

Implementation of Dynamic Tasks on Informatics and Computational Thinking

Valentina DAGIENĖ, Gabrielė STUPURIENĖ, Lina VINIKIENĖ

Vilnius University Institute of Mathematics and Informatics
Akademijos Street 4, Vilnius LT-08663, Lithuania

valentina.dagiene@mii.vu.lt; gabriele.stupuriene@mii.vu.lt;
lina.vinikiene@mii.vu.lt

Abstract. A well-known challenge on Informatics and Computational Thinking Bebras is based on task solving activity. The Bebras model is known as one of the methods that involves students to the learning and promotes students' reasoning. The Bebras motivates students to be interested more deeply in informatics. This goal can only be reached if the tasks are interesting and provoke some excitement. The interactivity is very typical for computers, so it is clear that a computer oriented challenge / contest should apply interactive elements to solve or explain tasks. This paper presents a tool that is used to code and design dynamic Bebras tasks. The paper has two main goals: 1) to analyse tasks used to engage student's in learning informatics fundamentals, and 2) to provide a description of the tool, so-called the *Bebras Lodge*, and illustrate application of this tool by discussing several examples.

Keywords: Bebras challenge, computational thinking, computer science education, informatics education, interactive task, dynamic task, Bebras Lodge tool, problem solving

1 Introduction

A detailed analysis based on activities used to engage students can be found in the paper of Carter et al. (2011). Authors considered competitions as a way to motivate, inspire students, test their abilities, understanding, and stimulate their enthusiasm. The practice in competitions/contests could be implemented in the futures studies.

The *Bebras challenge* started in Lithuanian in 2004 and currently spread to more than 50 countries. Over 2 million students participated in the challenge during 2016. In Lithuania 33 006 students from 3rd to 12th grades solved tasks in the first round of the challenge. The 385 best participants of the first round (from 9th to 12th grade) were invited to the second round. In Lithuania students are mainly asked to participate in the challenge during information technology (IT) lessons. Teachers engage students to participate in the challenge in order to attract their attention to the basics of informatics (computer science or computing) by solving short, game-based tasks (Dagiene and Stupuriene, 2016).

The main features of the *Bebras* tasks are concept-based, well design, interactive, visualized, and solvable. Multiple-choice questions and dynamic tasks are designed to be

solvable by participants during the challenge. Multiple-choice tasks are questions with opportunity to select one correct answer from four. Dynamic tasks are the following: students have to manipulate objects visually, or students have to write and run small programs. Pozdniakov et al. (2016) discussed types of tasks and paid attention to dynamic tasks. According to Pozdniakov et al. dynamic tasks should include interactive components, e.g. drag-and-drop operations, point-and-click mouse interactive functionality, text or integer input boxes.

Germany, Lithuania, Romania, Singapore, Spain, Switzerland, The Netherlands, Ukraine, and USA have dynamic tasks in their annual *Bebras challenge*. In these countries, *Bebras Lodge* tool is used as a tool to code tasks interactivity (<http://bebras.licejus.lt/login>). Finland, Hungary, and Russia indicated that they have plans to use the *Bebras Lodge* tool in the nearest future. Dynamic tasks are popular in France as well, but these tasks are developed by using their own designed *Bebras contest management system*. Italy implement tasks interactivity by using HTML5. In Estonia, Finland, Japan, and Russia dynamic tasks are supported by *JavaScript*. Belarus, Canada, Serbia, Turkey, Croatia, Indonesia are planning to add some kind of dynamic tasks or develop more interactive task features.

Bebras Lodge is an authoring tool developed for coding and implementing dynamic tasks. Users can simply create their own informatics tasks and share them with other users. *Bebras Lodge*'s job is to make dynamic tasks content creation as easy as possible. The tool gives users the power to build dynamic tasks for learners without having to develop a full technical understanding of the process. The created content could be reused and modified flexibly. Content may be exported using for example SCORM. *Bebras Lodge* allows to package up and export a created task as a SCORM file, which comes in the form of a .ZIP file. Almost every learning environment is equipped to handle content in this format. Tutorials and documentation are available for users.

