

---

# Practical Logo Team Competitions

Irina Kuznetsova, [info640@spb.edu.ru](mailto:info640@spb.edu.ru)

Sophia Gorlitskaya, [sophiag@yandex.ru](mailto:sophiag@yandex.ru)

Saint-Petersburg City Palace of Youth "Anichkov Palace", <http://anichkov.testing.spb.ru/>,  
Secondary School 640 of general education, Saint-Petersburg, Russia

## Abstract

A school team programming contest is, on the one hand, an exciting game that allows us to develop and apply our intellectual and team-working skills. On the other hand, it is a very efficient way to improve our understanding of the best ways to teach programming and technology in schools.

School and university programming contests have a long lasting tradition in Russia. For many years, Russian teams successfully participate in international programming competitions.

In 1996-1998 Saint-Petersburg Gymnasium 470 organized annual summer schools for students interested in programming and computer science in general. The summer schools welcomed students from Saint-Petersburg, Murmansk, Syktyvkar, Moscow, and other cities. In the course of the summer schools, Logo was proposed as a primary programming language.

In organizing the summer schools, we have been trying to follow the principle formulated by Seymour Papert [1]: to provide students with an environment to realize their ideas, without putting rigid bounds on their fantasy. While a traditional course in informatics was not able to give students enough time and freedom to implement their algorithmic ideas, our summer schools allowed every participant to develop her own systems project and to gain enough background in Logo programming to work independently.

Using the experience we gained in organizing the summer schools, in 1997 we launched Team Olympiads in Practical Logo Programming.

Many Russian schools accepted Logo as the first language for teaching programming. In 2005-2006, the Ministry of Education of Russian Federation officially recommended installing Logo systems in schools, starting from MicroWorlds 2.0. Nevertheless, many teachers of informatics still considered Logo as a programming language bound to encoding simple commands for controlling the Turtle and various peripheral devices. It is often ignored that Logo enables all modern programming techniques, and environments like MicroWorlds 2.0 and especially MicroWorlds 3.0 provide instruments for a wide spectrum of various scientific studies and modeling. Moreover, these instruments are accessible for students of all levels and backgrounds: from primary school kids to doctors of sciences.

In this paper, we summarize the ten years experience we earned while organizing Annual Team Competitions in Practical Logo Programming (Team Olympiads) for students of 5th-8th grades that took place in Saint-Petersburg, Russia in 1997-2007.

We classify around 70 problems that were designed for the Olympiads, and we discuss important features related to formulating and solving Olympiad problems in Logo. We show how these features highlight principal difficulties that arise in teaching the theory and practice of computer science and technology in primary and secondary schools. By presenting examples of Olympiad problems, we also demonstrate the impressive diversity of technological methods that can be applied in the Logo environment.

## Keywords

Logo, school informatics, pedagogical aspects in programming, object approach in programming, computer modeling

## Introduction

In 1996-1998 Saint-Petersburg Gymnasium 470 organized annual summer schools for students interested in programming and computer science in general. The summer schools welcomed students from Saint-Petersburg, Murmansk, Syktyvkar, Moscow, and other cities. In the course of the summer schools, Logo was proposed as a primary programming language.

Up-to-date Logo environments are characterized by the following features:

- *Object approach.* Object approach is employed (implicitly or explicitly) in all Logo products: in designing her projects, a user is allowed to elaborate built-in Logo objects, by changing its properties and extending their functionality. As we observed, a positive background in Logo environment helps students successfully apply object and system approach in exploring other programming packages.
- *Interactivity.* A logo project involves a number of *personages* – interacting entities whose behavior is programmed by the user. This turns programming into a difficult but exciting process which efficiently develops algorithmic skills and system way of thinking.
- *Multimedia support.* The use of audio and video effects increases the expressive power of Logo projects.

All these features turn Logo into an excellent environment for scientific studies in schools. That is why we chose Logo as a principal language for our programming contests.

Many Russian schools accepted Logo as the first language for teaching programming. In 2005-2006, the Ministry of Education of Russian Federation officially recommended installing Logo systems in schools, starting from MicroWorlds 2.0. Nevertheless, many teachers of informatics still considered Logo as a programming language bound to encoding simple commands for controlling the Turtle and various peripheral devices. It is often ignored that Logo enables all modern programming techniques, and environments like MicroWorlds 2.0 and especially MicroWorlds 3.0 provide instruments for a wide spectrum of various scientific studies and modeling. Moreover, these instruments are accessible for students of all levels and backgrounds: from primary school kids to doctors of sciences.

In this paper, we summarize the ten years experience we earned while organizing Annual Team Competitions in Logo programming for students of 5-8 grades that took place in Saint-Petersburg, Russia in 1997-2007.

