



You are not logged in.

[Log in](#) | [Register](#)

Volume 3

2003

Published by the

MAA

[Copyright Notice](#)

[Privacy Policy](#)

[Editorial Board](#)

[About JOMA](#)

[Mathlets](#)

[Developers' Area](#)

[Archive of past issues](#)



Mathematical
Sciences Digital
Library

Journal of Online
Mathematics and its
Applications

Digital Classroom
Resources

[About MathDL](#)

Quick Interactive Web Pages with Java Sketchpad

[Michael E. Mays \(e-mail\)](#)

[West Virginia University](#)

Contents

1. [Introduction: Geometer's Sketchpad and Java Sketchpad](#)
2. [A Simple Example: .gsp to .html](#)
3. [A More Complicated Example: Displaying Units](#)
4. [Animation and Interactive Applets](#)
5. [Conclusion](#)

Resource Notes

- a. If your browser has an old version of the Java plug-in, you may need to download and install a current version in order to run some of the applets in this article. All of the applets have been tested and work with Version 1.4.2 of the Java Runtime Environment, Standard Edition, in Internet Explorer, Netscape, and Mozilla. You can download the current version (free) from [Sun Microsystems](#).
- b. You do not need to own *The Geometer's Sketchpad*® to read this article. However, to view source code for *Sketchpad* examples or to create your own examples, you do need this product. See the first paragraph of the Introduction for information about acquiring it.

1. Introduction: Geometer's Sketchpad and Java Sketchpad

The Geometer's Sketchpad® is a Dynamic Geometry® exploration environment available for Macintosh® and PC computers from [Key Curriculum Press](#). *JavaSketchpad* is software that facilitates development and distribution of mathlets by publishing sketches from *The Geometer's Sketchpad*® on the Internet. The *JavaSketchpad* Developer Release 4 is integrated into *Sketchpad* version 4, or it can be downloaded independently from [Key Curriculum Press](#). Besides the downloadable software, the Key Curriculum site has detailed information about creating *JavaSketchpad* applets and a gallery of examples to explore.

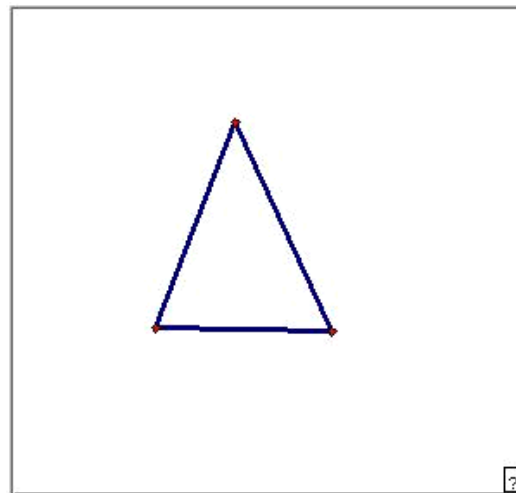
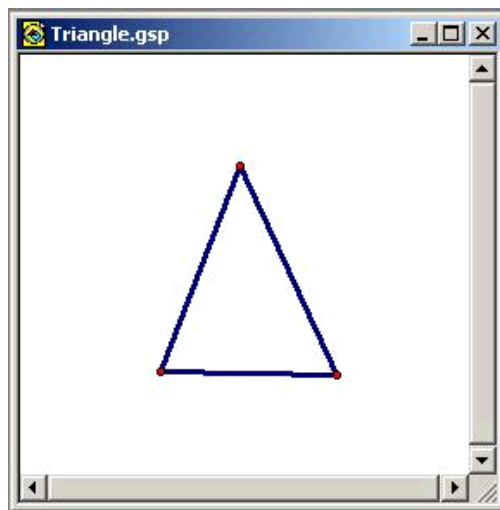
Although *JavaSketchpad* requires a Java-enabled browser to view the sketches, you do not have to program in Java to create them. Instead, you can build a sketch in *The Geometer's Sketchpad* Version 4 and save it in HTML format. The sketch as saved in HTML looks like an old fashioned (pre-Version 4) gsp script, with calls to class files provided in the /jsp folder via parameter passing.