In this paper we analyse *Bebras Lodge* tool that supports interactivity of tasks used in Lithuanian *Bebras challenge* on Informatics and Computational Thinking. The paper is organized as follows. Section 2 provides the most important features of tasks used to involve students to learning and promote their reasoning. In section 3 we are going to overview *Bebras Lodge* tool and provide some examples. At the end is Conclusion.

2 Tasks that engage students reasoning

This section presents related works on educational tasks features according tasks development and using goals. These tasks bring together students and teachers in order to rethink learning or teaching content and implement knowledge in an interesting and successful way.

Many computers based activities, games, programming, challenge / contest support the teaching of computational thinking skills. Game-based learning and constructivism are emphasized by Kalelioğlu et al. (2016). They provide a detailed literature review based on computational thinking that is “a very important and useful mode of thinking in almost all disciplines and school subjects” (p. 590).

The students' success depends on the relation between tasks of challenge and reasoning used by students to solve tasks. Researchers (Boesen et al., 2010) emphasized the importance between tasks characteristics, types of tasks and reasoning due to the

influence to student grade, learning interest, teacher work, and textbook author. Furthermore, they paid attention to mathematical reasoning that is influenced by student and social environment.

Barak et al. (2011) discussed animations that motivate students. Their goal in the research was to find out how animated movies were used to promote students' ability to explain and understand scientific concepts. They claimed that animation encourage students to explore concepts, provide possible answers to daily life questions, motivate to learn science. Other aspects of effective learning are an interactive visualization. Patwardhan et al. (2015) claimed that interactive visualization provide opportunity to practice, develop analytical skills. They mentioned interactivity as manipulation of content of dynamically and "manipulation of parameters that influence the phenomena/concept being depicted" (p. 294).

Interactive elements of the task (graphic, animation, etc.) support student's understanding of a content and how they construct meaning from the presented content. The following goals of the dynamic tasks benefit could be mentioned: increased student engagement and motivation, measurement of higher order thinking skills, promoted students' reflection by solving tasks, better evaluating the cognitive and problem-solving skills (Strain-Seymour et al., 2009).

Crisp (2010) focused on the interactive assessment in order to improve student capabilities on the case of a higher education. He claimed that interactive assessment is innovative and should be developed using questions such as drag-and-drop, hotspot, extended matching questions. Also he provides a question design strategy for interactive tool in common learning management systems.

Focus on mouse interaction styles such as drag-and drop, point-and-click are important for students. The detailed analysis is provided in the paper of Inkpen (2001). It is emphasized that the point-and-click interaction style is more attractive, faster solvable and motivating than drag-and-drop, the impact of interaction depends on the steps needed to solve problem. In addition, the author emphasized the importance of children's interaction with computer and the applicability of educational software.

2.1. *Bebras* tasks for engaged learning

Teachers are searching for the tasks that not only provide understanding of Informatics concepts and how these concepts are implemented in real life situation, but also for the tools used to develop abilities of students to analyse, think, and find the best solution of the problem. Task selection influences these goals and students interest in the future studies, work. Dagiene et al. (2017) emphasized the importance of the *Bebras challenge* tasks as possibility to develop computational thinking. *Bebras* tasks are examples of problems that help teachers explain informatics basic and improve students' abilities. Tasks inspire teachers and students to work together (teacher and students, students and students) and collaborate in decision-making in order to find the best solution. Each new informatics concept is connected to realistic situations of a particular task.

Vaniček (2014) analysed the criteria of the tasks for *Bebras challenge*. He

emphasized that task interactivity is attractive due to graphic components and manipulation with the mouse. Based on students' opinion, interactive questions are more attractive than multiple-choice questions used in the challenge as well. It is influenced by opportunity to think and experiment. Students pay attention and spend more time to solve interactive questions than other type questions.

