USE OF EDUCATIONAL TECHNOLOGY „PHYSLETS" IN PHYSICS EDUCATION

Arnis Voitkans
University of Latvia, Latvia

Abstract

Use of ICT can provide many opportunities for physics education, and one of these opportunities is a use of physics simulations. Physicists professionals use complex simulations to solve real world problems, but for educational use simulations/animations not always have to be so complex. More lightweight simulations and animations can be used. Such simulations usually are more easily distributable through Internet. Simulations and animations are important in physics education because they can help to visualize different physical processes and to provide some interactivity.

This paper is focusing on a use of Physlets – small portable Java Applets, which can be used to simulate and animate different physical phenomena. Purpose of this paper is to give a brief overview of Physlets and to demonstrate in few examples the possibilities of adaption and customization of Physlets in learning material.

Keywords: *Physlets*, *Educational physics simulations*.

Introduction

Nowadays there are many different resources related to physics and not only physics education in Internet. People develop different peaces of content, simulations, animations and share with them. One of the biggest repository of such educational resources is MERLOT (Multimedia Educational Resource for Learning and Online Teaching) [1]. MERLOT provides a peer reviewed resources for different disciplines of education, including physics.

In physics education few of the most remarkable resources are **PhET** (Physics Education Technology at the University of Colorado), which provide a collection of simulations and virtual laboratories [2] and **Physlets** [4]. Both of these resources are marked as *MRLOTS Editor's Choice*. MERLOT provides many other resources related to physics education, but in this paper the main focus is on Physlets.

Physlets are small and portable **Phys**ics Java app**lets** – hence the name, and are copyrighted to Wolfgang Christian. Since they are designed for the use in web, it is easy to integrate them in online physics materials, particularly in HTML pages.

Physlets have some advantages for educational use:

- Physlets are simple and lightweight, that means they are fast to download and execute, and there are fewer distracting details [3].
- Physlets are technologically flexible:
  - Since they are using Java technology they can be opened and viewed in many browsers on many operating systems.
  - There is no need to do Java programming, it is enough to use JavaScript directly in HTML pages to script different problems.
- Physlets are pedagogically flexible:
  - Physlets can be used in different levels of education, for example, in primary and secondary schools. Some Physlets are suitable for university level education, for example, those related to quantum mechanics. It is possible to create many very different simple and complex physical problems for education.
  - Physlets can be adopted for different didactical purposes. For example, they can be integrated in a "traditional" material to enrich the content, or they can be used in tests, or just like a spark for discussions between students.

- Physlets are freely usable and distributable for non-commercial use.

The target audience of this paper are physics teachers with informatics background or any other physics teacher who want to incorporate interactive elements in their physics materials, with some technical help can adopt and use Physlets in their educational material.

Overview of Physlets technology

In this chapter a brief overview of the technological aspects of Physlets is given. This overview is intended just to illustrate the main concepts of Physlets, how do they work in general, and this information is not extensive. For a more detailed information please look in the online documentation [6].
As it was mentioned before, Physlets are based on Java technology and are implemented as Java classes. There are many Physlets Java classes. Few of them are mentioned below:

- **SApplet4** – this class incorporates all general methods which are in all Physlets, for example, methods related with clock and time. This allows, for example, to pause the clock at the current time or reset it in every Physlet. SApplet4 class also incorporates methods related with data connections between different Physlets so it would be possible to send and receive data. SApplet4 is inherited by all other Pyslets classes.
- **Animator4** is probably the most versatile Physlet [3]. It is designed to animate shapes and images on the screen. For example, it is possible to animate the trajectory of thrown ball. Of course, it is one of the simplest uses of this Physlet, and much more complicated animations are possible. See Figure 1 for example.
- **EField4** is designed for a use in electrostatics. It is possible to add charged particles and to observe forces which are acting on the particles. It is also possible to visualize potentials and forces around charged bodies in a real time while dragging particles (Figure 2).

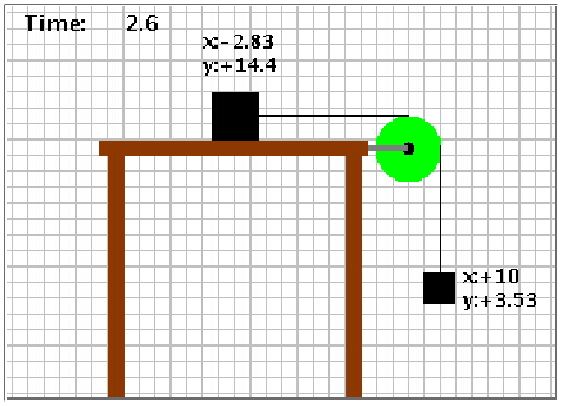


Figure 1: More complicated animation with two masses ([4], problem 8.6.5)

