Using e-Learning to describe the software development process

Inside evidence - Charles Boisvert, Norwich City College

Teachers of I.T. are continuously confronted with one problem: the production of software, web pages, or even documents is a dynamic process, which is carried out incrementally; yet to describe it, we are frequently confined to static media such as books and handouts.

The problem is particularly felt in programming, as it is at all times a difficult subject, that has to be addressed step by step. All too often, I prepare a lesson to find that even simple examples need a succession of handouts to present: a straightforward initial solution, followed by a series of improvements, until the work is explained but the students take home a barrage of materials that confuse them once they are away from live delivery.

To try and improve on that situation, I have developed a web site to present the kind of progressing material that is so common in the programming process.

I will first describe the site, then explain how its tutorials were written, and some evaluation. Finally I evaluate the materials, but also my experience of practitioner research.

Website and Tutorials

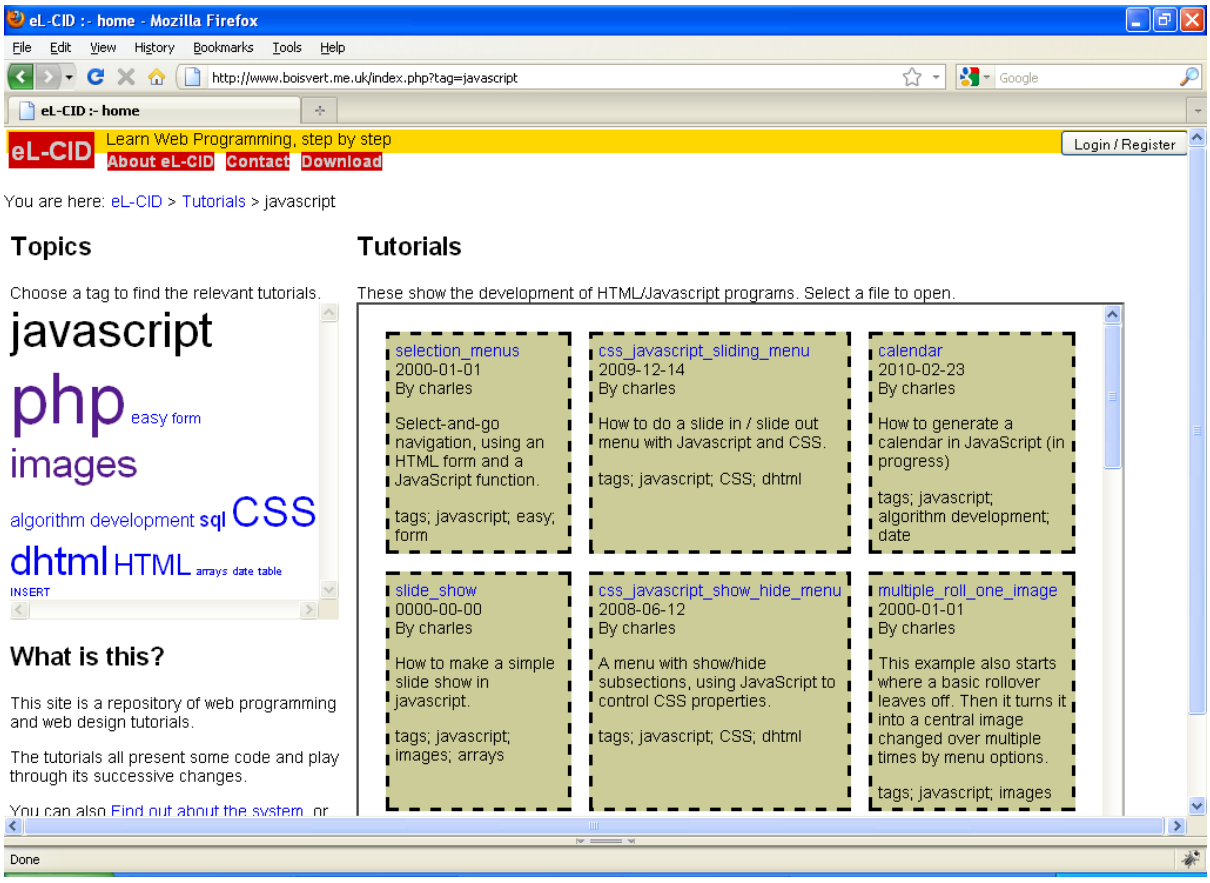

Figure 1: web site (http://www.boisvert.me.uk)

The website (figure 1) gathers over 40 web design and programming tutorials.

