

Physlets: Java Tools for a Web-Based Physics Curriculum

Wolfgang Christian^{†‡}, Mario Belloni[§], and Melissa Dancy[¶]

Physics Department, Davidson College, Davidson, NC 28036, USA.
wochristian@davidson.edu, <http://webphysics.davidson.edu>

Abstract. An approach to developing curricular material that couples a software design philosophy with physics education research (PER) is described. It is based on open Internet standards such as Java, JavaScript, and HTML as well as research into the effectiveness of computer-based physics instruction.

1 Overview

“Good educational software and teacher-support tools, developed with full understanding of principles of learning, have not yet become the norm.”
How People Learn: Brain, Mind, Experience and School from Committee on Developments in the Science of Learning, National Research Council National Academy Press, 1999.

The impact of instructional software on mainstream physics instruction has, at present, been minimal. At American Association of Physics Teachers (AAPT) meetings in the 1980s, it was common to see participants sharing floppy disks and trading software for the computer-enabled educational reform that everyone knew was sure to come. It didn’t, at least not in the form envisioned by the conference participants. Little of the early educational software was adopted by the mainstream teaching community and almost none of it is still being used today. In contrast, printed material from the much earlier post-Sputnik curricular reform movement — the Berkeley Physics series, for instance — is still available and useful to physics educators, although the pedagogy upon which it was based has gone out of fashion. Will this scenario be repeated and are we doomed like the Greek hero Sisyphus to forever push computational physics up the hill of curriculum reform? Can we expect widespread adoption of computation in the current curricular reform initiative? And, if so, what strategies should we adopt to insure that computational-rich curricula being developed today will be adopted and be in widespread use a decade from now?

[†] email: wochristian@davidson.edu

[§] e-mail: mabelloni@davidson.edu

[¶] e-mail: medancy@davidson.edu

[‡] Supported in part by National Science Foundation grant DUE-9752365.

It is not surprising that many thoughtful teachers are unwilling to invest the time and energy needed to incorporate computational and educational software into their curriculum. Learning the intricacies of a software package is a poor investment of time for instructors, authors, and publishers if the half-life of the textbook publication cycle is longer than the half life of the computer technology. However, the case can now be made that this throwaway cycle for educational software need not repeat itself and that key technologies are available that enable authoring and distribution of curricular material that will withstand the test of time. Commercial applications have, in effect, provided the education community with a rich and flexible set of standards that are likely to prevail. This paper describes an approach based on virtual machines, meta-languages, and open Internet standards that couples a software design philosophy with research into the effectiveness of computer-based instruction.

2 Physlets

Physlets – “Physics applets” – are small, flexible Java applets that can be used in a wide variety of applications [1]. Many other physics-related Java applets are being produced around the world – some of them very useful for education. However, the class of applets that we call “Physlets” has several attributes that make it valuable for educators.

1. Physlets are pedagogically neutral. Physlets can be used as an element of almost any curriculum with almost any teaching style. Because of their dynamic interactivity, Physlets are ideally suited for interactive engagement methods [2–4] such as Just-in-Time Teaching [5], Peer Instruction [6], and Tutorials [7]. In addition, Physlets can also be used as traditional lecture demonstrations and can be given as end-of-chapter homework.
2. Physlets are simple. The graphics are simple; each Physlet problem should be designed to involve only one facet of a phenomenon, and should not incorporate very much in the way of a user interface. This use keeps Physlets relatively small and eases downloading problems over slow network connections, and removes details that could be more distracting than helpful.
3. Physlets are flexible. Both physical and non-physical situations can be created. All Physlets can be set up and controlled with JavaScript, meaning that a dynamics modeler, such as the Animator Physlet, can be used for almost any subject in mechanics with small changes in the JavaScript and not the Java code. Data analysis can be added when needed with a second Physlet using inter-applet communication.
4. Physlets are web-based. They can run on (almost) any platform and be embedded in almost any type of HTML document, whether it be a homework assignment, a personal web site, or an extensive science out-reach site.
5. Physlets encourage collaboration because they are free for noncommercial use at educational institutions. Physlet archives, that is, compressed archives containing compiled Java programs, can be easily downloaded from the

Davidson College WebPhysics server:
<http://webphysics.davidson.edu/applets/applets.html>.