The paper is organized as follows. First we specify the goals and organizational principles that we put at the basis of our Olympiads. Then we describe our methodology in formulating Olympiad problems and defining formats of the problem solutions. Next we propose a rough classification of the Olympiad problems and highlight the principal obstacles that arise in understanding theoretical and practical aspects of computer science. We conclude the paper with general observations about the optimal methodologies in addressing Logo modeling problems.

### Goals and Organizational Principles of Olympiads

As the organizers of the Olympiads, we were primarily aiming at:

- demonstrating the potential of Logo in enabling students of all ages and backgrounds to study and model processes in mathematics, physics, and other sciences
- promoting Logo, as the best educational environment for setting up computer experiments as well as for teaching programming to primary and secondary school students
- advertising programming itself, as an exciting kind of intellectual activity that helps bridging the generation gap in families and schools [1,4]

- identifying students gifted in programming
- establishing informal links between organizations - participants of the Olympiads

Initially, we proposed two editions of Logo as the basic environment in our Olympiads: LogoWriter 3.2 and LogoMiry 2.0 (Russian version of MicroWorlds 2.0 produced by the Moscow Institute of New Educational Technologies). Faced with the increasing popularity of MicroWorlds 2.0 and MicroWorldsPro (LCSI), we decided later to stop using LogoWriter 3.2. However, because of some inconsistencies in released version of LogoMiry 2.0, we had to maintain both options for the Olympiad problems: LogoMiry 2.0 and MicroWorldsPro.

Since 2004 Logo Olympiads are organized in two modes: on-site and remote. In the remote mode, the registered participants first receive the set of basic instructions. Then the problems themselves are published on-line at the pre-determined time.

In 2005, we designed a web interface that allows users to run Logo applications on the computers that do not have Logo environment installed. The interface is based on the freeware MicroWorlds Web Player (LCSI). However, certain restrictions imposed by the web interface complicated the organization format of our Olympiads. As a result, later we decided so far not to use the web interface in the competitions.

The basic principles we have in mind in selecting the Olympiad problems are the following:

- The problems expect every Olympiad participant to actively work for at least 2 hours. The expected amount of work should not be less than what the best participants can potentially perform in the given time frame.
- The very same problems should be accessible for participants of 2-6 grades as well as for participants of 7-9 grades [2].

After the Olympiad a document summarizing the problems and their solutions is published at <http://logo.dtu.spb.ru>. Teachers can use this document in their classes. In the future, we prospect organizing specialized forums and seminars on the lessons learnt from the analysis of the problems and the proposed solutions.

## Defining Problem Statements and Solution Formats

In this section, we summarize the principles we chose to formulate an Olympiad problem and to define an expected format of the problem solutions.

### Using Plain Text

Every problem is stated in natural language text. If necessary, a figure or an animated image may be used to illustrate the problem.

In recent years, many teachers point out the frightening decline in the ability of school students to read and understand written text, preempted by video and audio information. The reading comprehension skills remain however necessary for adequate reception of the cultural inheritance of human kind, not mentioning obtaining higher education or high-qualified job.

We deliberately intend to maintain and improve the consistent reading skills of our students. To facilitate the problem perception, the problems proposed for every Olympiad are unified by a common *story*.

For example, in 2004, the Olympiad participants were expected to solve the imaginary problem of surviving on an unpopulated island.

In 2005, all problems were on studying the motion of a given object (Turtle). A particular class of problems involved the formalism of Johan Gielis [3] that describe simple geometrical objects (a circle, a triangle, a rectangle, etc.) as well as more complicated images resembling natural silhouettes such as starfish, snail, etc.

In 2006 all problems were related to the Magic Market where participants met magicians, mysterious animals, dwarfs, etc. In 2007 every participant was in search of the Treasure hidden by the pirates on an island in the Caribbean Sea (see an example in Figure 1).

The problems are presented on a web server. All information related to the Olympiad remains available long after the Olympiad.

### Variants of Solutions

The solutions can be presented in either of the following ways:

- a) Natural language text
- b) A sequence of annotated commands for Logo command center
- c) An annotated complete program in Logo Procedures Sheet.

Each of these ways brings certain number of credit points.

The participants are encouraged to submit extensive explanations of the ideas of their solutions, in plain text or comments to their programs.

A solution presented in way (c) is considered complete if it contains a description of how the Logo programs are called.

