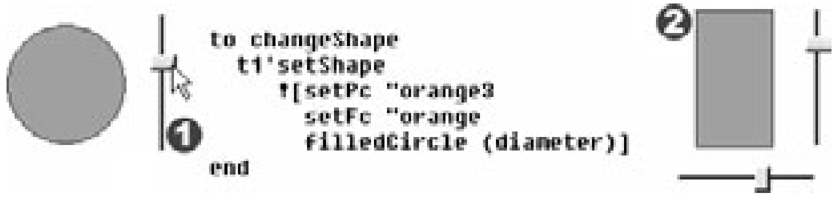


Fig 1: Turtle t1 draws a line in the background (to clarify the difference), while turtle t2 “draws” its own shape by “setShape” using the same language. Any turtle created in such way can move and rotate – its shape isn’t drawn into the background

3 Imagine Logo Turtles as Gauges

Shapes of Logo actors (i.e. turtles) which are specified in such programmable way may change in accordance with some external settings. Figure 2 for example illustrates a situation, when slider 1 has an event attached to it, which will – whenever triggered by resetting the slider – address object t1 and make it accept the actual value of the slider as its actual diameter and update its shape correspondingly. Naturally, there may be several external factors which dynamically trigger the process of updating a shape.



```
to changeShape
  t1'setShape
     ![setPc "orange3
       setFc "orange
       filledCircle (diameter)]
end
```

Fig 2: Rectangle 2 will dynamically change its width and height according to a pair of attached sliders

Objects which are parts of the environment may employ their programmable shapes to look like numbers, cards with words or sentences, fractions, ... simply any building items appropriate for the topic of the concrete environment for exploratory learning. Such object:

- is a common Logo turtle, which means it can easily be moved and rotated – without any need to think about how to specify its newly rotated shape,
- can contain its own internal information, its own private data, state and attributes,

- can have shape which dynamically changes depending on its actual state (for example, its actual position) or in accordance with some external factors (for example variables stored in the page, attributes of some other objects etc.). Using the mechanism of user-defined events, any change of any of those factors may trigger forward propagation to update all dependant shapes, see [To03],

- can employ the mechanism of parallel independent processes to – besides reacting to any external changes – continuously execute some other operations or movements. An object can thus monitor some other values (not affecting its own shape) or relations, it can move, change its size, colours etc.,

- can contain any number of sensitive spots which differently react to clicking or dragging by the mouse. For example, if we drag red point 1 of the angle, see Figure 3, its internal setting size will change and thus the shape of the angle will change as well. If we drag its blue point 2 the whole angle will rotate around its central point 3. If we drag the perimeter 4 of the angle, we can increase or decrease its radius. Moreover, any angle object can be either transparent or coloured. This example demonstrates how flexible and interactive these objects are.

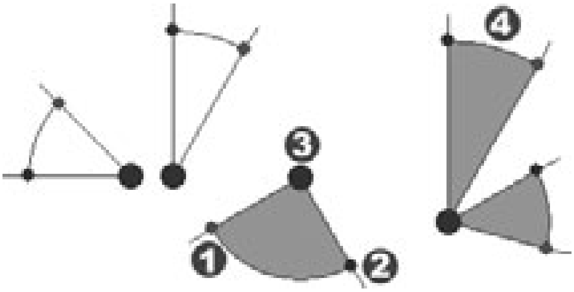


Fig 3: We extensively exploit programmable shapes for representing mathematical objects or building blocks in various exploratory microworlds

In the Space Travel Games Construction Kit project, see [Ka06a], we have started to engage such dedicated objects neither for building nor for modifying the behaviour of any actor or tool – their purpose has only been to continuously visualize (through their shapes) actual attributes of the things involved. We started to call such objects gauges. Observations reported by Kahn et al. prove that gauges can play important role in increasing the children's motivation and support them while exploring, modifying and building behaviours in their learning process.