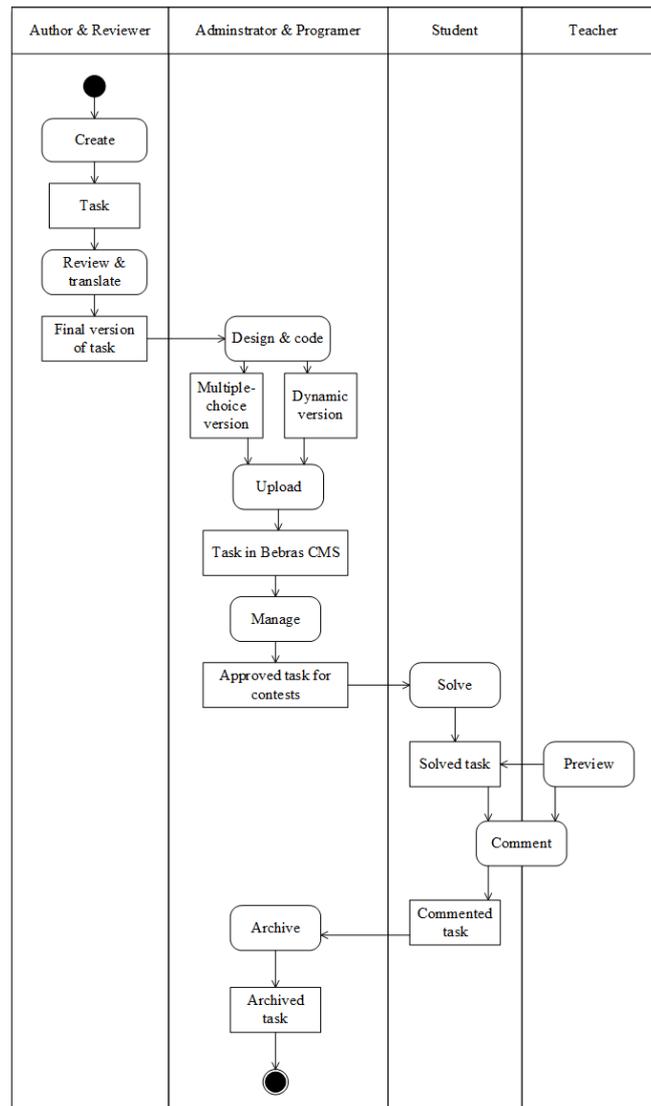


Fig. 1. Activity diagram for the *Bebras* tasks

Tomcsányiová et al. (2011) described interactive questions as tasks which are solvable by manipulating objects within the task assignment and leaved the objects in a state that corresponds to the tasks assignment. Furthermore, multiple-choice questions

are mentioned as interactive questions according the using interactive helpers that provide direct manipulation of the task's objects in order to experiment with them (Tomcsányiová et al., 2011, p. 204).

The development of *Bebras* tasks in Lithuania is a process, which involves various activities and needs different resources. Fig. 1 depicts activity diagram and following steps that describes the tasks preparation and usage in the challenge:

1. National challenge organizers (authors) prepare a set of task proposals. They provide the most promising 8-10 tasks to the task *Repository* (SVN)¹ at least one month before the *Bebras* International Task Workshop (yearly in May). During the workshop experts from the *Bebras* community countries review, improve the tasks. National challenge organizers select tasks from the task pool. Selected tasks are translated into the national language and edited by a linguist. The final version of tasks is prepared by national challenge organizers.
2. Static tasks (multiple-choice) are developed by using *Bebras contest management system* (lt.bebras.lt) and dynamic tasks (interactive questions, animated questions, open-ended etc.) are developed by using *Bebras Lodge* tool and uploaded to the *Bebras contest management system*. When all tasks are prepared in the *Bebras contest management system*, system administrator creates a set of tasks according to the participants age groups (manage approved tasks).
3. During the challenge teachers give the permission for students to solve the task set; participating students solve tasks and submit their answers. After the challenge solved tasks are open for comments and discussions.
4. Later, all tasks are archived in the *Bebras contest management system*. All of them can be reused.

In the assumption, the development of interactive questions, problems involve students to the analysis of the content, motivate them to solve problem by reasoning not guessing the correct answer.

Furthermore, interactivity, animation and problems with the visual, interesting story engage collaboration between students and teachers. Scientists, teachers, students are brought together for the reason that they will discover the attractiveness in the informatics by solving *Bebras* tasks. Due to that not only tasks' content is important, but also tools selection for tasks development and challenge management as well.