3 Educational Software Design

The Model-View-Control (MVC) design pattern is one of the most successful software architectures ever invented. It is the basis for the Smalltalk programming language and was used extensively in designing user-interface components for the Java 2 platform. It is well suited to the design of interactive curricular material using Physlets. In this design pattern, the model maintains data and provides methods by which that data can change; the control gives the user the ability to interact with the model using input devices such as the keyboard and the mouse; and the view presents a visual representation of some or all of the model's data. Although there is usually one model and one control, there will often be many views.

A compiled object oriented language, such as Java, should be used to implement models and views because these objects are often very complex and computational speed is important. They are best implemented as separate applets. But the control object is different. The control object is accessed infrequently in comparison to a CPU clock cycle and need not be compiled. More importantly, it should be customizable by the curriculum author because it is difficult for a programmer to anticipate every author's needs. A curriculum author would certainly expect to graph relevant physical quantities as an object moves, but it would be difficult for a programmer to anticipate the various combinations that are needed. For example, will an author want to plot position, energy, or force? What if there are dozens of objects? A scripting language such as JavaScript embedded into an HTML document provides an ideal solution to this problem because the author can change the behavior to suit his or her needs.

Another advantage of using JavaScript to control a model and its views is that control can be distributed through the narrative. The ubiquitous HTML anchor tag — the tag that usually takes the reader to another HTML document — can also be used to execute JavaScript. So can HTML buttons, timers, and user-initiated events such as resting a mouse pointer on an image. In contrast, a monolithic Java applet that combines model, view, and control would remain virtually unchanged from one pedagogic context to the next. HTML augmented with JavaScript enables Physlets to share a common user interface that eases the learning curve across pedagogic contexts.

4 Scripting

The pendulum provides a good example of how to use the MVC pattern to create a Physlet-based exercise. The pendulum model consists of a system of three first-order coupled differential equations. The ODE Physlet solves this

model and the solution is passed to the two views shown in Figure 1 using inter-applet communication. The view on the left shows an animation of a red ball at the end of string. The view on the right is a phase space plot.

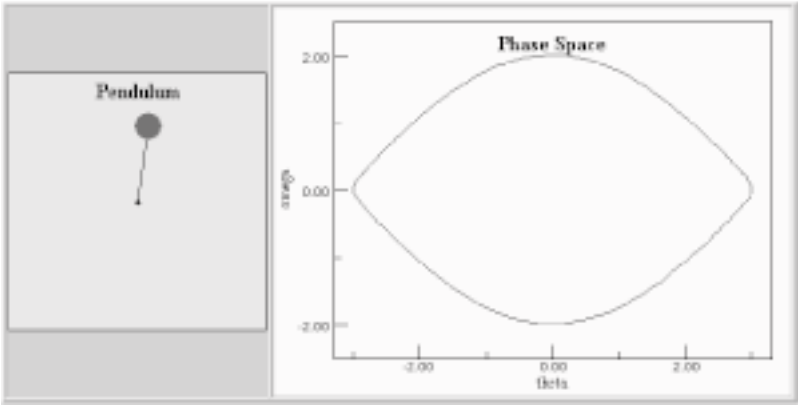


Fig. 1. Pendulum simulation with two data views.

Before an applet can be scripted, it must be properly embedded into an HTML page. Embedding an applet is similar to embedding an image. For example, the DataGraph applet—a typical data-analysis Physlet—is embedded using the following *applet* tag:

```
<applet
  codebase="../../../classes/"
  archive="DataGraph4_.jar,ODE4_.jar,STools4.jar"
  code="dataGraph.DataGraph.class"
  name="plot_view" width="400"
  height="300">
  <param name="AutoScaleX" value="true">
  <param name="AutoScaleY" value="true">
  <param name="ShowControls" value="true">
</applet>
```

The codebase attribute specifies the directory where the files are located, the archive attribute specifies which files are needed to run the simulation, and the code attribute specifies the object that contains the entry point to the applet. In this case we are running the DataGraph Physlet. This applet can now be referenced by its logical name, *plot_view*, when using JavaScript. Other applets are embedded in a similar fashion except that the ODE applet has zero height and zero width because it does not have an on-screen representation; it merely solves the pendulum’s differential equation.

