STEAL THIS APPLET!

Douglas Ensley
Shippensburg University
Shippensburg, PA 17257
deensley@ship.edu

Barbara Kaskosz
University of Rhode Island
Kingston, RI 02881
bkaskosz@math.uri.edu

"Steal This Applet!" is a collection of applets on the website www.flashandmath.com with the focus of "instructor customization without programming." The collection is a somewhat surprising result of the grant project, "Tools and Training for Developers of Mathematics and Science Teaching Materials in Flash" (NSF DUE-0535327) for which the authors are co-Principal Investigators. The NSF project has a primary focus of teaching ActionScript programming to mathematics and science instructors and curriculum developers in order to establish and nurture a cadre of Flash developers within the educational community. Since the "Steal This Applet" collection specifically allows users to customize applet functionality *without* programming, it represents, in effect, an orthogonal path for future Flash development.

While the site www.flashandmath.com/stealthisapplet has over a dozen examples, this paper will focus on the philosophy and design of one particular example, and leave the rest for online discovery by the interested reader. Before presenting the example, we need to say a few words about XML (Extensible Markup Language), the format we use for passing data to our applets.

According to the World Wide Web Consortium (W3C), "Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere." There has been a great deal of activity in the development of specifications for XML for virtually every type of software application, web-based or otherwise, and this can be intimidating. The good news is that in order to customize an applet from the "Steal This Applet" collection, one need only understand the small number of tags and attributes defined for that particular applet, and these are always well documented. Anyone who can read html code can certainly understand the structure of our simple XML files.

We will use the Riemann Sums example from our site as our case study. The instructor who would like to steal this applet sees the interface shown in Figure 1 and makes decisions in the authoring interface, to include the following:

- Which approximation method(s) should be available to students?
- What should be the range and starting value for the number of subintervals?
- What examples should be initially available to the students?
- What color scheme looks best on the display technology at hand?
- How much should the student be able to edit?

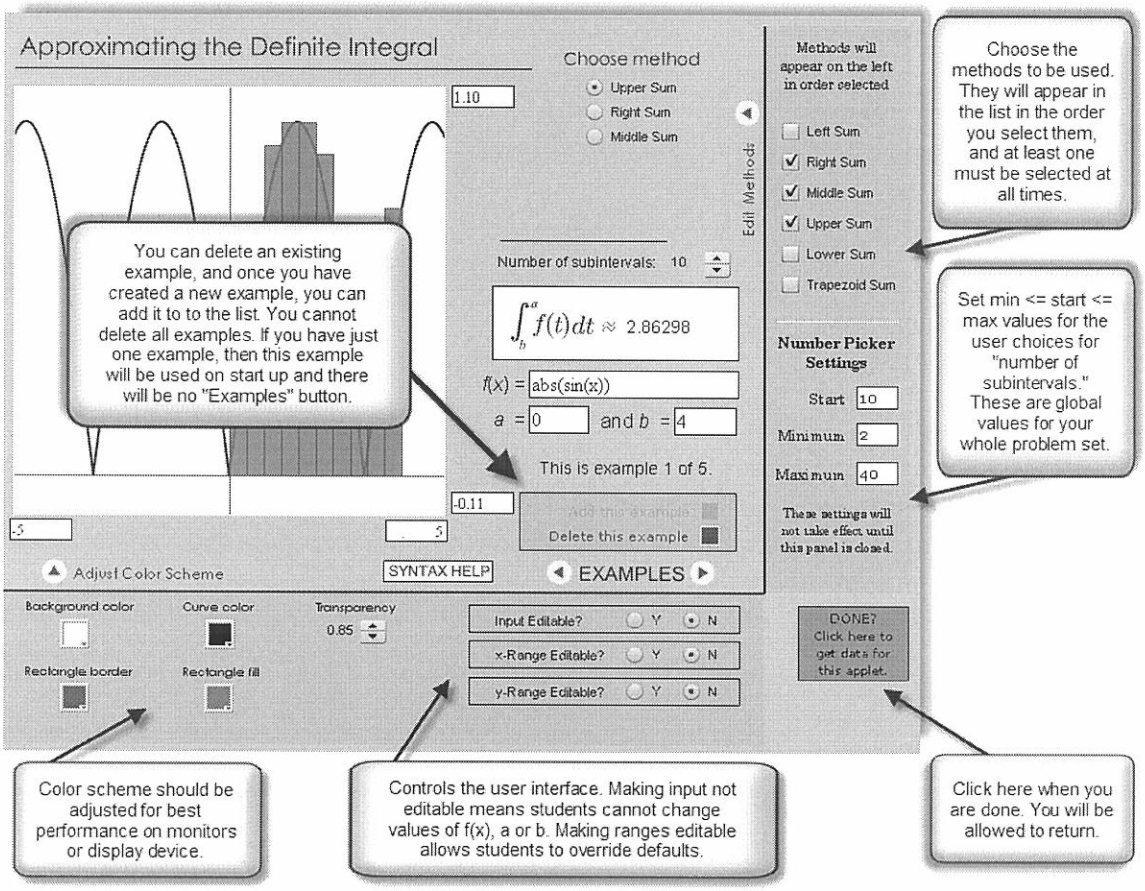

Figure 1. Customizing the Applet

The choices shown in Figure 1 lead to the XML code shown in Figure 2 when the user presses the "DONE?" button. The user then saves the XML code to a text file called riemannData.xml that must reside in the same directory as the applet file.