### An Example Problem

*Three ships (Fla, Floo, and Chood) are located in different harbors (see the figure). The captains of the ships come to know about the Treasure hidden on a little island in the Caribbean Sea. The ships simultaneously started sailing and simultaneously arrived at the island. Every ship was sailing directly to the island with a constant speed that did not exceed 5 steps per time tick.*

- (1) *For each ship, determine its maximal allowed speed and the initial distance from its harbor to the island.*
- (2) *Write commands of motion for each ship.*
- (3) *Write procedures of motion for each ship and encompass the procedures in a program.*

Obviously, students of different ages and backgrounds comprehend the problem statement differently. This will be reflected in the proposed solutions. That is why the solutions are evaluated by taking into account the ages of the participants and their abilities to handle the Logo environment.

On the other hand, this is an arithmetic problem, and its solution can be expressed using an arithmetic approach taught in 5<sup>th</sup> and 6<sup>th</sup> school grades.



**Solution (a).** *Let  $s_1$  be the distance between ship Fla and the Island,  $s_2$  be the distance between ship Floo and the Island, and  $s_3$  be the distance between ship Chood and the Island. Let  $s_1$  be larger than both  $s_2$  and  $s_3$ . Since Fla was initially more far away from the Island, it should have had the maximal allowed speed, namely 5 steps per time tick. Thus Fla needed  $t_1 = s_1 / 5$  time ticks to reach the Island. Since the ships reached the Island simultaneously, the speeds of Floo and Chood are, respectively:  $v_2 = s_2 / t_1$ ,  $v_3 = s_3 / t_1$ .*

This solution demonstrates that the participant fully understands the problem and knows how to solve it. If he has enough background in Logo programming, he can complement the solution with the following commands:

**Solution (b).**

*Fla, show distance "Island ; distance between Fla and Island*

*Chood, show distance "Island ; distance between Chood and Island*

*Floo, show distance "Island ; distance between Floo and Island*

These commands return the required distances. The solution demonstrates that the participant is able to work with numbers and perform computations in Logo.

Participants with programming skills may define the parallel processes describing the motions of the three objects. To make the program more general, the solutions should use variables. For example:

**Solution (c).**

*to RASST*

*Island,*

*make "sfla distance "Fla ; distance between Fla and Island*

*make "sfloo distance "Floo ; between Floo and Island*

*make "schood distance "Chood ; between Chood and Island*

*end*

*\* Computing the speeds \**

*to Speeded :fla :floo :chood*

*make "v\_fla 5 ; speed of Fla*

*make "time :sfla / :v\_fla ; time for Fla*

*make "v\_fli :sfloo / :time ; speed of Floo*

*make "v\_chood :schood / :time ; speed of Chood*

*end*

*\* Programming the ships' motion \**

*to plavv :v\_chud :v\_floo :v\_flu*

*Chood, towards "Island repeat :time [fd :v\_chood wait 1]*

*Floo, towards "Island repeat :time [fd :v\_floo wait 1]*

*Fla, towards "Island repeat :time [fd :v\_flu wait 1]*

*end*

*\* Main program \**

*to index*

*task1*

*RASST*

*SPEED :fla :floo :chud*

*plavv :v\_chood :v\_floo :v\_flu*

*end*

**Returning to the Initial State and Saving the Results**

The problems can be tackled in an arbitrary order. Moreover, it is recommended to start solving the problems that are feasible to solve in the given 1.5-2 hours, depending on backgrounds and ages of participants.

Therefore we propose to present the solutions in a certain way which allows us to easily run every task. Once a new task is run, the objects of the previous task should be removed. To

ensure this, we developed special standard procedures for launching tasks, as well as procedures for returning to the initial state.

In the first Olympiads, solutions proposed by the participants were often lost because the procedures of submitting and saving the results were not standardized. Of course, the organizers and coordinators of the Olympiads were trying to help all the participants, especially those of younger ages, to save their solutions in a stable storage, but this did not always work. This is the reason why, since 2003, we include specialized programs for automatic saving of the results in the standard package of Olympiad problems.

In 2006 and 2007 the results we proposed an option to save results in text files. Each time a package with Olympiad problems is accessed, the system automatically creates a collection of files associated with the team name and the current version.

Nevertheless, each participant is expected to consider saving her results as a must. More precisely, she should periodically paste necessary information in a special text window and then click the Save button. As we already mentioned, the name of the file is defined automatically.

As a result, in 2007 no result losses were observed. The use of text format allowed us to keep the amount of submitted data reasonably small so that the Olympiad jury could easily handle it.

### Problem Classification

Analyzing the results of our Olympiads allowed us to classify the Olympiad problems according to their difficulties and factor out the principal obstacles that arise in understanding theoretical and practical aspects of computer science.