After an applet is embedded, JavaScript is used to invoke its methods using typical object-oriented dot notation starting with the `document` object as the container. For example, *plot_view* is initialized as follows:

```
document.plot_view.setLabelY("omega");
document.plot_view.setLabelX("theta");
document.plot_view.setTitle("Phase Space");
```

Almost all Physlets support one or more `addObject` methods designed to instantiate Java objects inside the applet. This method has the following signature:

```
addObject(String name, String attributes);
```

These objects usually, but not always, have an on-screen representation. The first argument is the name of the object to be added, and the second is a comma-delimited list of parameters. An important feature of the `addObject` method is that it returns a unique integer identifier, `id`, that can later be used to change properties of the object. For example, the circle shown in Figure 1 was added to the *animation_view* using the following JavaScript statements:

```
ball_id=document.animation_view.addObject(
    "circle", "x=0,y=-1.0,r=10");
document.animation_view.setRGB(ball_id, 255, 0, 0);
```

The `addObject` method is very forgiving. Parameters can appear in any order and not all parameters need to be specified. Default values are overridden only if the parameter appears in the list. Incorrect and unsupported parameters do not affect the applet and are ignored.

Systems of first-order ordinary differential equations (ODE) can be incorporated into a simulation using the ODE Physlet. For example, the script that models the pendulum simulation can be written as:

```
document.ode.setDefault();
document.ode.addObject(
    "ode", "var=t, value=0, rate=1");
document.ode.addObject(
    "ode", "var=theta, value=3, rate=omega");
document.ode.addObject(
    "ode", "var=omega, value=0, rate=-sin(theta)");
document.ode.parse();
```

The ODE Physlet must now pass the variable values `t`, `theta`, and `omega` to the two views, the plot and the animation.

Inter-applet communication is performed directly in Java, but the connection must first be established using script. The data-source object in the sending applet must implement the `SDataSource` interface and the receiving object must implement the `SDataListener` interface. Both of these objects register their capabilities in a superclass common to all Physlets. JavaScript can then be used

to set up a data connection between the source and the listener using a method with the following signature:

```
makeDataConnection(int sid, int lid, int series,  
    String xfunction, String yfunction);
```

The first two parameters, *sid* and *lid*, are integer identifiers for the data source and data listener objects, respectively. The third parameter is a user-defined number that can be used by the data listener to keep track of multiple data sets. The last two parameters are strings representing mathematical functions of any of the data source variables. These functions are evaluated as data passes through the connection so that a single datum, (xfunction, yfunction), is delivered to the listener. For example, the JavaScript necessary to pass the omega and theta variables from the ODE applet to the animation view and the plot view is

```
graph_id=document.graph_view.getGraphID();  
ode_id=document.ode.getSourceID();  
document.ode.makeDataConnection(  
    ode_id,graph_id,1, "theta","omega");  
document.ode.makeDataConnection(  
    ode_id,ball_id,1, "sin(omega)","cos(omega)");
```

The first two JavaScript statements are necessary to retrieve the integer identifiers for the Java objects that will receive the data. The inter-applet communication mechanism is very fast (because it is implemented in Java) and flexible (because it is implemented in JavaScript). The only change that needs to be made to a script in order to change the data being delivered is to edit the functions, that is, the two string parameters.

5 Curricular Innovation

We have developed close to one thousand Physlet-based problems over the past four years in support of a number of introductory physics texts. A selection of these problems is available on the CD that accompanies the *Physlets* book [1]. More importantly, the Physlets upon which these problems are based are freely distributable for non-commercial educational purposes and are now being adapted to support various curriculum reform initiatives.