One of the challenges in the Space Travel Games Construction Kit is to make the Lander safely land on the surface of a planet. To manage this, children explore, apply and modify behaviour gadgets (pieces of behaviour encoded in instructions) and visual gauges that allow them to monitor several key quantities of the Lander. In Figure 4 we see the gauges panel (A) with icons for three different models of gauges (B). If we decide to use one of them, we click it and drag its instance (C) into the working space.

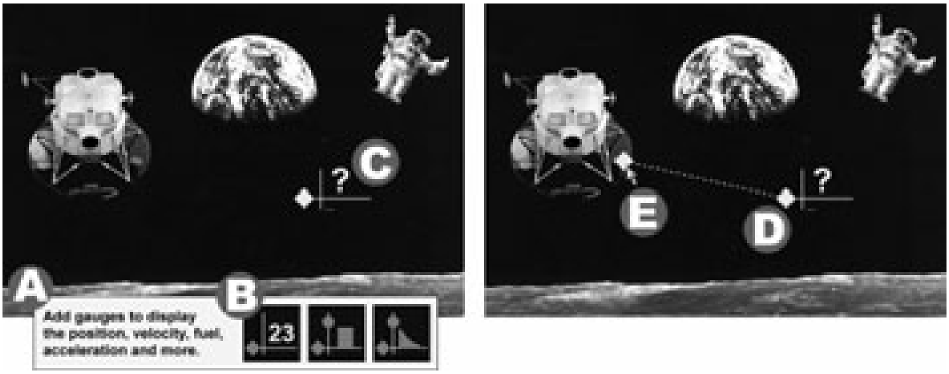

A
B
C
?
Add gauges to display the position, velocity, fuel, acceleration and more.
23
?
E
D

Fig 4: While the gauge C isn't activated, it shows as a default value a question mark – any other initial value could be confusing

To activate the gauge we have to drag its anchor (D) towards an object we want to explore, the Lunar Lander (see E) in this case. When we release the anchor, the object displays the menu (F) of all different values it exports to the gauge, see Figure 5, and thus allows us to get into its internal state in the most readable way.

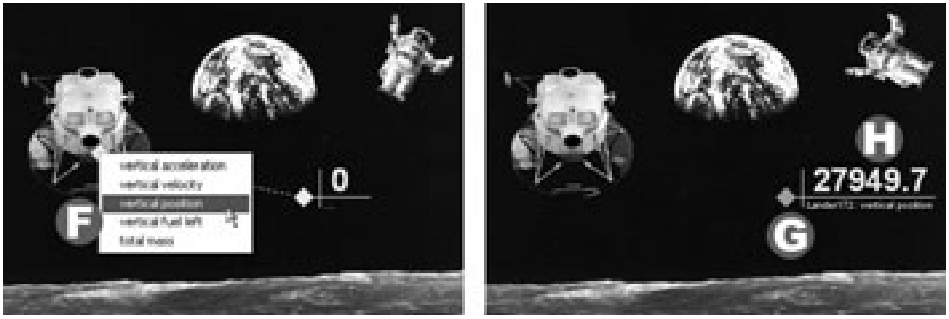


Fig 5: When the gauge is activated, the colour of its anchor alters (see G) and the gauge starts to continuously display the actual value of a chosen variable (see H). The gauge also displays the name of the object and the variable currently monitored

The gauges showed in Figure 6 have some additional control elements: circle 1 can be dragged vertically – in this way we can change the scale of all collected values, for example, between f(x)/1 and f(x)/10. Similar extensions and modifications of gauges are limitless, which results from the fact they are specified in ordinary Logo commands. Their look, colours, sizes and any additional control elements are designed by the author of the microworld, see Figure 6. Designing them is similar to drawing them in traditional Logo turtle way. The gauge in the right part of the figure, for example, contains four anchors to monitor four different quantities in parallel.

