Experience gained from the development of a library for creating little on-line educative applications

Monika Tomcsányiová[1], Peter Tomcsányi[1] and Karolína Mayerová[1]

[1] Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia
{tomcsanyiova, tomcsanyi, mayerova}@fmph.uniba.sk

Abstract. Our aim was to develop an easy to use JavaScript library that allows a teacher student or a teacher, who knows only basics of HTML and JavaScript, to implement little educative applications similar to Bebras interactive tasks. Using the method of design-based research we developed such a library and named it sprites.js. In our paper we describe the experiments with the library, which focused on several different groups of participants and their experiences with library sprites.js. In each of the group we examined both the process of creating the applications as well as the created applications themselves. The result of our work includes the library sprites.js, the course of JavaScript for teacher students, the new tool for implementing interactive tasks for our Bebras contest, and our knowledge that users with little programming experience can develop little educative applications when using a properly designed library.

Keywords: JavaScript, Bebras contest, educative programs

1 Introduction

Creating little educative applications that can be accessed on-line from a browser of any computer or mobile device is a quite easy task for a programming professional, but at the same time it is a very hard task for an informatics teacher or student who knows only basics of HTML and JavaScript. The most usual types of applications a teacher may want to implement are interactive microworlds for enhancing the knowledge in a very particular area. Therefore the aim of our research was to develop an easy to use JavaScript library that would allow a teacher, who knows only basics of HTML and JavaScript, to implement little educative applications similar to interactive tasks [1] of Bebras [2].

Using the method of design-based research [3] we developed such a library and named it sprites.js [4]. At first we describe the research cycles that led to the development of the library (section 2). Then we describe the experiments made by using the library in several setups: in an advanced JavaScript course for Master students of Applied Computer Science (section 3), in a JavaScript course for teacher students (section 4), with a teacher student preparing her master thesis (section 5), and for developing and running the interactive tasks of our national Bebras contest (section 6). All these experiments were conducted and evaluated using qualitative methods [5].

2 The development phase of sprites.js

The development of sprites.js was conducted in several stages. At first we conducted a survey on the development of mobile applications by students of Applied Computer Science at our faculty, which are more experienced programmers than our teacher students. In one of their compulsory subjects the students had to develop little educative applications for the Web using JavaScript with any libraries or frameworks at their choice. We analyzed their solutions and, based on this analysis, we specified the features and functions that should be contained in a library that would allow a much less experienced programmer to develop such applications.

Then we started the development of sprites.js. We used the design-based research method [3], which assumes a tight cooperation between the developer of the library and its users. Therefore the developer cooperated with the developer of little educative applications for supporting the teaching of the informatics subject at the primary and lower secondary school.

In the first iteration the design of the library enabled only the basic cooperation with Web browsers. To make the specification better, we selected some interactive tasks of the Bebras contest [2] as the ones we want to implement using the library.

In the next iterations we enhanced the library. The version of sprites.js at the end of this phase was a JavaScript library which solves the technical details of a typical little educational application. The central class of the library is the *Sprite*, its methods allow defining its reactions to mouse and/or touch actions. The library uses the Canvas object of HTML5 therefore the applications can run also on mobile devices.

3 Experiment with master students of Applied Computer Science

During the winter term of the academic year 2016-17 we implemented an experiment with 1st year master students of Applied Computer Science. In their Web programming course they had to design and implement little educative applications. They had to do it twice, once using the library sprites.js, once without using it.

We wanted to know whether students that are experienced programmers when implementing the assignments on their own will use a similar way of visualizing the graphical elements of the application as it is done in sprites.js or not. Because the students had enough experience for developing applications with the library as well as without it, we can compare their two implementations and draw some conclusions.

In the versions of applications without sprites.js the students were allowed to use any libraries of their choice. Usually they included the well known library JQuery, which requires the programmer to be familiar with it and also with CSS at at least intermediate level. In the article [6] our colleagues analyzed three applications programmed by the students and also the approach they had taken for implementing it without the library sprites.js. In two of them the students took the “classic” approach using <img> objects and DOM. The third application implemented its own way of handling graphics that was similar to the library sprites.js. The results of this experi-

ment support our initial idea that programming little educative applications without using the Canvas element would be quite complicated and would require a lot of purely technical knowledge of HTML5 and CSS.

4 Experiment with undergraduate teacher students

In this part of our research we focused on the question: what is the minimum necessary knowledge of JavaScript programming for a programmer to be able to implement applications similar to interactive tasks of the Bebras challenge if the main part of working with images will be implemented by sprites.js?