Looking closely at the XML file, shown in Figure 2, we see that it is not so mysterious after all. Once we understand the structure of the particular tags and attributes, we could customize the applet functionality by directly editing this file with any text editor. In fact, many of the applets on the "Steal This Applet" collection use directly edited XML files instead of custom instructor tools. In each of these cases, complete documentation of the tags and attributes is given on the site.

```
<xml>
<colorscheme bground='16777164' curve='0'
rectborder='10040115' rectfill='16724787'
transparency='0.7' />
<modify inputedit='true' xedit='true' yedit='true' />
<numsubintervals min='1' max='80' startval='10' />
<methods val='0,1,5,2' />

<problem expression='2*x + 1'>
<xrange min='-1' max='4' />
<yrange min='-0.77' max='7.70' />
<intrange min='1' max='3' />
</problem>

<problem expression='sqrt(9 - x^2)'>
<xrange min='-4' max='4' />
<yrange min='-2.00' max='6.00' />
<intrange min='0' max='3' />
</problem>

</xml>
```

Figure 2. The XML data for our choices

Figure 3 shows the applet resulting from placing the XML file from Figure 2 and the applet file in the same directory. (The instructor actually directs students to an html file that contains the applet file as an object.) Note that the applet in Figure 3 displays only the appearance, examples and functionality that we, as the instructor, have chosen.

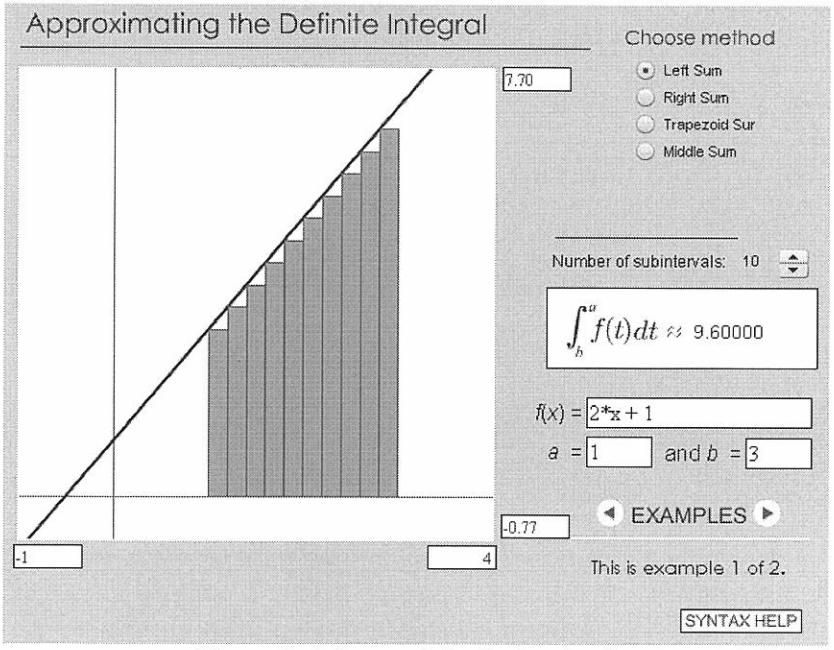

Figure 3. A Successfully Stolen Applet

One of the most exciting features of the Steal This Applet project is the potential for customization of applet functionality in order to have broadly sharable assets. The screen

shot in Figure 4 shows a clever applet designed by Barbara Margolius of Cleveland State University that presents a "reform calculus" problem in the form of a matching game. Not only are the displayed graphs specified by expressions in the companion XML file, but the instructions and the labels are also user-specified so the same applet can be used for entirely different purposes in other courses, as shown in Figure 5.