Each tutorial is presented as if the program code was being edited in front of the user, as shown by Figure 2.

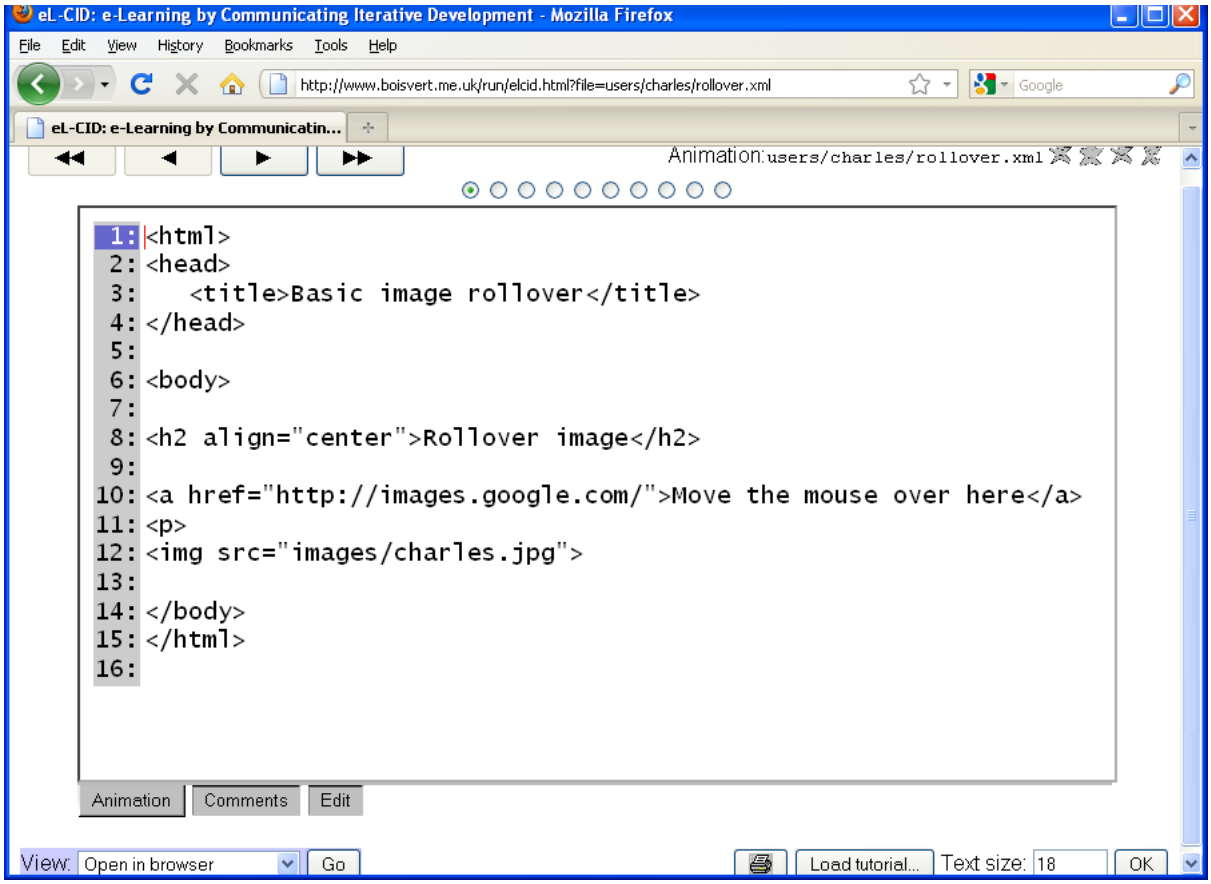

Figure 2: a tutorial

Arrow commands let users navigate, giving them control over the presentation; students may also run the code at any intermediary stage, print it, as well as print tutor comments, and finally edit the code or copy it to develop their own ideas.

In this way the tutorials offer a flexible alternative to print, pre-recorded video or live delivery.

Student Evaluation

I collected feedback from past students through web site statistics and questionnaires. The site data showed a wide diversity of usage of the tutorials, which is usually the case with tools that support discovery learning. I expected complexity of the interface to be a problem, but in practice the students navigated it without difficulty in the classroom. Questionnaire responses from the past academic year (2008-09) showed that the students mainly needed more and a wider range of tutorials.

This is not surprising given the site usage. This also coincided with my personal experience of the technique: I found that composing tutorials was a painstaking process and that it was difficult to keep up with student demand. As a result, I created editing tools to speed up the development of the site.