We conducted the experiment with three undergraduate teacher students, which chose the JavaScript Programming course from a group of compulsory elective courses. The three students were quite different. Student A is a good programmer; he achieved good results in his previous programming-related courses. Student B seems to be less proficient, he spends less time on his work during the semester, but at the end of the semester he is able to manage the exams of programming-related subjects quite well. Student C appeared to be the weakest one of the group, at the beginning of the semester he worked quite diligently, but later in the semester when the assignments became harder he started to lose his interest.

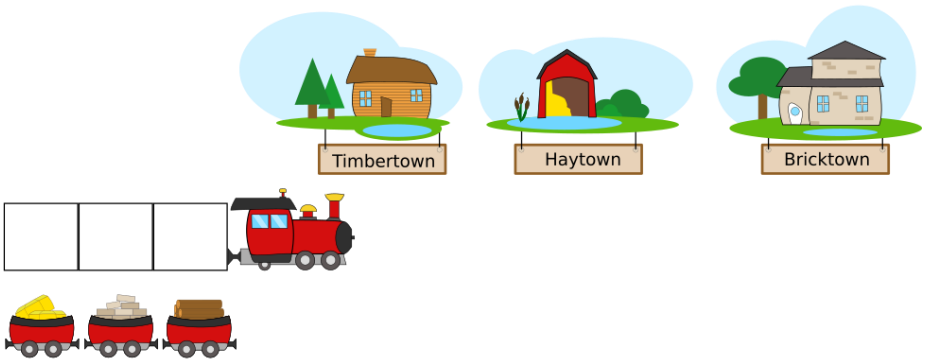
We created a JavaScript course for these students. The course built on their programming experience gained previously: two semesters of programming in Pascal, one semester in Python and in the same semester they attended a course of Imagine Logo. Therefore the initial part of the course was quite quick. The first visible problems arose when the students needed to use elements of OOP, and especially when we started to teach them how to programmatically read and paint pictures. The students consider reading and painting pictures to be a very hard task. For dealing with pictures we finally adopted an approach similar to the one used inside the sprites.js.

After implementing a few assignments, which did not use sprites.js, we created two templates that explain the syntax and semantics of sprites.js functions and methods for handling pictures. In one of them the end-user has to drag pictures to their correct place (drag and drop activity, see Fig. 1), in the other one the end-user has to click some pictures (click activities). These two templates were then used as templates for implementing further Bebras tasks.

There were no problems during the lesson. Students B and C implemented two assignments, student A even three assignments while the third one had no prepared graphics, so he had to create them from the screenshots of the Bebras task that we provided. Student A implemented the third assignment in about 15 minutes while previously, when not using sprites.js, it took him about 60 minutes to implement equivalently complex assignments.

During the next lesson the students were preparing for the final exams and had to solve one assignment in two ways, without sprites.js as well as with it.

The Beaver City train has to deliver materials to three towns along the train line: Timbertown, Haytown, Bricktown. Timbertown needs timber. Haytown needs straw. Bricktown needs bricks.
The train passes the cities in the order shown in the picture below.Attach the train carriages to the locomotive so they can be delivered easily in the right order by simply detaching the last carriage in each station




```
var targets = [];

onDragDrop = function(sprite) {
  var target = sprite.findOverlapped(targets);
  sprite.placeAt(target);
}
var background = new Sprite("background.png", 410, 120);
var names = ["straw", "bricks", "timber"];
for (var i = 0; i < 3; i++) {
   var carriage =  new Sprite(names[i] + ".png", 35 + i * 70, 260, dragSprite);
   targets[i] = new Sprite("empty.png", 45 + i * 60, 197);
};
```

Fig. 1. The template for drag and drop activity, based on the Bebras task 2016-AU-07 Train

During the final exams we conducted a non-participant observation. Student A solved the final exam assignment, both without sprites.js and with it in 80 minutes. The longest time he spent on painting the pictures without using sprites.js. Another sub-task that took him quite long was the implementation of testing the end-user's correct answer.

The experiment supports our statement that if the teacher students get a reasonably powerful yet easy to use JavaScript library, they can learn how to use it to implement applications that are suitable for informatics education at a lower secondary school.

5 Experiment with a teacher student preparing her thesis

We conducted an experiment with a teacher student in the 1st year of her Master studies. She studied JavaScript herself because she needed it for her master thesis. During the semester she was provided with the manual for sprites.js that contained examples and templates for developing applications using the library. Similarly to the other students, she had problems with programmatically painting pictures in different web browsers when she did not use sprites.js. She was not able to work out the solutions herself. On the other hand, she was able to use sprites.js and quite easily implement the applications she had to without any additional explanations from our side.

