USING GEOGEBRA TO CREATE APPLETS FOR VISUALIZATION AND EXPLORATION

Mike May, S.J.
Department of Mathematics and Computer Science, Saint Louis University
220 N Grand Blvd, St. Louis, MO, 63103
maymk@slu.edu

GeoGebra is a free, open source, cross platform program that can easily be used to construct demonstrations, visualizations, and exploration tools for courses in mathematics from algebra through calculus. The software allows the user to export demonstrations as applets embedded in a dynamic web page, to be used either over the web or through a java enabled browser locally. This paper explores using GeoGebra to create applets, either to be used for demonstrations for the teacher or to be used as exploration tools by the student. Throughout the paper will refer to the minicourse handouts [1] used at ICTCM XX.

First we start with examples to illustrate some of the features that can easily be incorporated into applets with GeoGebra. Then we discuss the process of saving a GeoGebra construction to a dynamic web page or applet. We then look at the use of javascripts to enhance the web pages. Finally we provide collection of resources.

Section 1 – A fast tour of examples - Screenshots of the example applets we discuss in this section are in the minicourse handouts [1] and the applets themselves are available at: http://www.slu.edu/classes/maymk/GeoGebra [2] with other examples available at the GeoGebra wiki [3].

The first example ([1], p. 4) is a simple tool for exploring the behavior of families of functions, where the function is defined with up to three parameters and the student is to study how changing the parameters changes the behavior of the graph. The applet itself has three sliders for dynamically viewing the given family of functions. The demonstration/exploration tool can be used to cover standard material in a number of courses. Both the graph and the formula for the function appear in the applet. Buttons in the upper left corner of the applet allow the user to select an object, zoom in or out, or to move the graph. Above the applet, the user is given enough text to explorer the mathematical point of the page. Below the applet are three lines of javascript generated controls. The first line lets the user enter a different function. The second line is a drop down menu that lets the user select from a number of preset examples. The third line allows the user to set the viewing window. The javascript code used for these control lines is available in the minicourse handouts ([1], pp. 19-20).

The second example ([1], p. 5) connects the slope of the tangent lines to a curve to the graph of the derivative. While the mathematical content has changed, the structure of the web page is very similar. The third example ([1], p. 6) looks at approximations to a function by Taylor polynomials. The noteworthy shift in this example is that the user can obtain significant text output through a javascript text area. Note that the output is structured to have data about the visualization embedded in descriptive text. The javascript code for the text box construction is available in the minicourse handouts ([1], p. 21).

The third example ([1], pp. 7-8) explores the construction of parabolas and hyperbolas. This example is noteworthy since the descriptive text changes based on the conditions of the construction, either commenting that we have a hyperbola, an ellipse, or a degenerate conic. This is done with a collection of text strings that are made visible (an advanced property) based on a conditional involving the length of the major axis and the distance between the foci. The example on the proof of the Pythagorian theorem ([1], p. 11) uses the same basic technique to walk students through 7 steps in a proof.

The dynamic web page on straight lines ([1], p.9) shows two applets on a single web page. This is done by copying the applet tag from one html page to another and then noting that that html page now needs two ggb files to work correctly.

The example showing that the three bisectors of the sides of a triangle ([1], p. 10) always meet in a single point is noteworthy for two reasons. It uses checkboxes to make objects appear and disappear in the demonstration. We will also walk through a simplified version of this construction later in the paper.

Section 2 – Functionality of GeoGebra – Pages 12-17 of the handouts [1] give an overview of some of the functionality of GeoGebra. The easiest interface to begin with is the collection of icon menus ([1], p. 12). They are fairly self explanatory, with for example an icon of a circle with a point on the circle and a point at the center being used to construct a circle by giving the center and a point on the circle. We also include a complete list of the icons in the menus with their names ([1], p.13).

While the icon menus give a fast start, there are times when it is useful to use the input line to give a text command. We often do this when we want to specify the name of the created object. We also do it with commands that don't have icons. A list of many relevant commands, organized by course material is given in the handout ([1], p. 15-17)

The default state of GeoGebra give an "Algebra window" in addition to the drawing window. (See [1], p. 14.) This lets you match drawn object up with algebraic formulas of the same objects. On can open a dialog box to change properties of any object, either visible properties like the font color and size, or the definition, or the name of the object.

Section 3 – Creating a demonstration – It is worthwhile walking through a construction that we will turn into an applet. We will construct a demonstration that shows that the three perpendicular bisectors of the sides of a triangle all meet in a single point. Start by opening GeoGebra on your own machine. Select the point icon and click three times in the drawing window to create points A, B, and C. From the line menu (third icon from the left), select the line segment icon (second icon down), then choose each pair of points to draw lines a, b, and c. Next select the line bisector icon (fourth icon over, third down) and select each line to construct the three bisectors, d, e, and f. From the point menu, select the intersect icon, then select two of the bisectors to create point D where they intersect. Choosing the select icon, you can now drag the vertices of the triangle around and see that the three bisectors are always concurrent. Selecting one of the bisectors, then properties from the edit menu, edit the properties of d, e, and f to make the lines heavier and give them a distinct color.

