Improving Physics Simulations with HTML5 Technologies

by

Thomas J. Bell

A proposal for a project submitted
In partial fulfillment of the
Requirements for the
Degree

of

MASTER OF SCIENCE

in

Instructional Technology & Learning Sciences

Approved by:

____
Major Professor
Douglas Holton

____
Head, Department of
Instructional Technology &
Learning Sciences

UTAH STATE UNIVERSITY
Logan, Utah
2011

Improving Physics Simulations with HTML5 Technologies

Since the inception of the W3C HTML5 working group, and its standards document the web has been a buzz about HTML5's new features and elements. HTML5 in combination with an A grade browser now offers web developers an environment or platform that reaches the power of a desktop OS. On top of faster JavaScript engines and AJAX, "The Open Web" as a platform has quickly become the development platform of choice. One of the most significant additions to the HTML5 standard is the canvas element. The canvas element allows for bitmap manipulations through a scripting interface. This yields native advantages that were only possible in the past with plugins such as Adobe's Flash player and Java applets. Web simulations created using these two plugins have received a lot of attention because they are not supported on mobile platforms. Therefore, as mobile technologies and devices spread worldwide there is significant interest in these browser native technologies. HTML5 can offer new opportunities in the area of web graphics and simulation and will be necessary to reach near ubiquitous end user access.

This project will take advantage of these new web technologies. It will use new JavaScript frameworks and innovative HTML5 elements to create visually and pedagogically sound simulations. The initial project scope will be limited to the conversion of existing physics simulations. However, the project aims to create a stable framework that will be modular and extendable. This will leave the possibility of end users creating their own simulations. This potential will not been seen with in this project because of time constraints.

Statement of problem

Technologies have gone by the wayside as the web has evolved. Adobe Director is one such technology. The shockwave player is losing support, attention and has nearly become obsolete. Adobe Director was at one point the industry standard when creating web-based simulations needing an advanced physics engine. Many of the existing interactive physics simulation have limited proliferation because of plug-in support. Mobile platforms do not support all plugins. Therefore, in order to increase proliferation and ease of use, existing Adobe Director Simulations need to be re-authored using new standards and newer well-accepted technologies. This project will convert a number of Adobe Director Simulations using HTML5's new canvas element, and the EaselJS JavaScript framework.

Needs Assessment

Physics faculties of Weber State University have found web-based simulations very useful when teaching physics courses online. However, reports of the existing simulations not working in many of the new environments have significantly increased. Students have come to depend on these simulations. They are now a critical piece of the core curriculum. The need arose to convert these simulations to a newer platform. While looking at the existing technologies, it became apparent that it was important that these simulations work on a variety of platforms and environments. HTML5 technologies as a platform met this critical requirement.

Definitions

HTML5: This specification defines the fifth major revision of the core language of the World Wide Web: the Hypertext Markup Language (HTML). In this version, new features are introduced to help Web application authors, new elements are introduced based on research into prevailing authoring practices, and special attention has been given to defining clear conformance criteria for user agents in an effort to improve interoperability (http://dev.w3.org/html5/spec/Overview.html).

HTML5 canvas element: The canvas element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly (http://dev.w3.org/html5/spec/Overview.html#the-canvas-element).

EaselJS: The new Canvas element in HTML5 is powerful, but it can be difficult to work with. It has no internal concept of discrete display elements, so you are required to manage updates manually. The Easel JavaScript library provides a retained graphics mode for canvas including a full, hierarchical display list, a core interaction model, and helper classes to make working with Canvas much easier (http://www.easeljs.com).

Assumptions

Each simulation will convert easily to the new platform. There may be simulations that will be difficult to translate to the EaselJS framework. Each simulation will receive a conversion score to prevent time loss. This score will encompass three technological and algorithmic factors of a simulation, UI needs, physics complexity, and graphics needs. The scope of this project is not necessarily concerned with the pedagogical effectiveness as it is in revising technological effectiveness and ease of use. Nevertheless, the technological conversion process allows evaluation of each simulation on a cursory pedagogical level. Creating a rubric will help define the criteria of pedagogical evaluation. Ivo Wenzler ‘s research will help in devising a rubric. The rubric will include a simplification of the Wenzler’s Ten Commandments of simulation (Wenzler, 2008).

EaselJS and the canvas element are stable and work well in all browsers. EaselJS is actively developed and is in an alpha state. This means there may be changes within the framework and the Application Programming Interfaces (APIs) will change. EaselJS version 0.3.1 will be the only version used to mitigate the effect of breaking changes to the APIs. The project will only upgrade to newer versions of the framework only if security errors are introduced. There is also a chance that interests in the EaselJS project may drop. To guard against losing code a fork of the project will be on hand and developers will maintain a local copy. Implementing a stringent testing phase will insure each simulation works in as many browsers as possible.