In choosing the problems we particularly aim at *surprising* the participants. We do not propose similar problems. The problems do not always expect the participants to write programs applying Logo commands. The main requirement we impose is to be able to read and understand the posed problems. Some problems were borrowed from published collections of mathematical and algorithmic problems and adapted to the Logo environment.

### Problems on Native Wit and Logic

These problems expect the ability to read comprehensively, to think logically, and to express the corresponding results. Such problems are typically chosen from collections of entertaining problems and proposed for the younger participants.

Having solved such a problem, the participants are usually expected to express the solution in Logo: write the text and save the project.

### Mathematical Problems

Many problems assume the ability to work with numbers and to apply basic mathematical skills. These problems are usually proposed to the participants of 5<sup>th</sup> and 6<sup>th</sup> grades.

We put a particular emphasis on problems in geometry, since these problems usually admit elegant solutions in the Logo environment.

If a participant does not have necessary mathematical background, she still can handle the given problem in the Logo environment, maybe even guessing the answers. Interestingly, even this approach is very useful, since it brings new understanding of mathematics and computer science.

### Modelling and Measurements

Problems that involve various kinds of measurements and selecting parameters of proposed models are of special interest in our Olympiads. The problems expect the participants to measure the parameters of certain objects, perform certain calculations, and apply the results of these calculations in practice. A problem of this class typically includes an image or an animation that facilitate understanding the problem statement. The participants are also provided with the procedure that created the animation. This procedure can be used for finding a solution to the

problem. We are convinced that reading and analyzing programs written by others is an important element in learning how to program. The solution to the problem typically involves measuring the object parameters and undertaking calculations based on the results of the measurements. All these operations are performed using Logo commands.

### Problems on Text Processing

The problems related to managing texts, such as converting texts from one format to another and searching the text for a given pattern, are particularly interesting. The younger participants improve their knowledge of the language and invent algorithms for locating given sequences in texts. The elder participants are expected to also apply these algorithms in practice.

### Algorithmic Problems

The Logo environment allows the students to learn basic algorithmic abstractions: linear sequences of actions, cycles, if-then and case constructions. The environment maintains the use of global and local variables, parallel processing, semaphores, etc. In our Olympiad problems, we encourage participants to learn how to use all these constructions. The use of variables facilitates the programming process and improves the quality of results. The use of parallel processes is helpful in modeling simultaneous actions of several objects.

## Concluding Remarks

Initially, our Olympiads were intended to advertise the Logo environment and to improve the Logo programming skills of the participants of the Olympiads. However, we observed an interesting phenomenon: faced with a problem, many participants do not make an effort to understand what the problem requires exactly. Instead, they often replace the given problem with another problem that they previously studied in school. We thus came to the conclusion that it is particularly important to express the problems in the most concise and unambiguous way. To prepare for an Olympiad, a student should train her skills in factoring out the most important part of a given problem, its idea.

Further, we expect the following methodology to be optimal in addressing a Logo modeling problem. The first step would be to think of how to set up an experiment in Logo. For this, it would be helpful to describe the experiment: the setting typically includes Logo objects and commands (programs) that manage these objects and report the measured parameters. Clearly, the experimental setting (the model) should be portable to any machine on which the corresponding version of Logo is installed. Once the setting is prepared, it can be used for experimental studies. A conclusion drawn from the results of the experiments is often considered as the solution to the problem.

For students with a positive programming background, we foresee the following elements to be important to successfully participate in our Olympiads:

- The use of variables in algorithms.
- The use of *list* variables for elegant programming constructions, such as cycles.
- The use of selection algorithms, such as locating a list element with a given identifier or a list element satisfying a given criterion.

We would like to emphasize our intention to use the Logo environment primarily for computer modeling, and not necessarily for solving problems in a theoretical manner which is often required in traditional programming contests.

To conclude, we would like to cite Nicholas Negroponte who wrote in his preface to Seymour Papert's book "Connection Family":

«Kids bring a new culture to the family landscape, a culture which has at its core the extremes of being simultaneously personal and global. Children understand computers because they can control them. They love them because they can make their own windows of interest» [5].

## References

1. Seymour Papert, Mindstorms: Children, Computers and Powerful Ideas, Dsic Books, Inc. Publishers / New York, 1989
2. Abramova G.S., The Age Psychology, Textbook for University students, Moscow, Akadem A, 1998 (in Russian)
3. Johan Gielis, «A generic geometric transformation that unifies a wide range of natural and abstract shapes», a review of the article can be found at <http://astronomy.swin.edu.au/~pbourke/curves/>, 2002
4. Seymour Papert, The Connected Family - Bridging the digital generation gap, Longstreet Press, Atlanta, Georgia, USA
5. Nicholas Negroponte, Being digital, Alfred A. Knopf New York, 1995