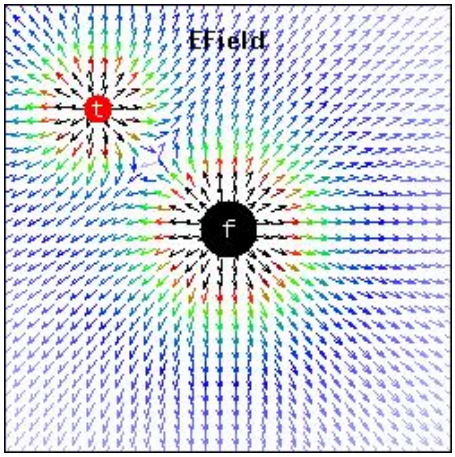


Figure 2: Example of EField Physlet – two charged particles and electrical field around them [4]

- **Optics** Physlet allows to set an optical workbench using different elements – lenses, mirrors, dielectrics, different sources of light – and then to see how light rays are propagating through these elements. Optics Physlet can be very interactive – it is possible to drag optical elements around and to see how propagation of light rays are changing (Figure 3).

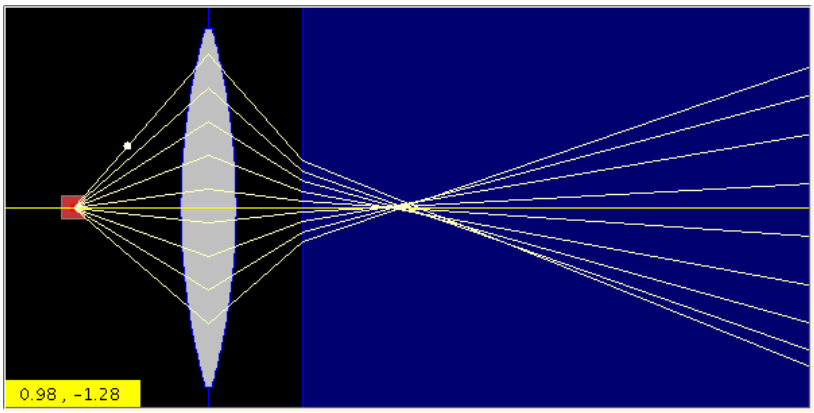


Figure 3: Light source, lens and dielectric [4]

- **DataGraph** is used to graphically show data from other Physlets or functions (Figure 4). It is possible, for example, to connect DataGraph with Animator and to show components of the speed of the motion (Figure 5).

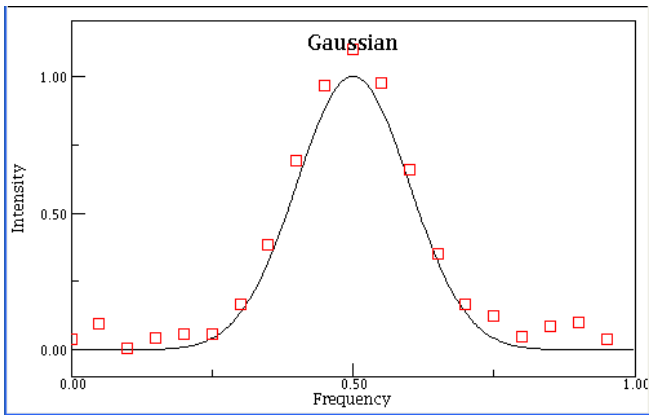


Figure 4: Datagraph with generated data [4]

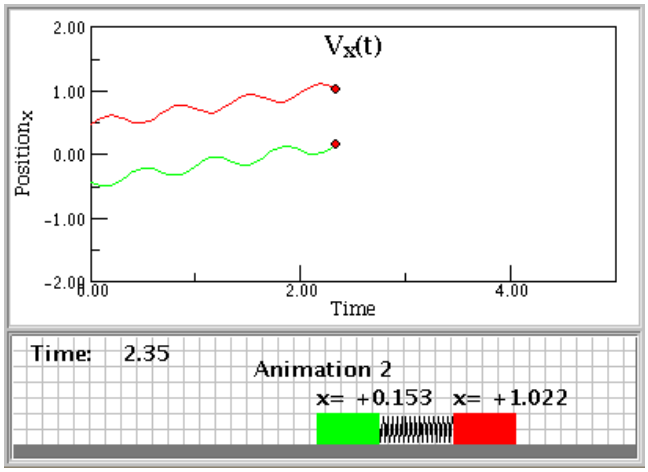


Figure 5: DataGraph Physlet in conjunction with Animator ([4], problem 7.1.4)

- **DataTable** allows to collect and show data form physical processes simulated/animated in other Physlets in a row-column format (Figure 6).