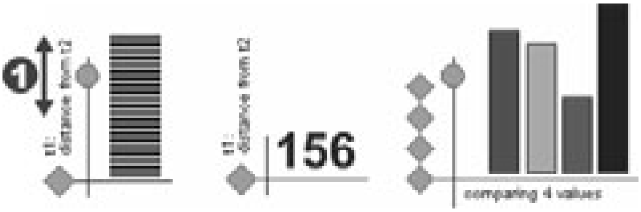


Fig 6: Gauges may have very different shapes

Based on the educational projects mentioned above and research conducted within them we may specify benefits and properties of engaging gauges in the interfaces for learning:

- Gauges are visual and highly interactive objects and thus are useful tools to observe dynamic processes, changing states, relations and quantities.
- They allow us to access areas of the environments (microworlds, interfaces etc.), which are due to different reasons hard to access or completely hidden. It is the designer of the environment who decides which attributes or quantities will be made available in which context.
- Their styles, sizes, colours etc. are relatively easy to change by the designer (or even by the user) as far as these are basic Logo objects with shapes specified by the Logo commands. Advanced user or designer may either modify them or build new gauges with different look.
- They are highly flexible tools which may contain several additional control elements for customizing the visualization, for example the size, scale, units of measurement, frequency etc.
- They can be employed in different layers of learning in the sense of [Ka06a], namely at the first layer for reading and observing, at the second and third layers for manipulating, predicting, testing and active exploring and at the fourth layer as a means for projecting modifications or new development within the environment.
- Gauges can be used in different architectures: naturally they would fit into component-oriented systems (see [KKH97]) or in less modular systems where they may provide the probe into private internal states of other objects.
- As reported by Kahn, gauges support collaborative learning (see below).
- Gauges support employment of multiple representations in parallel. We have managed to benefit considerably from this property in our Visual Fractions.

4 Visual Fractions

Exploratory learning has been our main vision also in recently completed CoLabs project. Our aim has been to develop educational microworlds for collaborative work and learning – collaboratories. As one such collaboratory we decided to create a complex dynamic interactive computer environment, which would allow children to explore and discover fractions and fractional relations by themselves, or with peers and teachers. We wanted our collaboratory to offer a set of powerful jigsaw puzzle pieces for building one's own understanding of the topic.

Visual Fractions provides ten different visual representations: pie, box, decimal fraction, percentage, ratio, picture, family, number line, fraction and balloon. Fraction objects can be connected together to create dependences, which means that some objects represent values of some other objects. These dependencies are dynamic – we can change the value of an independent object and consecutively the value of a dependent object will change. These dynamic dependencies and interactions give us opportunity to observe how a fraction represented in one way looks like in another representation, what happens if we change the value of some object, investigate when and find the reasons why a fraction represented in one way cannot be expressed in another way etc. The relations between objects can also be more complex – more objects can depend upon the same object. Also the chains of dependencies can be created.

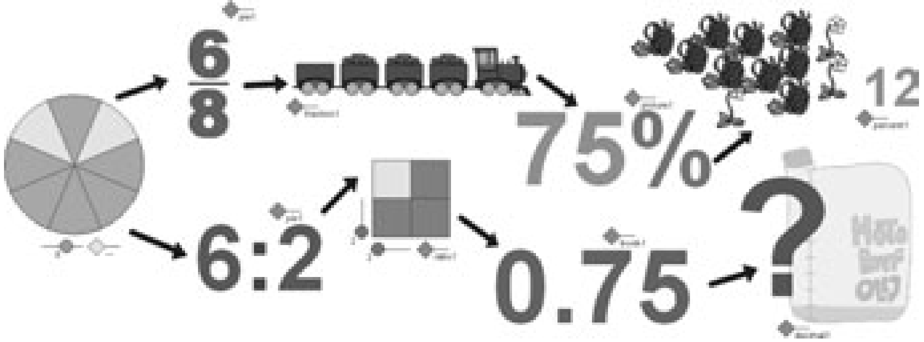


Fig 7: Two chains of dependencies of fraction objects staring with a pie with the value of 6/8. Fractions represented by a picture (a train, an oil canister...) have fixed denominator (for the train in this figure it is 4, for the oil canister it is 10). Therefore they sometimes cannot represent a value of another fraction, e.g. we cannot represent 6/8 by a canister with 10 segments

For each fraction object several properties can be set up. In some representations we can determine which control gadgets of the object will be presented to the user. Some other settings are specific mathematical properties. E.g. pie, box and fraction representations have a property that determines whether the object should always be reduced to the smallest denominator.