Everybody draws a triangle as their first *Sketchpad* sketch. If you have *Sketchpad* installed on your computer, you can click [here](#) to open a gsp sketch of a generic triangle. Whether you have *Sketchpad* installed or not, you can click [here](#) to open the *JavaSketchpad* version. Inspect the HTML code of Triangle.htm, and you will see the meat of the construction in the lines

```
<PARAM NAME=Construction VALUE="
{1} Point(127,64);
{2} Point(81,183);
{3} Segment(2,1)[color(0,0,128),thick];
```

```
{4} Point(183,185);  
{5} Segment(4,2)[color(0,0,128),thick];  
{6} Segment(1,4)[color(0,0,128),thick];  
">
```

which specify the initial placement of three points and three lines. The following figures show how similar the results are.



JavaSketchpad is a stand-alone product that can be used independently of *The Geometer's Sketchpad*. Key Curriculum Press grants the right to freely use *JavaSketchpad* applets in your own Web pages, provided you do so for non-commercial use only. Other permissions or rights to *JavaSketchpad* require a written agreement letter from Key Curriculum Press; contact Key Curriculum Press for more info.

[top](#)[next](#)[MAA ONLINE](#)[MATH FORUM](#)[NSF](#)



You are not logged in.

[Log in](#) | [Register](#)

Volume 3

2003

Published by the

MAA

[Copyright Notice](#)

[Privacy Policy](#)

[Editorial Board](#)

[About JOMA](#)

[Mathlets](#)

[Developers' Area](#)

[Archive of past issues](#)



Mathematical
Sciences Digital
Library

Journal of Online
Mathematics and its
Applications

Digital Classroom
Resources

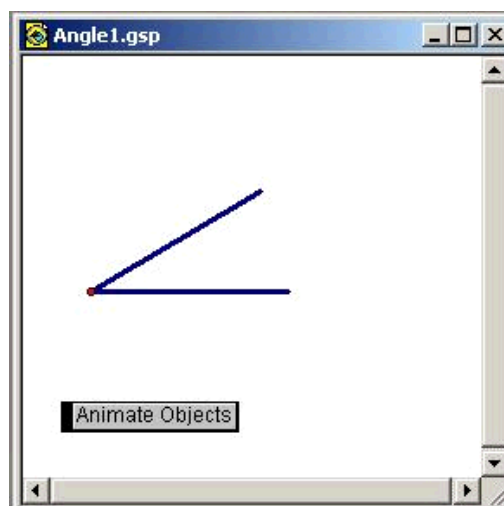
[About MathDL](#)

[top](#)
[previous](#)
[next](#)

Quick Interactive Web Pages with Java Sketchpad

2. A Simple Example: .gsp to .html

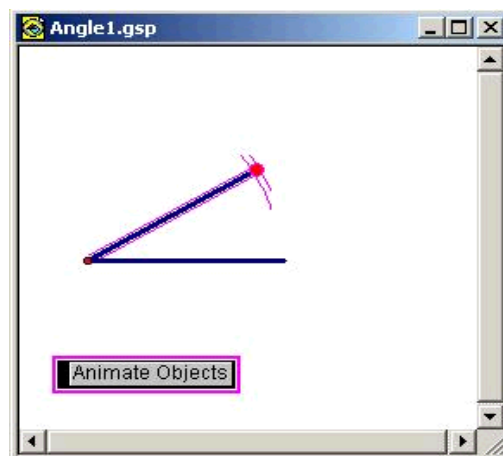
Suppose you want to provide your students an animated illustration of an acute angle. In *Sketchpad* you might use a horizontal line segment as one side of the angle, the arc command to create an arc in the first quadrant, a point on the arc to determine the terminal side, and an animate button to allow the point to move on the arc. This approach is illustrated in the following *Sketchpad* sketch -- to download the gsp file, click on the sketch.

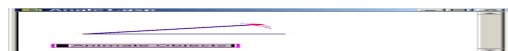


This works in *Sketchpad*, but you can not save it from *Sketchpad* as a working HTML file. The problem reported is

"Some objects in this sketch are not supported by this version of JavaSketchpad, and will not appear in your applet. (These objects have been selected.)"

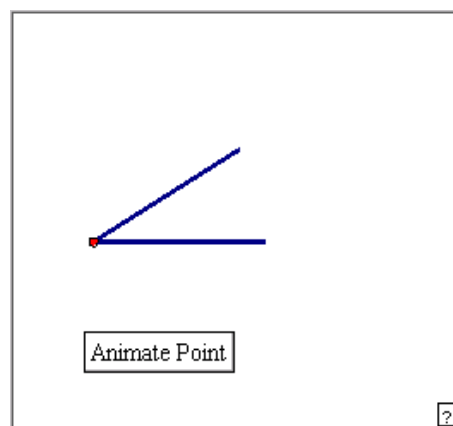
The [applet](#) saved is not functional, but the HTML page displayed in *Sketchpad* looks like this:





The problem is that the current version of *JavaSketchpad* does not support the arc construction, which means the point cannot be selected on the arc, and so the Animate Objects button is broken as well. As you will see when you click the [applet](#) link, only the unbroken objects survive.