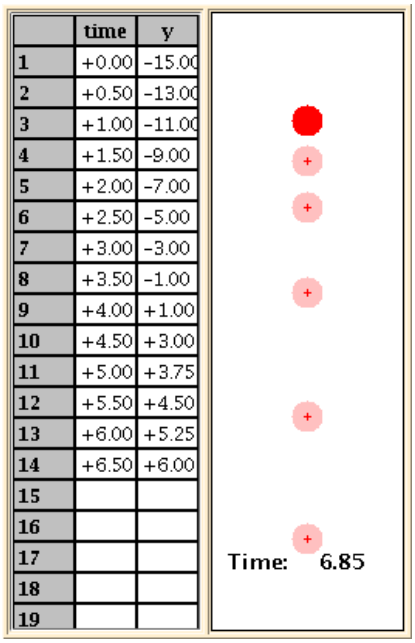


Figure 6: DataTable in conjunction with Animator

Physlets examined here are just few of much bigger collection. Interesting Physlets like Circuit, Faraday, Doppler and many other are not included in this overview. For a full list and description of Physlts please look in Physlets HomePage [4].

Example of adaptation of Physlet

Let's look at a simple example how to adapt and modify Physlet. For the purpose of this example let's choose one of the Physlet problems form Physlets HomePage [4] – let it be "Tour of Physlets: Example 11" (Figure 7), but it could be any other problem.

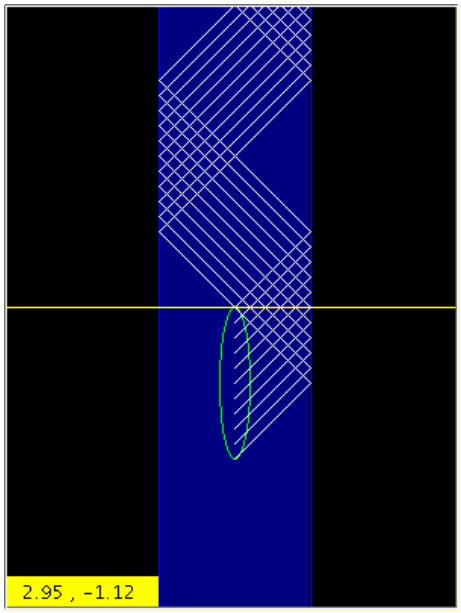


Figure 7: Demonstration of inner reflection in dielectric (Tour of Physlets: Example 11, [4])

Original HTML source code of this Physlet is:

```
<html>
  <head>
    <title>Tour of Physlets: Example 11</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <style></style>
        <script language="JavaScript">
        function fiber(){
                document.OpticsApplet.setDefault();
                document.OpticsApplet.addIndexChange(1.0,1.4);
                document.OpticsApplet.addIndexChange(2.0,-1.4);
                document.OpticsApplet.setDirection(1);
                document.OpticsApplet.setPropertyDrag(true);
                document.OpticsApplet.setDrag(true);
                document.OpticsApplet.setRayRGB(255,255,255);
                document.OpticsApplet.addISource(1.5,-0.5,100,-1.0);
        }
        </script>
  </head>
  <body onload="fiber()">
        <applet codebase="."
                archive="Optics4_.jar,STools4.jar"
                code="optics.OpticsApplet.class"
                name="OpticsApplet" id="OpticsApplet"
                width="300" height="400" hspace="0" vspace="0"


align="middle">
        <param name="ShowControls" value="false">
        </applet>
</body>
</html>
```

Code 1: Original script code of "Tour of Physlets: Example 11"

In the codebase option in <applet> tag path to Physlets classes (Optics4_.jar,STools4.jar) should be specified. Considering that in this example files Optics4_.jar,STools4.jar are in the same directory as the HTML file, path is "."
Now let's increase dielectricity of the central dielectric to 1.5 and let's move dielectric little bit closer to the left side, so lines

```
        document.OpticsApplet.addIndexChange(1.0,1.4);
        document.OpticsApplet.addIndexChange(2.0,-1.4);
```

should be changed to:

```
        document.OpticsApplet.addIndexChange(0.8,1.5);
        document.OpticsApplet.addIndexChange(1.8,-1.5);
```

Let's also move the light source little closer to the left side and lower it a bit. We will also change the angle of the light (the last number -1.0 corresponds to -45 degrees). Let's change

```
document.OpticsApplet.addISource(1.5,-0.5,100,-1.0);
```

to:

```
document.OpticsApplet.addISource(1.3,-1,100,-0.3);
```

Finally let's add a new element – mirror . This directive has to be added to the JavaScript code:

```
document.OpticsApplet.addMirror(2.5,0);
```

The result is shown in Figure 8.