Beside fraction representations there are other useful tools in Visual Fractions including basic arithmetic operations (addition, subtraction etc.), operations for comparing (less, equal, greater than), regions, generators, textboxes and components for representing True and False values. These tools work on the principle of dependencies as well. For example, regions are used to find out whether groups of objects meet certain condition, like whether all objects within the region are less than (equal, greater than) the region’s leading value, whether all objects in the region are identical, true, ordered, whether their sum equals the region’s leading value. With generators we can create sets of values that are either randomly or in the specified order used as input values (parameters) to an exploratory task.

Since the beginning of the Visual Fractions development we have planned to support different forms of its usage in the learning process. We consider the situation when a pupil solves pre-prepared activities the least innovative style of learning, the least challenge to collaborative learning and constructivist explorations. At the opposite pole to such learning situation is a group of pupils working in a common learning space – either at their computers or in front of a large interactive smart board, where together with their teacher they solve problems and discover relations using additional blocks of the fractions building set and thus explore mathematics in a more efficient way. They can build dynamic fraction models with dependencies; they can modify the items' settings and observe the effects. They can construct experiments about fractional relations; they can discuss the ways how the models behave. The teacher may stimulate the explorations by questions like What would happen if... or How to make it evident that... or How could we explain it or demonstrate it to others? In such learning situations the distinctions between users of activities and authors of activities are disappearing. Solving problems in such constructionist way means explore relations between objects, modify their representations, add new blocks and insert them into the chains of dependencies and discuss the consequences.

5 Developing Interfaces for Collaborative Learning

In our research we explore what Imagine Logo offers to target groups of users who expect small, immersive, open and flexible microworlds supporting on-line collaboration – developed for everyday learning situations by the means that are accessible also to non-professional developers of educational software. If we examine the development of Imagine microworlds from the on-line collaboration point of view, we can distinguish four dimensions of this process:

- Technicalities – copes with the low level communication through the TCP/IP protocol to run the transfer of data between Imagine kernel and the network. It also provides complete or partial support for overcoming linguistic barriers etc.
- Connection interface – is responsible for creating and activating the connection between two or several computers.
- Sharing – covers higher network programming (communication) by exploiting already running connection between computers. It is responsible for sending and receiving all kinds of information and objects, and thus creating and managing common work space, characters and behaviours.
- Activity – implements the educational goal through designing and programming the microworld itself (covers all programming except network communication).

In [KW06] we have analyzed the development process along each of these dimensions; here we will concentrate on building connection interface only. The tasks being solved here include creating and activating an on-line connection between two or several computers with running Imagine (through local network or Internet). Based on our previous experience in building such interfaces we have developed a metaphor, which hides all technical details and concepts of networking such as server, client etc. and makes it feasible to understand and carry out the following steps.

To establish and maintain such connection the Imagine "Net" class of objects is used. An object (instance) of that class represents a connection for all participants of the activity, i.e. two or several computers. One of the participants – let us call him an initiator of activity – creates a Net object and activates it (steps 1 and 2, see Figure 8). We may consider the initiator as the one who invites us to join.

Other user (or users) must know that there is an open invitation and must react by taking the following steps: he creates a "Net" object as well and accepts the invitation (steps 1 and 3; step 1 is identical for all, step 3 can be run only after step 2). This is similar to making a phone call: step 1 means "have a mobile phone", step 2 "switch it on". Step 3 means either "dial the number" of a friend (i.e. enter his IP address) or "look up his name in the address book" and press it (i.e. enter the other user computer's name).

As soon as the active connection is established, all differences among participants (who was inviting, who accepted) can be forgotten – all of them become equal partners (as if connected in the peer-to-peer private network, although the developer must understand the difference between the initiator-server and other users-clients). When creating the connection (step 1) each user may specify his own nickname, which will later be used by other users to address him. When the connection is activated and established, the users will not need any technicalities (like IP address) any more.