3 *Bebras Lodge* tool

In this section, we describe a tool used in the *Bebras* challenge in order to create dynamic tasks. *Bebras content management system* is mainly used for management of the challenge as well. The system supports multiple-choice questions with opportunity to select one correct answer from four variants. Therefore, the system allows developing multiple-choice tests directly. By default, choices are displayed in random order during the challenge. For multiple-choice tasks, administrator fills these fields: task ID, title, task description and possible answers, answer comment if it is needed, choose task

¹ <https://tortoisesvn.net/>

difficulty, age group, language. *Bebras contest management system* didn't support the development of dynamic tasks. There is one of the reason why *Bebras Lodge* tool was realized. As was mentioned before, *Bebras Lodge* is used by 9 countries. The number of created dynamic tasks is growing (from 40 tasks templates in 2013 increased to more than 113 in 2016). Part of tasks has the same type of interactivity, but they are developed in different languages or changes in the text.

To promote students' motivation in the challenge each year organizers provide several dynamic tasks. Lithuania provided 26 dynamics and 20 multiple-choice questions in 2016. Dynamic tasks are developed using *Bebras Lodge* tool and imported to the *Bebras contest management system*.

The *Bebras Lodge* consists of two main parts:

- A Componential part – a *JavaScript* library for creating and manipulating components inside a task;
- A Lodge used interface – the interface built on top of Componential part in order to make task development seamless.

The Componential library depends on two other libraries: *jQuery* - for easier cross-browser functionality and *RaphaëlJS* – for SVG graphics. In the *Bebras Lodge*, users can easily generate dynamic tasks if they have a little bit of programming knowledge. A typical task consists of *JavaScript* code, images, cascading style sheets, correct answers, Meta information. When user login to *Bebras Lodge*, he/she can start creating a new task, work on existing task or use the task template. Some basic metadata such as task ID, name, authors, language is added, others components should be added. Components are added by dragging and dropping components onto the tasks canvas in the visual interface or defining them with *JavaScript*. The following API components are available: Task, Component, HTML, RaphaelComponent, TextField, Picture, Switch, SwitchGroup, Graph, InfixText, LinearLayout, and GridLayout

All components have various parameters. For example, a rectangle component will have a colour, size and coordinates. Some parameters are common to all components, among these are name, coordinates and a Boolean flag denoting whether the component is part of the answer. In terms of rendering components are of two types: HTML components and *Raphaël* components. If parameters of components are not enough for users, they can develop new components by writing *JavaScript* code. To make it easier for users, a list of task templates is developed and added to the system. Users can add data parameters (not to be confused with component parameters) and upload images. These parameters and images (in form of string URLs) are accessible in the task *JavaScript* code and in the visual interface. Each component can be marked as part of the answer to the task. For example, user might be required to drag some boxes into the correct positions and enter the sum of the weights of those boxes into a text field. When in the "Live" mode, the user can solve the task and add the current configuration as the correct solution. The answers can also be added manually.

Users can use these templates to create a simple new dynamic task. Dynamic tasks are exported in *JSON* format. Exported *JSON* file contain all information needed to render the task, additional resources that might need to be downloaded are images listed in the exported file, *JavaScript* framework for rendering the tasks, and *Raphaël* library. Templates of the following general types of tasks are available at the *Bebras Lodge*:

1. **Drag & drop.** Pick up an object and place it on one of the fixed locations.
2. **Click objects.** Click on an object and make something happen with it (picture changes).
3. **Simulation.** Show an animation as explanation of the task or the input of the user.
4. **Open text.** Enter any text.
5. **Open integer.** Enter text, but with specified symbols (a restricted answer).
6. **Matching.** Match one object to one other object.
7. **Selection.** Select one or multiple objects from a task.
8. **Graph.** Connect one object to multiple other objects.
9. **Grid.** Select an object and effect other objects while doing so.

The documentation and manual for the tools are prepared for the beginners. User can use provided examples or use given templates of codes.

The dynamic tasks' tool is used not only for task preparation for the challenge, but also for programming practice or dynamic task development in different online learning environments which don't support interactivity as simulation, animation, etc. Teachers could develop dynamic tasks using a tool available at <http://sprendimas.ugdome.lt/> (however, it is only in Lithuanian). The export to a SCORM is supported, so teachers can easily create dynamic tasks and import them to a particular system that supports SCORM standard.