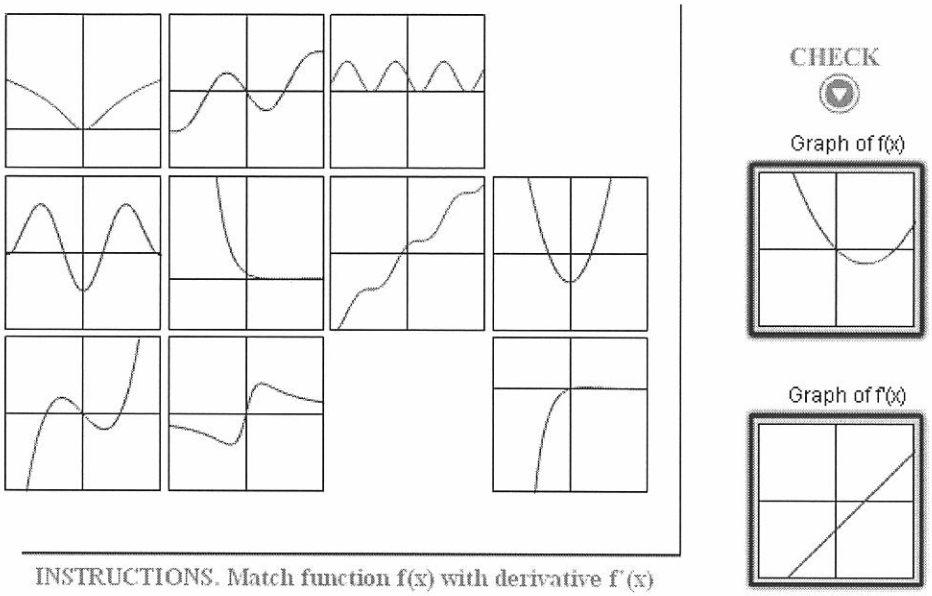

Figure 4. Customizing Functionality: A Calculus Applet

Figure 5 shows the same applet after changes to the XML file have been made to customize the functionality to a precalculus setting with no additional programming.

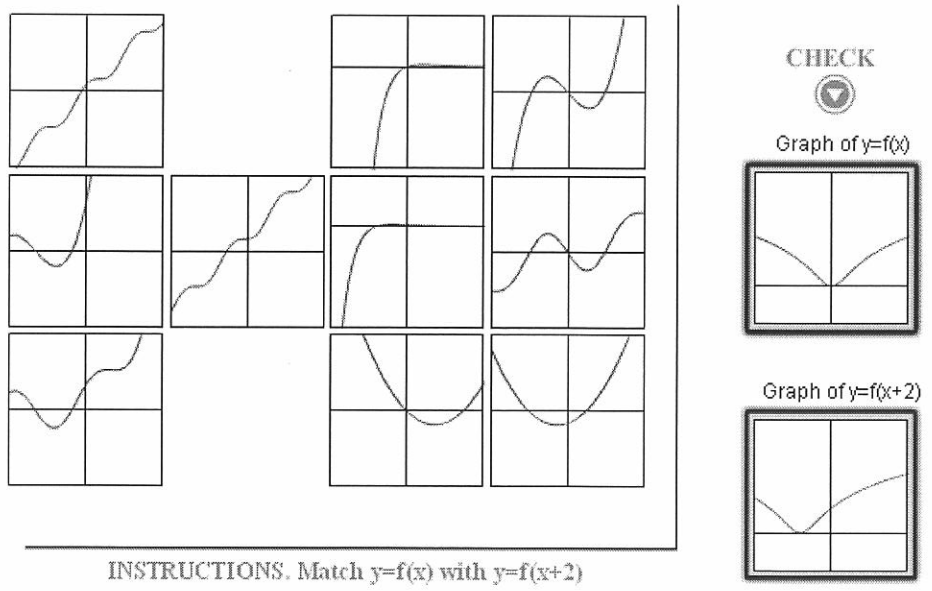

Figure 5. Customizing Functionality: A Precalculus Applet

This small collection of sharable applets represents a proof of concept for additional development of resources in Flash, Java, or other interactive, web-based formats. The next steps are to settle on some standards for the XML data and to develop a pool of common applets for high traffic courses such as precalculus and calculus. Future development may involve interaction with server-side scripts and databases, addressing SCORM compliance for integration with course management systems, or other issues that will advance the pedagogical value of these tools.

Acknowledgements. The idea for the collection has roots in Frank Wattenberg's work with "Lite Applets" and later with the OSSLET (Open Source, Sharable Mathlets) collection in the Mathematical Sciences Digital Library (MathDL) Classroom Resources. We fully developed these ideas for the first time during the Workshop on Better Practices for Math on the Web in July 2007 at the Mathematical Association of America Headquarters. The Matching Game applet is a modification of the work of Barbara Margolius, who in turn used custom classes developed during the 2007 MAA PREP workshop on Flash programming held at the University of Rhode Island.

Resources Cited

The "Steal This Applet" Collection, http://www.flashandmath.com/stealthisapplet, accessed May 1, 2008.

The Flash&Math ActionScript 3 Tutorials, http://www.flashandmath.com/, accessed May 1, 2008.

The Matching Game by Barbara Margolius, from the Calculus Resources Collection at http://www.csuohio.edu/success_in_math/calculusresources.html, accessed May 7, 2008.

Flash at the Beach: Creating Mathlets in Adobe Flash, 2007 MAA PREP Workshop Resource page at http://webspace.ship.edu/deensley/prep2007/, accessed May 7, 2008.

Mathematical Sciences Digital Library Open Source Sharable Applets (OSSLETs), http://mathdl.maa.org/mathDL/3/?pa=content&sa=viewDocument&nodeId=1032&bodyId=1224, accessed May 1, 2008.

Better Practices for Math on the Web, http://mathonweb.org, accessed May 1, 2008.