A work-around could be to draw a line segment in the first quadrant and animate the angle using a point on the line segment to locate the terminal side. [This version](#) of the Sketchpad file can be downloaded and run (if Sketchpad is on your computer) -- here is the working applet saved from it:



In contrast to our example on the first page, the applet here is embedded in this page, rather than opened in a new page. You may choose how to make the most effective use of the animated illustration that *JavaSketchpad* provides.

[top](#)[previous](#)[next](#)[MAA ONLINE](#)[MATH FORUM](#)[NSF](#)



top

previous

next

You are not logged in.

[Log in](#) | [Register](#)

Volume 3

2003

Published by the
MAA[Copyright Notice](#)[Privacy Policy](#)[Editorial Board](#)[About JOMA](#)[Mathlets](#)[Developers' Area](#)[Archive of past issues](#)


Mathematical
Sciences Digital
Library

Journal of Online
Mathematics and its
Applications

Digital Classroom
Resources

[About MathDL](#)

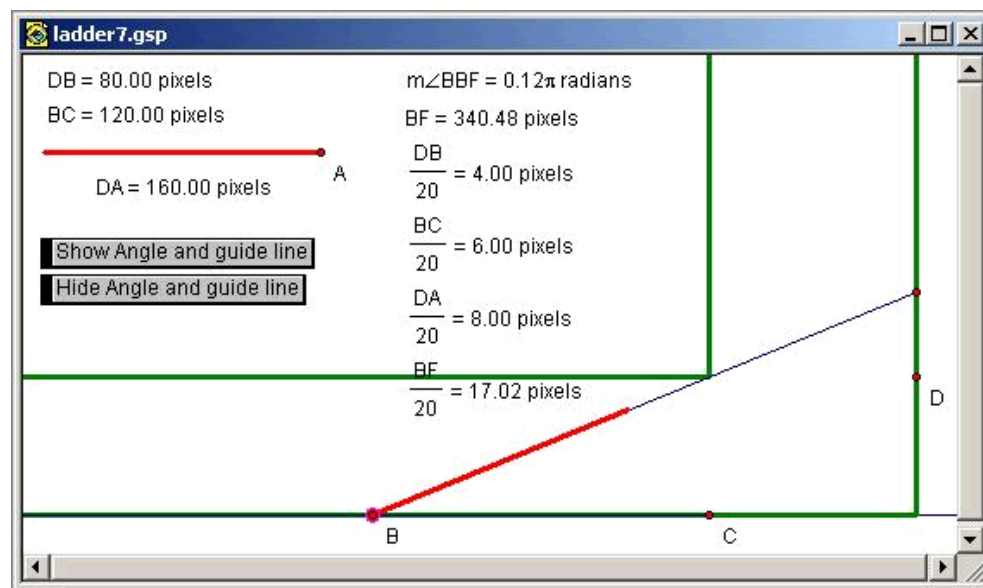
Quick Interactive Web Pages with Java Sketchpad

3. A More Complicated Example: Displaying Units

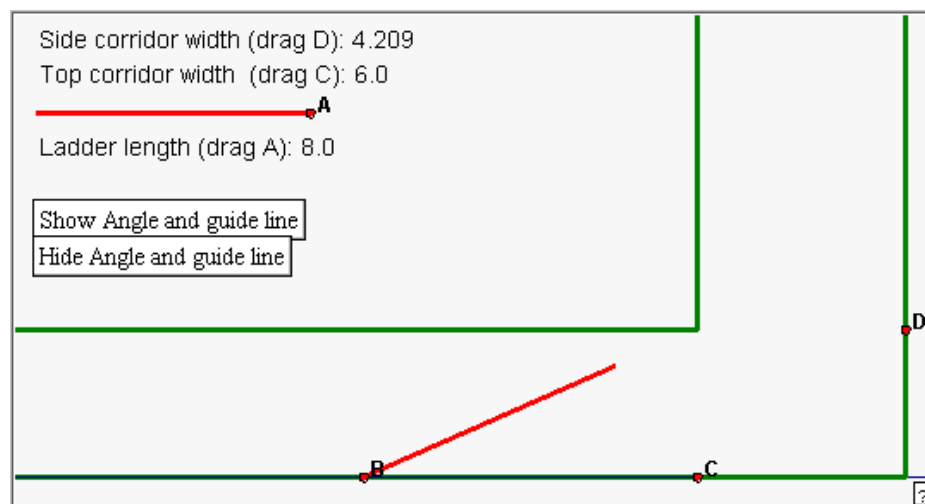
JavaSketchpad has a different feel from *The Geometer's Sketchpad*®: It has no menus, dialog windows, or toolboxes. This means you (as designer) have to think ahead to decide what interactions are appropriate and should be included. Also, there are limitations based on the Java virtual machine that runs the applet in the browser.