3.1 Dynamic task examples

Further, we are going to discuss some dynamic *Bebras* tasks examples coded and designed by using the *Bebras Lodge* tool. In 2016, the most used dynamic task in the *Bebras Challenge* were the following:

1. Concurrent directions (used by 12 countries);
2. Paint it black (used by 11 countries);
3. Spherical robot (used by 11 countries).

In Lithuania, these tasks were taken for different difficulty levels (easy, medium, hard) according to the age groups. The percent of students who solved these tasks correctly and tasks difficulty provided by challenge organizers is presented in Table 1.

Table 1. Dynamic tasks solvability by age group in Lithuania.

Title \ Grades	3-4	5-6	7-8	9-10	11-12
Concurrent directions		44.79 (medium)	70.45 (easy)		
Paint it black				33.17 (hard)	66.98 (medium)
Spherical robot	67.35 (hard)				

Table 1 shows that older students solved these tasks better than younger. Furthermore, more than 60% students solved "Spherical robot". Reasons why students were not so

successful in the task could be tasks difficulty provided by organizers, interactivity, understanding of informatics concepts, etc. However, we do not analyse these reasons, we look to the interaction that is given for students due to the different actions in order to get correct solution. Furthermore, we give a short overview for each tasks, why it is informatics and which basic informatics concepts are presented in the context of the task. Different concepts are depicted in these tasks.

Figures below (Fig. 2, 3, 4) shows the “Live” mode of the *Bebras Lodge* visual interface (area where user can test the task). In the left side is provided the created task with possibility to check the correct answer. Source code is visible in the right side. It is visible if user program the additional component that is not provided in the list.

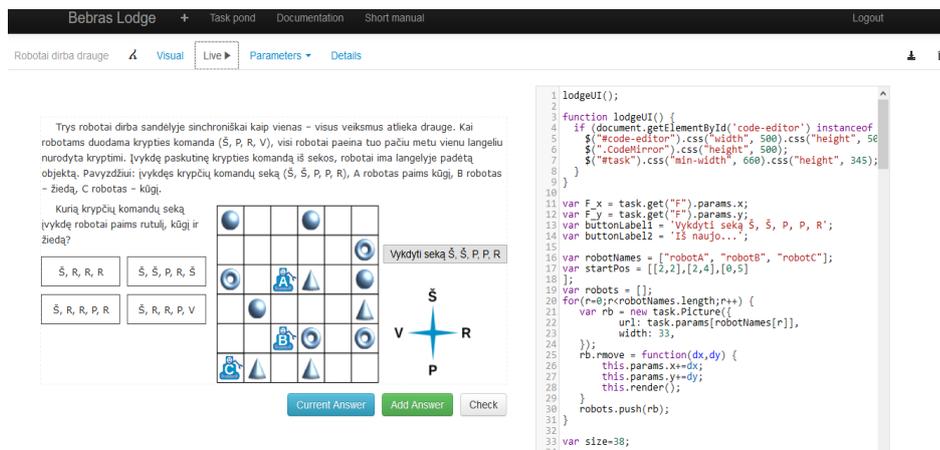


Fig. 2. Task *Concurrent directions* designed with *Bebras Lodge*

From the point of Informatics education, the first task (*Concurrent directions*) is related with algorithm, parallel programming and concurrent programming. In Lithuania, this task was given for grades 5-8.

When robots or computers are working together at the same time, we say that they are working in parallel. If we have a small number of robots or computers working in parallel, we could give each of them a different list of instructions. However, if we have thousands or millions of computers working together then it is not practical to write separate instructions for each. We have to give large groups of them the exact same instructions. In this task, we have the potential for an additional complication. The robots are working in the same space, and their instructions must be prepared more carefully to ensure that they do not crash into each other or otherwise block one another. Preparing parallel instructions in such a situation is an extremely challenging aspect of computer science called concurrent programming.