Teacher tools

The editor helped building new programming animations; at the time I write, 43 tutorials are available, and more are being created.

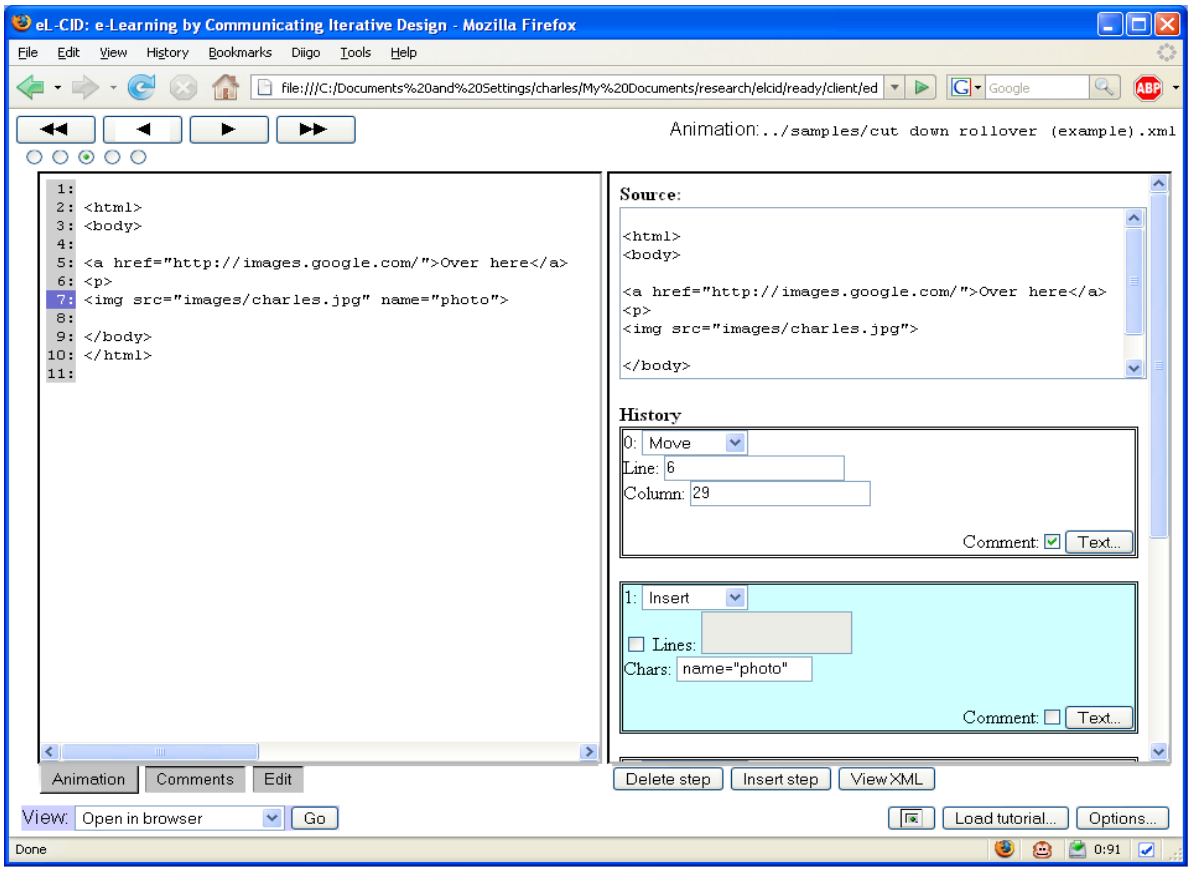

Figure 3: editing a tutorial

The online editor lets the teacher set an initial version of “source” code together with its modifications - operations like *‘move’* the cursor or *‘insert’* text. Writing a tutorial in this way turns out to be a reflexive process: as I create tutorials, I perceive the difficulties that students face when they go through a similar development process, more than if I simply wrote a single example program.

Conclusion

The website and the existing tutorials are available to all (http://www.boisvert.me.uk). They offer a diverse set of examples and help understand some aspects of web programming as part of a blended learning method.

The site, and especially, the tutorials, were a long time in the making. Remaining weaknesses are the lack of student social tools, and the poor usability of the editor for new tutorials. However, the research project has brought great benefits to my teaching, extending far beyond the materials that form a visible result.

Practitioner research has helped stay up to date in my field; sharpened my attention to student feedback; and forced me to consider the difficulties of each exercise that I made into a tutorial. I recommend the practice to any teacher.