Perhaps the most limiting is that, with no assumptions about screen geometry built in, the only units available for measurements are pixels, and there is no provision in *JavaSketchpad* for directly adjusting the number of decimal places of the measurements reported. *JavaSketchpad* does support calculations, however, so it is possible to convert from pixels to a more appropriate scale "behind the scene," hide the true measurement, and display the answer rounded appropriately.

For example, a *Sketchpad* model of the "ladder problem" shows how a ladder is to be taken around a corner between two intersecting corridors. If you have *Sketchpad* installed on your computer, you can click on the picture to download the gsp file.



The interactive *JavaSketchpad* applet displays appropriate units:



I made this change by directly editing the HTML code produced by *The Geometer's Sketchpad*, hiding the measurements in pixels and calculating the other measurements so the scale seems appropriate. For example, line {38} in the following code scales the hidden measurement of line {10} by an arbitrary 20 units, and line {42} converts the angle measure to degrees.

```

<PARAM NAME=Construction VALUE=""
{1} Point (16,266) [hidden];
{2} Point (517,266) [label('B'),hidden];
{3} Line (2,1) [color(0,0,128)];
{4} Perpendicular (3,2) [color(0,0,128),hidden];
{5} Point on object (4,0.0) [color(255,0,0),hidden];
{6} Ray (5,2) [color(0,128,0),thick];
{7} Ray (1,2) [color(0,128,0),thick];
{8} Point on
object (7,0.239521) [label('C'),color(255,0,0)];
{9} Point on
object (6,0.336064) [label('D'),color(255,0,0)];
{10} Distance (2,9,14,19,'DB = ') [color(0,0,0),hidden];
{11} Distance (8,2,14,39,'BC = ') [color(0,0,0),hidden];
{12} Parallel (6,8) [color(0,0,128),hidden];
{13} Parallel (7,9) [color(0,0,128),hidden];
{14} Intersect (12,13) [label('C'),color(255,0,0),hidden];
{15} Point on object (12,3.0) [color(255,0,0),hidden];
{16} Ray (15,14) [color(0,128,0),thick];
{17} Point on object (13,3.0) [color(255,0,0),hidden];
{18} Ray (17,14) [color(0,128,0),thick];
{19} Point on
object (3,0.0898204) [color(255,0,0),hidden];
{20} Ray (19,8) [color(0,0,128)];
{21} Point on
object (20,0.580357) [label('B'),color(255,0,0)];
{22} Line (14,21) [color(0,0,128),hidden];
{23} Intersect (6,22) [color(255,0,0),hidden];
{24} Segment (23,21) [color(0,0,128),hidden];
{25} Point (12,56) [label('D'),hidden];
{26} Parallel (3,25) [color(0,0,128),hidden];
{27} Point on
object (26,0.819361) [label('A'),color(255,0,0)];
{28} Segment (27,25) [color(255,0,0),thick];
{29} Line (21,25) [color(0,0,0),hidden];
{30} Parallel (29,27) [color(0,0,0),hidden];
{31} Intersect (3,30) [color(255,0,0),hidden];
{32} Circle (21,31) [color(0,128,0),hidden];
{33} Intersect2 (24,32) [color(255,0,0),hidden];

```

```
{34} Segment(33,21)[color(255,0,0),thick];
{35} Distance(27,25,42,81,'DA = ')[color(0,0,0),hidden];
{36} Distance(23,21,221,41,'BF =
')[color(0,0,0),hidden];
{37} Angle(2,21,23,210,119,'Angle B (radians):
')[color(0,0,0),hidden];
{38} Calculate(14,19,'Side corridor width (drag D):
','A20 /')(10)[color(0,0,0)];
{39} Calculate(14,39,'Top corridor width (drag C):
','A20 /')(11)[color(0,0,0)];
{40} Calculate(14,81,'Ladder length (drag A): ','A20
/')(35)[color(0,0,0)];
{41} Calculate(210,159,'Guide line length: ','A20
/')(36)[color(0,0,0),hidden];
{42} Calculate(210,139,'Angle B (degrees): ','A57.29578
*')(37)[color(0,0,0),hidden];
{42} ShowButton(10,106,'Show Angle and guide
line')(37,41,42,24,23)[color(0,0,0)];
{43} HideButton(10,127,'Hide Angle and guide
line')(37,41,42,24,23)[color(0,0,0)];
">
```