The second task (*Paint it black*) was given for grades 9-12 in Lithuania. Task represents a Boolean circuit that is one of mathematical computation models. An equivalence is one of basic Boolean operations. If the yellow cell is interpreted as 0 or FALSE and the black cell as 1 or TRUE.

```

1 lodgeUI();
2
3 function lodgeUI() {
4   if (document.getElementById('code-editor') instanceof
5     $('#code-editor').css('width', 500).css('height', 500)
6     $('#codeMirror').css('height', 500);
7     $('#task').css('min-width', 600).css('height', 345);
8   }
9 }
10
11 one=[1,1,0],[1,1,0],[0,0,0];
12 two=[0,1,1],[0,0,0],[1,1,0];
13 erg=[1,0,1],[1,1,0],[1,1,0];
14 thr=new Array(3);
15 xoff=[0,130,200];//offsets of the 3 tables
16 yes=210;
17 res=new Array(3);//player's results
18 es=3;//width height of image
19 con=false;
20 text=task.params['text1'];
21 question=task.params['question'];
22 new task.HTML({content:question}).move(0,170);
23 new task.HTML({content:text1}).move(0,0);
24 new task.HTML({content:'A kortelė'}).move(25,30);
25 new task.HTML({content:'B kortelė'}).move(165,30);
26 new task.HTML({content:'C kortelė'}).move(345,30);
27 new task.Picture({url:task.params.g1}).move(0,50);
28 new task.Picture({url:task.params.g2}).move(130,50);
29 new task.Picture({url:task.params.g3}).move(300,50);
30 task.canvas.image(task.params.pfeil,xoff[1]=120,97,63,17
31 task.canvas.image(task.params.pfeil,xoff[1]=120,yes+47,6
32 for(i=0;i<3;i++){
33   res[i]=new Array(3);
34   thr[i]=new Array(3);

```

Fig. 3. Task *Paint it black* designed by using the *Bebras Lodge*

Third task (*Spherical robot*) was given to grades 3-8 in Lithuania. A computer program is a sequence of instructions from a set of possibilities. This task requires one to write a computer program in a very simple programming language that consists of only four possible commands N, S, E, W. This introduces one important element of many computer-programming languages, sequential composition, which means following commands one after another in order.

```

1 lodgeUI();
2
3 function lodgeUI() {
4   if (document.getElementById('code-editor') instanceof
5     $('#code-editor').css('width', 500).css('height', 500)
6     $('#codeMirror').css('height', 500);
7     $('#task').css('min-width', 600).css('height', 345);
8   }
9 }
10
11 var atsakymas = new task.Component({answer: true, value:
12 }
13 function makeImg(url) {
14   return '<img style="max-width:none; width:164px; sr
15 }
16 new task.HTML({content:makeImg(task.params.kryptysImg)}))
17
18 xo=500;
19 yo=50;//offsets
20 vu=70;//vertical unit
21 u=40;
22 fpd=8;//frames per direction
23 dg=0.6;//diagonal distance per unit 1
24 var ox, oy, oz, order, ldx, frame;
25 Xo=0; Yo; Z=0; fail=false;
26 lauft=false;//animation on
27 stop=false;//force stop
28 function p(x,y,z) {
29   return "+(x0+u*(x+z*dg))+ "+(yo-u*z*dg+vu*y));
30 }
31 function hole(x,y,z) {
32   if(x==1&&y==1&&z==0) return true;
33   if(x==0&&y==2&&z==0) return true;
34   if(x==1&&y==3&&z==1) return true;

```

Fig. 4. Task *Spherical robot* designed by using the *Bebras Lodge*

During the challenge students have to submit the answers if the tasks are interactive. Furthermore, they are able to edit their answers during the solving time. A detailed interactivity is described in the Table 2 according to the action, which is done by students to solve task. Tasks were designed using different components and parameters.

Table 2. Students and computer interaction in particular task.

Task name	Task type	Interactivity
Concurrent directions	Clicking objects	Students are able to click on the buttons that will be animated in the picture. Picture gives for student visual effects of movements. Animation is provided as helper to check his/her solution. This task emphasizes a points and the control information flow.
Paint it black	Clicking objects	Students respond a task by clicking on the answers that he/she think are correct. According to students' actions (clicking) on the particular part of the picture, picture is changed and students are able to see changes in the picture immediately. Differences are seen by clicking the mouse button on the picture.
Spherical robot	Simulation	Students are able to type the answer and see animated answer. The constructed (typed in the text field) sequence is displayed as animated. These tasks emphasize the control information flow as well.