The complexity of the physics algorithms will easily translate from Adobe Directors built-in physics engine to pure JavaScript and canvas. Simulations that are more complex will require the implementation of the Box2dJS physics engine.

Limitations

The EaselJS library is relatively new; there are not many resources or documentation. This will become a problem if there are challenges converting graphically complex simulations. The community is small and there are only three main developers. However, this community is rapidly growing and resources are becoming available.

Inconsistent browser support for the canvas element is a major limitation. All major A-grade browsers support the canvas element except for Microsoft's Internet Explorer version 8 and below. This works against one of the goals of the project, usage on a wide variety of platforms. There are ways to overcome these inconsistencies between browsers. Using excanvas, a JavaScript library will ensure that these simulations work in all browsers. The excanvas library emulates the canvas element in Internet explorer. Microsoft has recently added support for the canvas element to Internet Explorer version 9.

Development time is always a limiting factor when converting older technologies to newer technologies. Each conversion score will help prioritize when it is to be converted. Lower scoring simulations are converted first. This will help conserve development time for work on more complex simulations.

Purpose and Objectives

Objective 1: Convert at least 10 of the existing 105 Adobe Director Simulations using the more acceptable technologies, the HTML5 canvas element and the EaselJS framework.

Objective 2: Increase the impact and usage of the simulations by making them available on more platforms specifically on iOS and android devices.

Objective 3: Combine skills learned in several classes to create a repository of learning objects that are valuable beyond the scope of this degree.

Review of Literature

There are many critics of simulations and their validity and effectiveness is constantly in question (Whitehouse, 2005). This has been a concern in recent years and many studies have researched the effectiveness of simulations in numerous problem domains (Hofstede et al., 2010; Zacharia, 2005). To that end, there is no conclusive evidence that simulations work for all problem domains or in all learning environments (Whitehouse, 2005). However, there is evidence of their strength especially when teaching physics (Balke & Scanlon, 2010; Kelly et al., 2008; Baser, 2006). It is then required to define what a simulation is and where they can be most effective within the scope of teaching physics.

> Wieman et al. (2010) explain, "A particularly important aspect of learning physics or chemistry is to develop mental models of the science. A sim can represent expert models more explicitly than other materials, by showing things such as explicit representations of electrons, vectors, or electric fields" (p. 227).

For the purpose of this project, we are defining a simulation as a web-based animation that is user configurable. In addition, as this relates to this project, users can configure and manipulate real world physical properties such as gravity or force.

Research states simulations can be highly effective in many learning environments and especially as interactive classroom demonstrations deployed on the web (Campbell et al., 2010; Wieman et al., 2010).

> Bell et al. (2008) concluded that a simulations "value… can be found in its capability to allow students to work with data, to enhance visualization of complex concepts or of unfamiliar places and objects and to facilitate communication and collaboration'' (p. 4).

Simulations provide an on-demand experience in which learners work at their own pace and control their learning (Bayrak, 2008). Simulations promote conceptual development when aligned with learning objectives and when integrated properly into general physics curriculum (Zacharia, 2003). This project has two advantages. The existing Adobe Director simulations are thoroughly tested. In addition, they align to core departmental learning objectives.

Research shows that simulations are effective within the curriculum scope of physics (Zacharia, 2005). Accordingly, it is important to investigate what is currently available. Several projects are comparable to the scope of the existing unconverted Adobe Director based simulations. The University of Colorado has created several freely available physics simulations under the project title **"PhET."** One of the technological drawbacks to the **"PhET"** simulations is that they are Java applets. This is not workable for our environment because it can limit the audience and goes against the major objective of the project. **"Physclips"** is another such project, that the University of New South Wales sponsors. Because **"Physclips"** use the Adobe flash player technology they are also platform limiting. **"Physlets"** is another interesting project created by Wolfgang Christian at Davidson College. The Physlet framework creates a bridge between Java Applets and JavaScript for use in a browser. This allows users to script their own Physlets. There is no native support for Java on mobile platforms. This eliminated **"Physlets"** as a possible platform contender for this project. There are many smaller simulations provided through merlot.org. These simulations suffer from the same technology barriers mentioned above. Throughout the investigation of these existing resources, no simulations were found using HTML5 technologies. While many of these simulations are pedagogically sound, they lack technological innovation. This project will bring these simulations up to speed while preserving their tried and true teaching foundation.

Procedures and Methodology

It is highly important that a careful analysis of the objectives is outline because this project has limited time and resources. Outlined below is a detailed description of each objective.

Design.