[top](#)[previous](#)[next](#)[MAA ONLINE](#)[MATH FORUM](#)[NSF](#)


[BROWSE](#)
[SEARCH](#)
[DISCUSSIONS](#)
[GET INVOLVED](#)
[AUTHOR SERVICES](#)
[MathDL HOME](#)
[JOMA HOME](#)
[HELP](#)

You are not logged in.

[Log in](#) | [Register](#)

Volume 3

2003

Published by the

MAA

[Copyright Notice](#)

[Privacy Policy](#)

[Editorial Board](#)

[About JOMA](#)

[Mathlets](#)

[Developers' Area](#)

[Archive of past issues](#)

[MathDL](#)

Mathematical
Sciences Digital
Library

Journal of Online
Mathematics and its
Applications

Digital Classroom
Resources

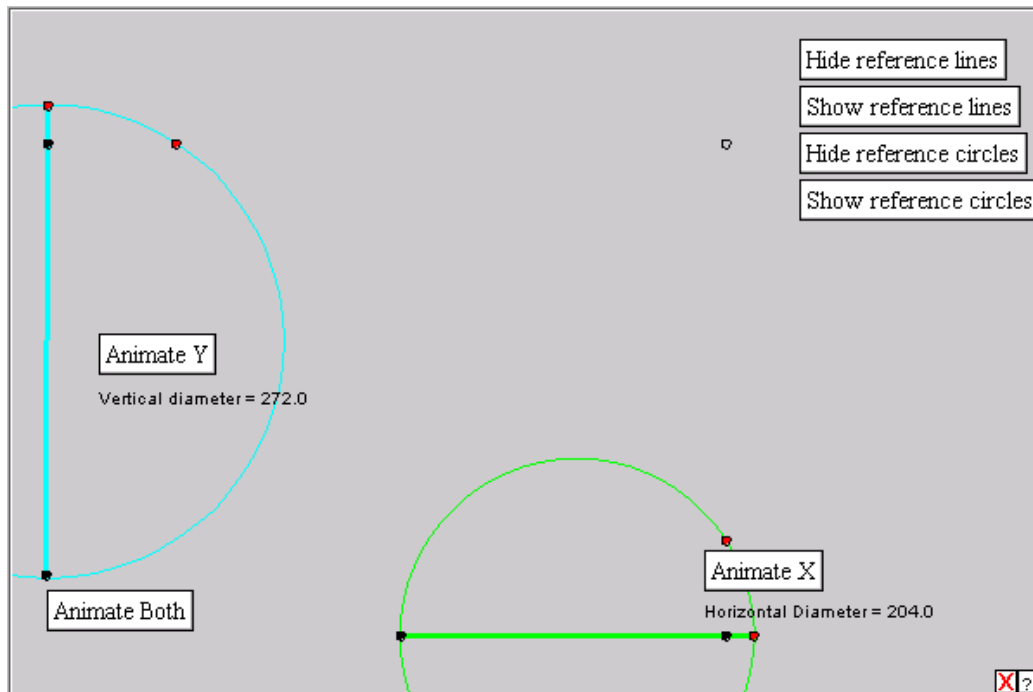
[About MathDL](#)

[top](#)
[previous](#)
[next](#)

Quick Interactive Web Pages with Java Sketchpad

4. Animation and Interactive Applets

Here is an example of a Lissajous curve tracer implemented in *JavaSketchpad*. To generate a Lissajous curve we want the two coordinates parametrized by sinusoidal curves of possibly different periods. It is easy to translate circular motion into sinusoidal curves by dropping a perpendicular. This version lets you keep track of the coordinates separately or together, adjust the periods by changing the diameters of the circles, and include guide lines to illustrate the horizontal and vertical components of the traced points.



The [Sketchpad file](#) can be downloaded if you would like to inspect or modify it. I find the trace prettier to watch on the applet version than on the *Sketchpad* version -- the shapes of the trace points provide it with a three dimensional quality.