Tasks are very important for both students and task developers (teachers): students have been “pushed” to think on computer science, educators should think about harmonization of curricula on computer science in school. Creative, interesting, and dynamic tasks are the main drive for the *Bebras challenge* on Informatics and Computational Thinking.

4 Conclusions

This research is based on the investigation of tools, which support the development of dynamic tasks. In Lithuania, two main systems are used for management of *Bebras challenge*. Multiple-choice questions are developed by using the *Bebras contest management system* whereas dynamic tasks are designed using the *Bebras Lodge* tool. The main focus of the article is *Bebras Lodge* usage. This tool was developed by Lithuanian researchers in order to encourage students to solve dynamic tasks in more interesting, attractive way than marking correct answer as in the case of multiple-choice questions.

According to learning by the practice and interest in tasks solution, more interactive tasks have been developed every year. Different types of interactivity are implemented in the *Bebras Lodge*. Our research dealt with a description of *Bebras Lodge* and tasks examples, however no analysis on students' abilities to solve tasks correctly according to context of the tasks and their interactivity levels are provided. A detailed analysis of factors that influence students' motivation, success by solving informatics tasks is needed to be done in the future.

Acknowledgements

We acknowledge the international *Bebras* Community for their collaboration and sharing of ideas about tasks. Sincere thanks Jari Koivisto, our longtime colleague at informatics education, for proofreading the paper and making suggestions.

References

- Barak, M., Ashkar, T., Dori, Y. (2011). Learning science via animated movies: Its effect on students' thinking and motivation. *Computers & Education* 56(3), 839–846.
- Boesen, J., Lithner, J., Palm, T. (2010). The Relation Between Types of Assessment Tasks and The Mathematical Reasoning Students Use. *Educational Studies in Mathematics* 75(1), 89–105.
- Carter, J., Bouvier, D., Cardell-Oliver, R., Hamilton, M., Kurkovsky, S., Markham, S., Mcclung, O., W., Mcdermott, R., Riedesel, C., Shi, J., White, S. (2011). Motivating All Our Students? In: Adams, L., Jurgens, J. J. (Ed.), *Proceedings of ITICSE*, Conference on Innovation and technology in computer science education, (27–29 June 2011, Darmstadt, Germany), ACM, 1–18.
- Crisp, G. (2010). Interactive e-assessment – practical approaches to constructing ore sophisticated online tasks. *Journal of Learning Design* 3(3), 1–10.
- Dagienė, V., Sentence, S., Stupuriene, G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. *Informatica* 28 (1), 23–24.
- Dagienė, V., Stupuriene, G. (2016). Bebras - a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in Education* 15(1), 25–44.
- Inkpen, K., M. (2001). Drag-and-drop versus point-and-click mouse interaction styles for children. *ACM Transactions on Computer-Human Interaction* 8 (1), 1–33.
- Kalelioğlu, F., Gülbahar, Y., Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing* 4(3), 583–596.
- Patwardhan, M., Murthy, S. (2015). When does higher degree of interaction lead to higher learning in visualizations? Exploring the role of “Interactivity Enriching Features”. *Computers & Education* 82, 292–305.
- Pozdniakov, S., N., Kirynovich, I., F., Posov, I., A. (2016). Contest “Bebras” on informatics in Russia and Belarus. *Olympiads in Informatics* 10 (Special Issue,) 55–65.
- Strain-Seymour, E., Way, W., Dolan, R., P. (2009). *Strategies and Processes for Developing Innovative Items in Large-Scale Assessments*. Pearson Education, Inc., New York.
- Tomcsányiová, M., Tomcsányi, P. (2011). Little beaver – a new Bebras contest category for children aged. *LNCS* 7013, 201–212.
- Vaniček, J. (2014). Bebras Informatics Contest: Criteria for Good Tasks Revised. *LNCS* 8730, 17–28.

Received August 28, 2017, revised September 4, 2017, accepted September 21, 2017