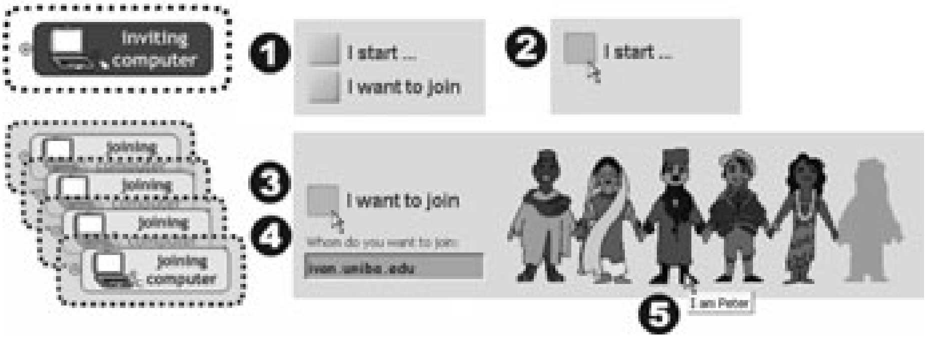


Fig 8: Sample interface for creating 1-team, identical-roles, up-to-6 users connection

The key motive for us why we continually study collaboration through Logo on-line microworlds is the ease with which we achieve high level of interactivity, visualization and openness. By openness of an educational application we understand its readiness to open the inside to its user and easily accommodate modifications of its internal resources. We distinguish openness (a) at the level of data – Can pictures, sounds, backgrounds or other data be accessed and modified or replaced?, (b) at the level of application didactics – Are there more ways how to use it?, and (c) at the level of tasks – Can other tasks or other type of tasks be inserted to fit into the same didactics? Net connections of Imagine Logo lead us even further than most of other Logo versions. It allows, for example, constructing our own version of forwarding, then publishing and sharing it with all other users – in the form of shared and open piece of code, shared behaviour or shared procedure.

6 Exploratory Learning in Our Future Teachers Training

The potential of straightforward development of visual interactive objects can easily find space for creative integration into surprisingly wide spectrum of situations and ages of learners – as illustrated by our innovative seminar on Discovering Basics of Informatics for future teachers of informatics. During 12 weeks we pass through series of interesting and complex topics, however, we treat them in a completely informal way encouraging modern constructionist learning. To achieve this we are developing small interactive software interfaces for exploratory learning, devoted to one or another key issue of informatics, for an example see Figure 9.

For our students the basics of informatics are hard and often unpopular domain. When we tried to analyze the reasons for their negative attitude, we came to the following conclusion: Because of the shortage of time the students are exposed to key issues and problems before they manage to discover those problems by themselves. We present to students formal analysis of the fundamental problems which aren't their fundamental problems yet. Moreover, we do so using language which is rather unfamiliar to them.

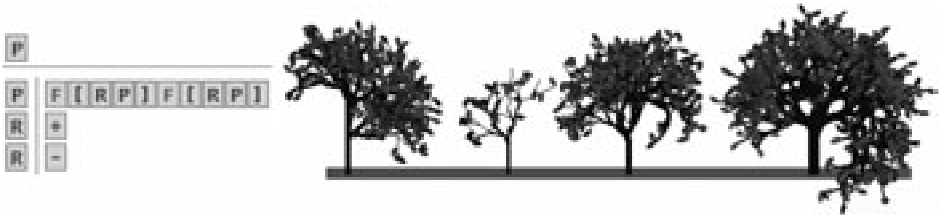


Fig 9: When developing software applications for our seminar we provide students with constructivist interfaces for exploratory learning. This figure shows an application, in which students construct L-systems, have them generate words and interpret them in a Logo turtle way

To improve this situation we want our students to obtain concrete experience within several key issues like constructing logic circuits, modifying them and exploring corresponding Boolean functions – before they start formal treatment of problems with minimization of DNF expressions etc. We want them to play with regular expressions before they start proving theorems about them, before they start talking about them using yet another formal language etc. Although it is too early to draw any serious conclusions about students' attitude to the basics of informatics, in [Ka06b] we could already report about how this innovation has accelerated students' interest in exploratory learning.