It would be foolish to predict the future direction of the computer industry and its impact on education. For example, streaming video is currently a hot technology, and both traditional broadcasters and software companies are competing to establish themselves in this market. However, research has shown that merely watching video has little effect on student learning, and it is unlikely that streaming video will change this result. Small cognitive effects have been shown to occur using video clips if the showing of the clip is accompanied with in-class discussion or if the clip is used for data taking and data analysis [8]. Similarly, database technology has become ubiquitous in our society. It is used to store consumer-shopping profiles for corporate marketing departments

and to manage Christmas card mailing lists at home. But little has been gained in attempts to tailor the curriculum to individual learning styles. Other high-end technologies, such as virtual reality, three-dimensional modelling, and voice recognition, will almost certainly come on-line in the coming decade. However, their most enduring effect on education may be to drive the price/performance ratio of consumer, and hence educational, hardware even lower. These technologies are unlikely to have a significant impact on undergraduate education without a corresponding curricular-development effort and research into their pedagogic effectiveness. Current commercial technologies may, in fact, already be good enough to implement the most effective teaching strategies. Unlike previously written educational software, software written using Internet standards, such as Java and JavaScript, should be accessible for years to come. For computation to have a long lasting impact on science education, it will also need to be based more on successful pedagogy than on the latest compilers, hardware, or algorithms.

5.1 PER: Physics Education Research

Physics Education Research, PER, informs us that technology does not necessarily lead to improved learning and that we are just beginning to understand how it is best used. Two PER researchers, Aaron Titus [9] and Melissa Dancy [10], have used Physlets to study the effect of animation on student assessment and student problem solving ability. Their research focuses on students, not on the Physlets themselves.

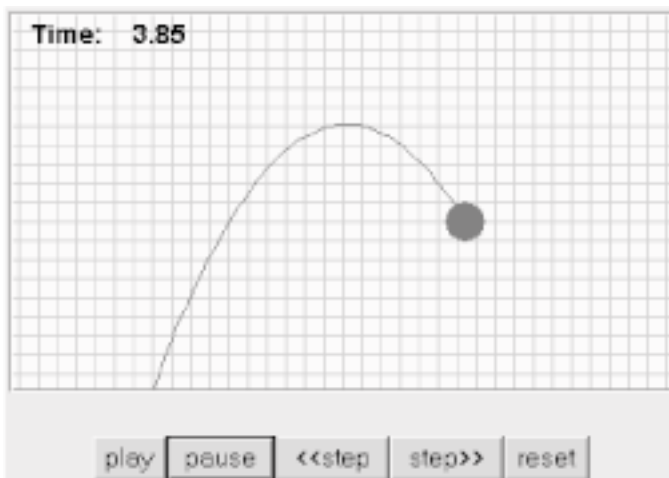


Fig. 2. A media-focused projectile problem.

Titus measured student attitudes and problem-solving approaches while they were solving Physlet-based problems [9]. The study distinguishes between media-enhanced problems where multimedia is used to present what is described in the text, and media-focused problems, where the student must use multimedia elements in the course of solving the problem. Titus found that media-focused problems are fundamentally different from traditional physics problems, and Physlets are ideally suited for these types of problems. Consider an example from kinematics. A traditional projectile problem states the initial velocity and launch angle and asks the student to find the speed at some point in the trajectory. This problem can be media-enhanced by embedding an animation in the text, but this adds little to the value of the problem. Alternatively, this same problem could be a media-focused Physlet problem as shown in Figure 2. In this case, no numbers are given in the text. Instead, the student is asked to find the minimum speed along the trajectory. The student must observe the motion, apply appropriate physics concepts, and make measurements of the parameters he or she deems important within the Physlet. (A mouse-down enables the student to read coordinates.) Only then can the student “solve the problem.” Such an approach is remarkably different from typical novice strategies where students attempt to mathematically analyze a problem before qualitatively describing it (an approach often called “plug-and-chug” and characterized by a lack of conceptual thought during the problem-solving process).

Dancy used Physlets to probe students’ conceptual understanding by using a standard diagnostic instrument, the Force Concept Inventory [12], in which all thirty static pictures (see Figure 3) were replaced by Physlet-based animations (see Figure 4) [11]. Both quantitative and qualitative data was collected from hundreds of students using the Physlet-based version and the results were statistically analyzed. The study showed that Physlet-based problems are less likely to elicit memorized responses because they allow students to respond to what they see, rather than what they read. Physlets tap into students’ intuition and deeply-held misconceptions by eliminating the additional step of translating from words or graphs. In general, students had a better understanding of the intent of the questions when viewing an animation and gave an answer that was more reflective of their actual understanding. Dancy speculates that this may be because the animation looks more like real life than something from a physics textbook.