6 Experiment with implementing Bebras tasks

In this experiment we gave sprites.js to three programmers of interactive Bebras tasks for the local contest in our country in 2016. The goal was to exchange the system that had been used previously and which was based on Flash with one based on HTML5 thus making our contest more accessible to tablet and smartphone users.

We started by implementing changes in the local contest system so that it generates the web pages for interactive tasks using HTML rather than Flash. In this phase some new requirements arose for the developer of sprites.js, so three more development iterations of the library were performed. The final changes included:

- Creating a template that includes checking the correctness of the answer given by the en-user of the application (the Bebras contestant).
- Allowing to use base64-encoded pictures instead of file names in order to reduce the number of files per task in order to minimize the number of http requests to the server during the contest.
- Placing several tasks at the same HTML page. Our contest system needs it for displaying one pupil's results on the same page. An Activity class, which takes care of one task's sprites, was added to the library.
- Providing information of the task's current state which can be sent to the server for saving it and later, when the contestants opens the same task once more, set the same state.

All the required enhancements made the code of the tasks less understandable to undergraduate teacher students. Therefore we kept two branches of sprites.js.

The process of creating tasks was quite straightforward because it has been carried out by experienced programmers. The most time consuming part was preparing the pictures. We used PNG format (instead of SVG which is preferred in the Bebras community) because the way sprites.js uses the pictures was not working with svg format on some of the common Web browsers. Using sprites.js the three Bebras task programmers implemented 19 interactive tasks that were used for the local Bebras contest in our country.

Compared to the Bebras Lodge [7, 8] system used by some other countries, we consider our system easier to use for less experienced programmers and simpler for implementation into the contest system because each interactive task consists only of one client-side .js file and no per-task server-side code.

The use of sprites.js sped up the transfer of the contest system from Flash into HTML and allowed some schools to use mobile devices for the contest. Particularly one school reported that they were solving the tasks in a train during their journey to an excursion that had been planned for the same day as the Bebras contest.

7 Conclusion

During the period of last two years we designed and, using the design-based research method, we developed the library sprites.js, which implements the basic way of communication with a Web browser, handling images and mouse or touch events. Using

its functions we are able to implement interactive Bebras tasks for next years of our local Bebras contests.

We designed and executed research experiments, which are experienced web programmers. We proved that in order to develop little interactive applications for the Web without a simple to use library requires considerable knowledge of HTML5 and related technology.

For the teacher students at our faculty we designed a JavaScript programming course and ran the course for the first time. The effect of the course will be that teacher students, which later become teachers, will be able to design and implement tasks similar to Bebras interactive task on their own only with the help of sprites.js and the templates that we taught them during the course. They can use such applications to support their teaching at lower secondary schools.

In the future we consider building a WYSIWYG environment around sprites.js so that the users do not need to code the coordinates of sprite's positions or create some kinds of graphics (e.g. one containing text only) in separate editors. We also consider ways to make the resulting Bebras harder to hack by the contestants.

Acknowledgements

The article was created as part of a project KEGA 080UK-4/2015: Support for the development of educational digital content for mobile devices.

References

1. Dagiene, 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.
2. Bebras International Challenge on Informatics and Computational Thinking (2017). http://www.bebras.org, last accessed 2017/07/01.
3. Bakker, A., Van Eerde, H. A. A. (in press). An introduction to design-based research with an example from statistics education. In A. BiknerAhsbahs, C. Knipping, & N. Presmeg (Eds.), Doing qualitative research: methodology and methods in mathematics education. New York: Springer.
4. Ľ. Salanci: library sprites.js. unpublished.- toto neviem ako zmeniť
5. Creswell, J. W. (2012) Educational research: Planning, conducting, and evaluating quantitative and qualitative research. 4th ed. Boston: Pearson Education, 2012. 650 p. ISBN 978-0-13-136739-5.
6. Hrušecký, R., Tomcsányiová, M.: The development of educational web applications using the library SPRITES, DidInfo and DidactIG 2017 Banská Bystrica : Univerzita Mateja Bela, 2017. p. 64-67, ISBN 978-80-557-1216-1.
7. DagieneV., STUPURIENĖ G., VINIKIENĖ L.: Informatics Based Tasks Development in the Bebras Contest Management System. 3rd International Conference on Information and Software Technologies (ICIST 2017), Springer-Verlag Communications in Computer and Information Science.
8. Bebras Lodge Homepage, http://bebras.licejus.lt/, last accessed 2017/07/01.