Convert at least 10 of the existing 105 Adobe Director Simulations using the more acceptable technologies, the HTML5 canvas element and the EaselJS framework. Several factors of this objective are quantifiable. Success will be easily determined by looking at the number of simulations converted. As part of this objective, it makes sense to discuss the selection of HTML5 technologies as a development platform. HTML5 first provides a stable native platform in several browsers. These browsers include Google Chrome, Internet Explorer 9, Mozilla Firefox 3.6, Opera 11, and Safari on all platforms. JavaScript as a scripting platform is robust and performs well. HTML5 and JavaScript is decidedly a stable solid foundation. JavaScript frameworks and helper libraries can simplify the cryptic canvas API. Selecting a JavaScript framework will aid in the development time and improve simulation performance. There are several JavaScript canvas frameworks to choose from including doodle.js, EaselJS, Mootools Canvas Library, Mootools ART, and Processing.js. EaselJS supports iOS and Android and has an API that is similar to Adobe Directors built-in scripting language called lingo. EaselJS was ultimately the best choice because of these factors.

The process of selecting a simulation to convert is as follows. Assign a conversion score to each simulation. This score is composed using three criteria UI needs, physics complexity, and graphics needs. To explain UI needs. UI components include sliders, radio buttons, checkboxes and windows. Each component is assessed one point. The number of lingo scripts, and cast behaviors will determine the physics complexity. Each script and cast member is assessed one point.

Graphic needs are determine by assessing one point for each graphic resource. After 10 simulations have been chosen the programming phase will begin. The RAD (Rapid Application Development) software development model fits this project well. The RAD model favors rapid prototyping with application planning interleaved. This model is favorable to this project because there are existing prototypes to work from. Another important factor of the RAD model is unit testing. Each simulation will go through robust tests to best accomplish the second objective relating to platform proliferation. Each simulation is given a browser compatibility rating. Testing will drive development.

Timing.

Estimates of development time will correlate to the conversion score. Because of the RAD model approach each point is roughly one hour of development time. Major tasks that are complete include platform selection, framework selection, directory structure and the HTML5 template code. With an average simulation conversion score of ten, the project will take an hundred development hours. This roughly equates to ten weeks.

Reporting Process

The expected outputs of the project are at least ten converted HTML5 simulations.

Conclusion

Proving that HTML5 and JavaScript are a solid foundation for simulations will have profound implications. When this project is successful it will open the door for new innovative simulations in other problem domains. This project has the potentially to change the way simulations are used and viewed on the web.

References

Baser, M. (2006). Promoting Conceptual Change through Active Learning Using Open Source Software for Physics Simulations. *Australasian Journal of Educational Technology*, 22(3), 336-354. Retrieved from EBSCO*host*.

Bayrak, C. (2008). Effects of Computer Simulations Programs on University Students' Achievements in Physics. *Turkish Online Journal of Distance Education*, 9(4), 53-62. Retrieved from EBSCO*host*.

Bell, R., Gess-Newsome, J., Luft,J. (2008). Technology in the secondary science classroom. NSTA Press, Arlington, VA

Blake, C. C., & Scanlon, E. E. (2007). Reconsidering Simulations in Science Education at a Distance: Features of Effective Use. *Journal of Computer Assisted Learning*, 23(6), 491-502. Retrieved from EBSCO*host*.

Campbell, T., Wang, S., Hsu, H., Duffy, A. M., & Wolf, P. G. (2010). Learning with Web Tools, Simulations, and Other Technologies in Science Classrooms. *Journal of Science Education and Technology*, 19(5), 505-511. Retrieved from EBSCO*host*.

Hofstede, G., de Caluwe, L., & Peters, V. (2010). Why Simulation Games Work--In Search of the Active Substance: A Synthesis. *Simulation & Gaming*, 41(6), 824-843. Retrieved from EBSCO*host*.

Kelly, J., Bradley, C., & Gratch, J. (2008). Science Simulations: Do They Make a Difference in Student Achievement and Attitude in the Physics Laboratory?. *Online Submission*, Retrieved from EBSCO*host*.

Wenzler, Ivo. (2009). The Ten Commandments for Translating Simulation Results Into Real-life Performance. *Simulation & Gaming*, 40(1), 98-109.

Wieman, C. E., Adams, W. K., Loeblein, P. P., & Perkins, K. K. (2010). Teaching Physics Using PhET Simulations. *Physics Teacher*, 48(4), 225-227. Retrieved from EBSCO*host*.

Whitehouse, K. (2005). Web-Enabled Simulations: Exploring the Learning Process. *EDUCAUSE Quarterly*, 28(3), 20-29. Retrieved from EBSCO*host*.

Zacharia, Z. (2003). Beliefs, Attitudes, and Intentions of Science Teachers Regarding the Educational Use of Computer Simulations and Inquiry-Based Experiments in Physics. *Journal of Research in Science Teaching*, 40(8), 792-823. Retrieved from EBSCO*host*.

Zacharia, Z. C. (2005). The Impact of Interactive Computer Simulations on the Nature and Quality of Postgraduate Science Teachers' Explanations in Physics. *International Journal of Science Education*, 27(14), 1741-1767. Retrieved from EBSCO*host*.