Both the Titus and the Dancy studies indicate that while computer-based animation can be used for cosmetic and motivational purposes, they are most effective under the following conditions:

1. The animation is integral to the question.
2. The student must interact with the animation to obtain data.

The effectiveness of Physlets likely depends on many factors such as how well the task targets known student difficulties, how students use visual cues given by the Physlet, how important visualization is to the given task, and the appropriateness of the Physlet to the given task. But both studies show that conceptual understanding is key to solving Physlet problems. Without strong conceptual

understanding, students are prone to guess, search for the “right” equation, and lack direction.

Physlet problems generally cannot be correctly solved using “plug-and-chug” methods. The fact that data is not given in the text of the problem requires that students apply proper conceptual understanding to the solution before analyzing data. Therefore, it also seems that Physlet problems may be useful for encouraging a “concept-first” approach to solving problems, where students consider the concepts or principles to be applied to the problem before making calculations.

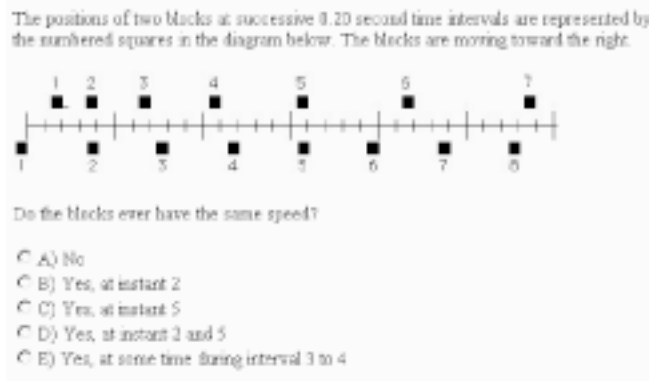


Fig. 3. A text-based Force Concept Inventory question: The positions of two blocks at successive 0.2-second time intervals are represented by the numbered squares in the diagram. The blocks are moving toward the right. Do the blocks ever have the same speed?

5.2 Just-in-Time Teaching

Although the media-rich content and interactivity provided by technology such as Physlets can be pedagogically useful, it can lack the human dimension that is important to effective teaching. Computer Assisted Instruction (CAI) has already been tried on very elaborate proprietary systems. It is unlikely to be improved significantly by being ported to the Internet. To be truly effective, the communication capabilities of the computer must be used to create a feedback loop between instructor and student. A new and particularly promising approach known as Just-in-Time Teaching, JiTT, has been pioneered at Indiana University and the United States Air Force Academy and further developed at Davidson College [5]. It employs a fusion of high-tech and low-tech elements. On the high-tech side, it uses the World Wide Web to deliver multimedia curricular materials and manage electronic communications between faculty and students. On the low-tech side, the approach requires a classroom environment that emphasizes

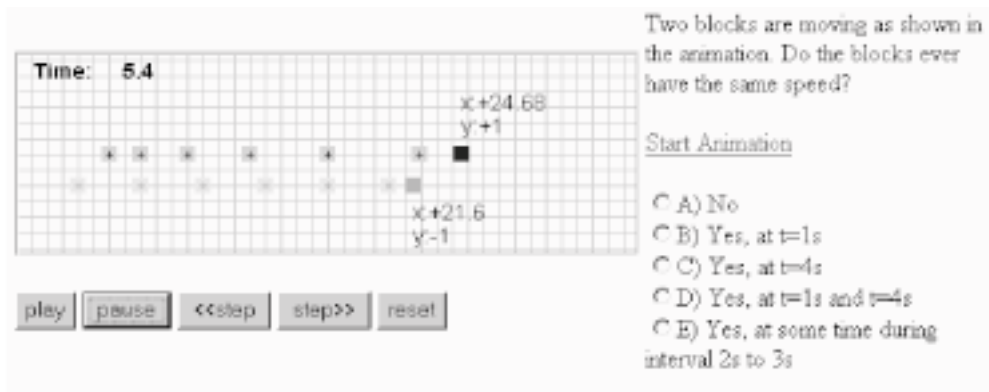


Fig. 4. A Physlet-based Force Concept Inventory question: Two blocks are moving as shown in the animation. Do the blocks ever have the same speed?

personal teacher-student interactions. These disparate elements are combined in several ways, and the interplay produces an educational setting that students find engaging and instructive. The underlying method creates a synergy between the Web and the classroom to increase interactivity and allow rapid response to students’ problems.