[top](#)
[previous](#)
[next](#)
[MAA ONLINE](#)
[MATH FORUM](#)
[NSF](#)


[BROWSE](#)
[SEARCH](#)
[DISCUSSIONS](#)
[GET INVOLVED](#)
[AUTHOR SERVICES](#)
[MathDL HOME](#)
[JOMA HOME](#)
[HELP](#)

You are not logged in.

[Log in](#) | [Register](#)

Volume 3

2003

Published by the

MAA

[Copyright Notice](#)

[Privacy Policy](#)

[Editorial Board](#)

[About JOMA](#)

[Mathlets](#)

[Developers' Area](#)

[Archive of past issues](#)

[MathDL](#)

Mathematical
Sciences Digital
Library

Journal of Online
Mathematics and its
Applications

Digital Classroom
Resources

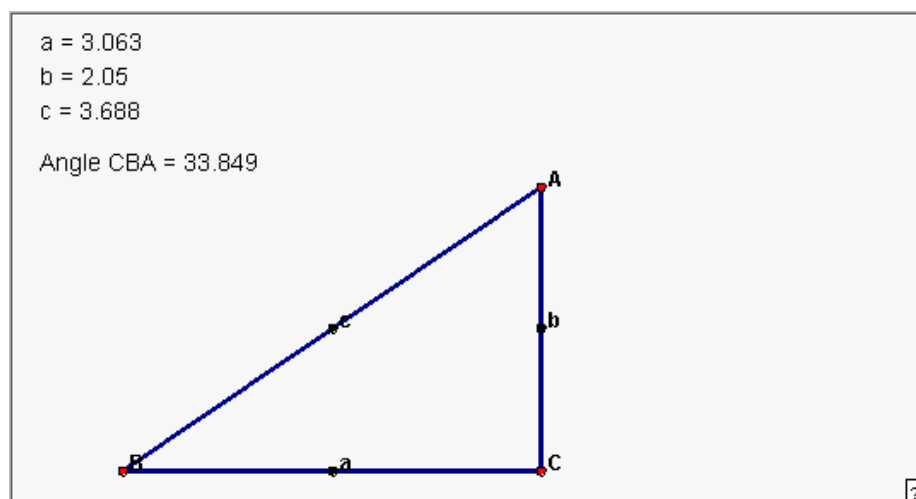
[About MathDL](#)

[top](#)
[previous](#)

Quick Interactive Web Pages with Java Sketchpad

5. Conclusion

The story goes that when tools such as Visual Basic and Delphi became available, software developers would use them to develop prototypes for products they were considering, and then when crunch time came at the end of the project timeline they would just ship the prototype. I started using *JavaSketchpad* to write online laboratories for a trigonometry course with the idea of writing Java applets of comparable functionality from scratch myself, but for the most part I have stuck with the original *JavaSketchpad* prototypes. For example, here is the *JavaSketchpad* version of an activity to manipulate the vertices of a right triangle, with the goal of calculating values of trig functions directly:



This applet was generated in *Sketchpad* by saving the sketch as an HTML file. (Here is the [gsp file](#) to generate the sketch.) The following applet shows the same idea implemented from scratch:



The Java applet was developed in the Integrated Development Environment Visual Cafe 4.5 ([zipped Visual Cafe project files](#)). Visual Cafe has been developed and distributed by Symantec and then by WebGain, but its current status is unclear.

So which applet is better? There is a smoothness to the *JavaSketchpad* applet that the IDE applet lacks. IDE has a distracting flicker when the points are moved, and needs to be refreshed when the applet moves offscreen or is covered up. On the other hand, I could label the line segments in IDE without displaying midpoints as I had to in *JSP*, and I easily implemented a units-changing feature. All the cosmetic objections to either version are probably fixable with a little more understanding of the platform, but there are two differences that are not:

1. It took an hour or so to write the *JavaSketchpad* applet, and a couple of days to write the IDE version. I find myself getting faster in Java with practice, but I can't see myself ever getting *that* fast.
2. If I wanted to publish the applet for profit, say in a textbook ancillary, the Java license from [Sun](#) makes the IDE version mine, but the *JavaSketchpad* version is by default limited to non-profit use.

Students using the lab don't seem to care which version they use. If you know how to use *The Geometer's Sketchpad*®, *JavaSketchpad* is certainly worth exploring as a way to port sketches to applets. As additional *GSP* features are incorporated into *JavaSketchpad*, it will become even more useful.

[top](#) [previous](#)