7 Some Interesting Observations

"Collaborative learning is a category of learning by cooperating, common exploring and discovering, creating solutions and solving problems, discussing and improving together", see [Wi00]. New digital technologies play unique role in such learning. Therefore, it is natural that one of the key challenges in the development of interfaces for learning at our department is the enhancement of collaborative exploratory learning.

By referring to several educational design projects we have illustrated a powerful idea of how to employ innovative features of Imagine Logo to support exploratory and collaborative learning. We (either alone or together with our partners in the projects mentioned above) have conducted research activities, which show interesting results. While testing the Space Travel Games Construction Kit, for example, gauges proved very successful in two player game – the ability to attach gauges to the opponent's Lander was a particularly successful feature, enabling one group to make a close comparison with the other and to adjust the strategy second by second, see [Ka06a].
Several experiments have been run during the development of the Visual Fractions at the universities of Warwick, Budapest and Bratislava, inspecting affordances of the interface, functional design of the tools, fraction representations and their control items – how and when were they used by children, how intuitive they are etc. In our own experiments with teachers and children we have come to rather interesting observation. Even if Visual Fractions is seemingly non-programming environment, using it in a constructionist way develops (and requires) certain level of algorithmic thinking. When teachers and children use it as authors – building models, exploring relations etc. – they develop their algorithmic skills in several aspects, see [LK05].

Bibliography

[BK01] Blaho, A.; Kalas. I.: Object Metaphor Helps Create Simple Logo Projects, Proc. of EuroLogo 2001, A Turtle Odyssey, Linz 2001, pp. 55 – 65

[Ka06a] Kahn, K.; Noss, R.; Hoyles, C.; Jones, D.: Designing for diversity through web-based layered learning: a Prototype Space Travel Games Construction Kit, CD Proc of 17th ICMI Study Conference Technology Revisited, HUT, Vietnam 2006

[Ka06b] Kalas, I.: Discovering informatics fundamentals trough interactive interfaces for learning. In: Informatics Education – The Bridge between Using and Understanding Computers. Berlin, Springer-Verlag LNCS 4226, 2006. pp. 13 – 24

[KB00] Kalas, I.; Blaho, A.: Imagine New Generation of Logo: Programmable Pictures, Proc. of WCC2000, Educational Uses of ICT, Beijing 2000, pp. 427 − 430

[KB03] Kalas, I.; Blaho, A.: Exploring visible mathematics with Imagine: Building new mathematical cultures with a powerful computational system. In: Marshall, G., Katz, Y.: Learning in School, Home and Community. IFIP Kluwer 2003, pp. 53 − 64

[KKH97]Kynigos, Ch.; Koutlis, M.; Hadzilacos, T.: Mathematics with component-oriented exploratory software, Int J of Computers for Mathematical Learning, Springer, Vol 2, No 3, 1997, pp. 229 – 250

[KW06] Kalas, I.; Winczer, M.: Building interfaces for on-line collaborative learning, Int. J Education and Information Technologies, 2006; 11 (3), pp. 371 – 384

[LK05] Lehotska, D.; Kalas, I.: LVF – Interface for Dynamic Mathematics, Proc of the 7th Int Conference on Technology in Mathematics Teaching. Bristol 2005, pp. 108 − 116

[Pa93] Papert, S.: The Children's Machine. Rethinking School in the Age of the Computer. Basic Books, 1993, New York.

[Pi73] Piaget, J.; Memory and intelligence: New York: BasicBooks.

[To03] Tomcsanyi, P.: Implementing object dependencies in Imagine Logo, Proc. of EuroLogo 2003, Porto, pp. 127 − 140

[Wi00] Wieserma, N.: How does Collaborative Learning actually work in a classroom and how do students react to it? Retrieved June 6, 2007, from www.city.londonmet.ac.uk/deliberations/collab.learning/wiersema.html