The JiTT pedagogy exploits an interaction between Web-based study and an active-learner classroom. Essentially, students respond electronically to carefully constructed Web-based assignments, and the instructor reads the student submissions “just-in-time” to adjust the lesson content and activities to suit the students’ needs. Thus, the heart of JiTT is the ‘feedback loop’ formed by the students’ outside-of-class preparation which fundamentally affects what happens during the subsequent in-class time. The students come to class prepared and already engaged with the material, and the faculty member already knows where classroom time can best be spent.

Although JiTT can be implemented fully using technically simple Web-based assignments, incorporating Physlet-based questions can heighten the extent to which student understanding can be probed and encouraged. The JiTT strategy as applied in physics education is richer for the incorporation of Physlets. Consider, for example, the puzzles shown in Figures 5 and 6. Students are expected to analyze each situation, apply the relevant physics, and answer specific questions. The faculty member then prepares a lecture in response to the student submissions.

It is interesting to compare the questions. The static Puzzle (Figure 5) involves nearly the same physics as the Physlet-based Puzzle (Figure 6), but what is required of the student in order to solve each puzzle is quite different. In each case, the student must understand the concepts of moment of inertia, torque, angular acceleration, angular velocity, and the relationships between those quantities. In each case, it also behooves the student to draw free body diagrams to

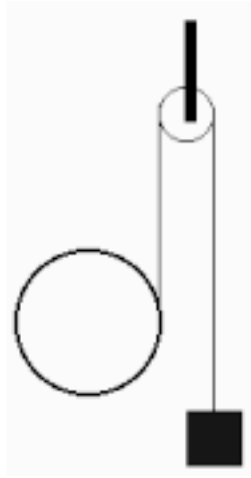


Fig. 5. JiTT Yo-Yo Puzzle Question: Make yourself a yo-yo by wrapping a thin string around a thin hoop of mass M and radius R . Pass the string around a pulley and attach it to a weight, whose mass is exactly half the mass of the hoop. Then release the system from rest. Describe the subsequent motions of the yo-yo and the weight. You may use equations to arrive at your answer, but you must state your result in plain sentences.

consider the forces involved. The static Puzzle involves the concept of rolling without slipping (because of the pulley) and can be solved completely with equations and subsequent English sentences of explanation. The dynamic Puzzle, however, requires some visual analysis and understanding of how the speed with which the mass falls is related to the physics quantities such as angular momentum and moment of inertia. It is clear from use of both static and Physlet-based questions that students who understand how to solve one of these sorts of questions do not necessarily know how to solve the other, so incorporating both types is an effective way to broaden and deepen all the students' understanding.

6 Conclusion

Based on our results, we believe that Physlets can be a valuable tool for creating interactive curricular material designed around the needs of the student. We have investigated using Physlets to alter existing curricular material. However, the greatest potential of Physlets will probably come from using Physlets to ask questions in ways which can not be done on paper.

When developing a Physlet-based problem it is important to ascertain a clear instructional purpose. Using a Physlet cosmetically to merely enhance visualization of a question is gratuitous. For maximum benefit, Physlet questions should require students to interact with the simulation. Students should be required to