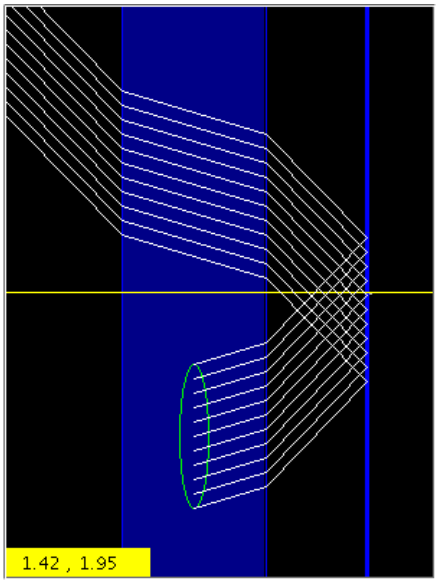


Figure 8: Modified example Physlet – added mirror and changed the angle of the light source

Resulting modified JavaScript code is:

```
<script language="JavaScript">
function fiber(){
document.OpticsApplet.setDefault();
document.OpticsApplet.addIndexChange(0.8,1.5);
document.OpticsApplet.addIndexChange(1.8,-1.5);
document.OpticsApplet.setDirection(1);
document.OpticsApplet.setPropertyDrag(true);
document.OpticsApplet.setDrag(true);
document.OpticsApplet.setRayRGB(255,255,255);
document.OpticsApplet.addISource(1.3,-1,100,-0.3);

document.OpticsApplet.addMirror(2.5,0);
}
</script>
```

It is still possible to interactively drag optical elements and to change their parameters (see Figure 9).

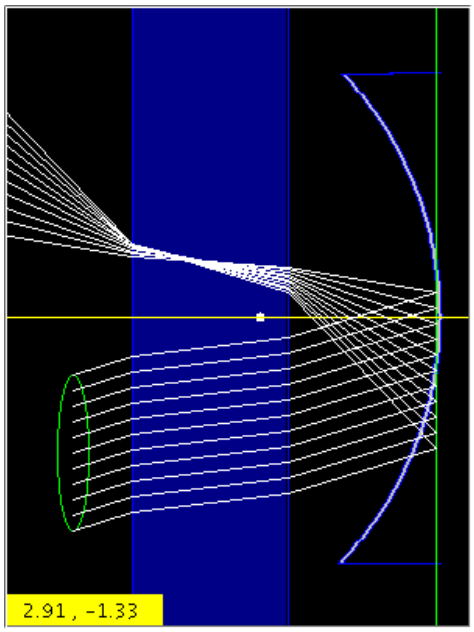


Figure 9: Interactively modified optical elements – mirror now is spherical and the light source is outside the dielectric

Please consult the Physlets documentation [6] when scripting or adopting Physlets.

Conclusion

Physlets are very useful tools for relatively easy creation and adoption of simple and lightweight simulations for physics education. It is also possible to easily publish Physlet problems in Internet because Java applet and JavaScript technologies are used. If modifications to existing Physlet problems are required, it is relatively very easy to make them – it is necessary just to change parameters of embedded Physlets and JavaScript code.

Considering the fact that Physlets are free for non commercial use and all the benefits they provide, they can be of great value for physics educators in schools and universities.

References

Multimedia Educational Resource for Learning and Online Teaching "Merlot". [last accessed 27.04.2007]. Available online: http://www.merlot.org/merlot/index.htm

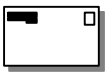
Physics Education Technology at the University of Colorado "PhET". [last accessed 27.04.2007]. Available online: http://phet.colorado.edu/web-pages/index.html

Christian W., Belloni M. (2001). Physlets: Teaching Physics with Interactive Curricular Material. Prentice Hall

Wolfgang Christian. Physlets® Home Page [last accessed 27.04.2007]. Available online: http://webphysics.davidson.edu/Applets/Applets.html

Belloni M., Christian W. (2003). Physlets® for Quantum Mechanics [last accessed 27.04.2007]. Available online: http://webphysics.davidson.edu/cise_qm/

Physlet Documentation [last accessed 27.04.2007]. Available online: http://webphysics.davidson.edu/Applets/physlet_documentation/documentation.html



Arnis Voitkans
Head of E-learning group, University of Latvia,
Raina boulevard 19-027, LV-1586, Riga, Latvia
E-mail: arnis.voitkans@lu.lv