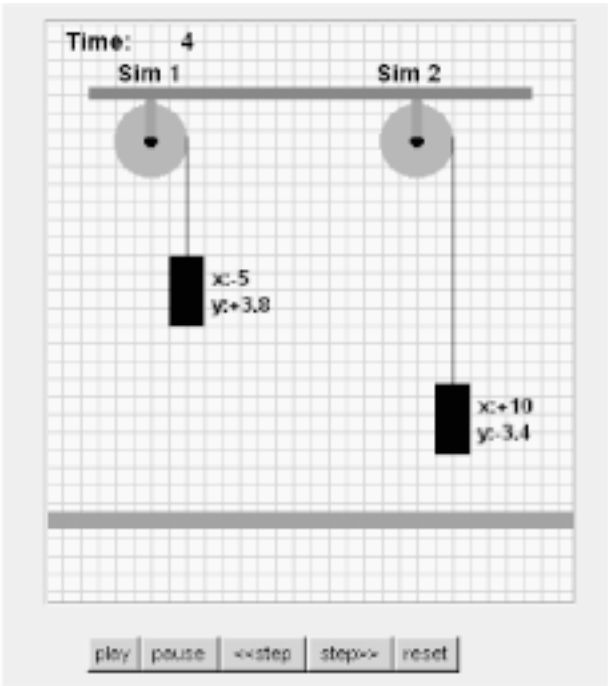


Fig. 6. Physlet-based JITT Moment of Inertia Puzzle Question: Rank simulation 1 and simulation 2 from least to greatest in terms of the moment of inertia of the wheel, the tension in the string, and the total angular momentum about the wheel’s axle after 4 seconds. The hanging weights have identical mass.

collect data, either numerically or visually. Research shows that if interaction is required, Physlets may influence how a student responds. Therefore, Physlet-based problems may be a more valid way to measure conceptual reasoning. The media-focus also makes them more challenging than traditional problems since novice solution strategies leave the student in despair.

Physlet problems are dynamic problems. Not only do they help students visualize a situation, they encourage the student to solve a problem the way a physicist solves a problem; that is, to consider the problem conceptually, to decide what method is required and what data to collect, and finally to analyze the data. It is akin to an open-ended laboratory experiment where students are not given instructions, but merely a question. They must decide what data to collect and how to most efficiently collect it. This quality seems to make Physlets well suited for evaluating students’ application of conceptual understanding to numerical problems and helping students identify weaknesses in conceptual understanding.

7 Acknowledgements

Portions of the work presented here are based on published and unpublished work in collaboration with Aaron Titus and Evelyn Patterson. We would also like to thank Harvey Gould and Larry Cain for the many helpful comments in reviewing this manuscript.

The authors would like to acknowledge the National Science Foundation, grant DUE-9752365, for its support of Physlets.

References

1. *Physlets: Teaching Physics with Interactive Curriculum Material*, W. Christian and M. Belloni, Prentice Hall, Upper Saddle River, 2001; <http://webphysics.davidson.edu/applets/applets.html>.
2. R. R. Hake, *Interactive-engagement vs. traditional methods*, Am. J. Phys., **66**, 64–74 (1998);
T. E. Sutherland and C. C. Bonwell, eds., *Using active learning in college classes; a range of options for faculty*, Jossey-Bass, San Francisco (1996).
3. D. R. Sokolo, *Using Interactive Lecture Demonstrations to Create an Active Learning Environment*, The Physics Teacher, **35**, 340 (1997).
4. B. Thacker, *Comparing Problem Solving Performance of Physics Students in Inquiry-based and Traditional Introductory Courses*, American Journal of Physics, **62**, 627–633. (1994).
5. *Just-in-Time Teaching: Blending Active Learning with Web Technology*, G. M. Novak, E. T. Patterson, A. D. Gavrin, and W. Christian, Prentice Hall, Upper Saddle River, 1999.
6. *Peer Instruction: A Users Manual*, E. Mazur, Prentice Hall, Upper Saddle River, 1997.
7. *Tutorial in Introductory Physics*, L. McDermott and P. S. Shaffer, Prentice Hall, Upper Saddle River, 1998.
8. R. Beichner, *The Impact of Video Motion Analysis on Kinematics Graph Interpretation Skills*, American Journal of Physics, **64**, 1272–1277. (1997).
9. A. Titus, *Integrating Video and Animation with Physics Problem Solving Exercises on the World Wide Web*, Ph.D. dissertation. North Carolina State University: Raleigh, NC. (1998).
10. M. Dancy, *Investigating Animations for Assessment with an Animated Version of the Force Concept Inventory*, Ph.D. dissertation. North Carolina State University: Raleigh, NC. (2001).
11. M. Dancy and B. Beichner. *Does Animation Influence the Validity of Assessment?*, (in preparation for the Journal of Research in Science Teaching).
12. D. Hestenes, M. Wells, and G. Swackhamer. *Force Concept Inventory*, The Physics Teacher, **30**, 141–158 (1992).