This will be enough to convince a student that the theorem is true. That will make the student ready to look at a proof. Select the point icon and put a point on one of the bisectors. Select the circle icon then select first the new point, then one of the vertices of the bisected side. The resulting circle will go through the other vertex for that side. Dragging the point on the bisector shows that the point of intersection is equidistant from all three vertices, setting up the argument that such a point must always exist.

Section 4 – Exporting to web pages - With GeoGebra, it is easy to convert a demonstration within GeoGebra into a dynamic webpage usable, either over the web or locally, from a java enabled browser. From the file menu, select "Export", then "Dynamic Worksheet as Webpage (html)." You will be given a dialog box with two panels, general and advanced. In the general panel you enter a title for the worksheet, and add text that will appear on the web page above and below the applet. The advanced dialog can be ignored for the time being. It gives the applet creator control over the options presented to the applet user. One can decide whether or not the user sees a GeoGebra toolbar or menubar, the size of the applet on the web page, whether or not the user can open a full GeoGebra window outside the web page, and where the web page will go for the jar archive.

After selecting export from this dialog you will be asked to name the file and select where on your hard drive it will be saved. For the first dynamic webpage, it is worthwhile to save the webpage in a new folder or directory. This makes it easier to keep track of the files that need to be kept together for the webpage to work. We named our file AppletCreation. This caused four files to be created. AppletCreation.ggb is a GeoGebra file that contains the data on the demonstration we constructed. AppletCreation.html is the dynamic webpage containing the applet. This is the page that you open with a browser. Geogebra.jar is the GeoGebra applet proper. While the html page can be modified to look for this file elsewhere, the default setup looks for this file in the same folder as the html file. Similarly, geogebra_properties.jar must be in the same folder as geogebra.jar. One folder can be used for a single pair of geogebra.jar and geogebra_properties.jar files and many pairs of html and ggb files. The created dynamic web page should launch automatically.

For the most part, you should not need to learn html to create web pages. For the minicourse, we used SeaMonkey, a free, open source, cross platform project derived from Mozilla that includes composer, a visual web page editor. This lets you edit the text on the web page and add links to worksheets without learning html. SeaMonkey can be found at http://www.seamonkey-project.org/. Nevertheless, it is worthwhile to look at the applet tag, which is the code of the html page that actually calls the applet. The true or false options are fairly self explanatory, but can be ignored usless you want to reset them. The applet name will be important in the next section when we discuss javascript additions. The codebase tells the browser where to look for the geogebra archive. The default of "./" indicaes that the jar file is in the same folder as the html page.

```
<applet name="ggbApplet" code="geogebra.GeoGebraApplet" codebase="./"
archive="geogebra.jar" width="900" height="591">
        <param name="filename" value="AppletCreation.ggb">
        <param name="framePossible" value="true">
        <param name="showResetIcon" value="true">
        <param name="enableRightClick" value="false">
        <param name="showMenuBar" value="false">
        <param name="showToolBar" value="true">
        <param name="showToolBarHelp" value="true">
        <param name="showAlgebraInput" value="false">
Sorry, the GeoGebra Applet could not be started. Please make sure that Java 1.4.2 (or
later) is installed and active in your browser (<a href="http://java.sun.com/getjava">Click
here to install Java now</a>)
</applet>
```

Section 5 Back to javascript - Several of the examples in the minicourse handout contains lines of javascript controls to either allow the user to pass information like viewing window or selection of a prepared example to the applet, or to retrieve information, like a Taylor polynomial, from the applet in a form that can be copied and pasted into a text document. It is worthwhile to explain just enough of the javascript code in the handout pages for you to cut paste and modify. We will refer to pages 19 and 20 of the handouts.

To use javascript to enhance Geogebra, you need to add two blocks of code, a form which turns into a set of buttons, text blocks, check boxes, and drop down menus, and a script that includes the functions that tell your computer what to do when you interact with the form. Since these two blocks, and the applet tag can appear in various places in the webpage, they need to be named. We consistently name the applet ggbApplet (the default name) though we modify that name when we have two applet tags in a single webpage. In the function scripts the applet is then document.ggbApplet, since the reference is to the same webpage.

Our most common control panel is a form for setting the view window on the applet. In the sample code it is named ViewForm. This form has four text fields, named LowXField, HighXField, LowYField, and HighYField, to collect the obvious four values. The Form also contains a button to collect those four values and send them to the applet. Entering a value in one of the fields, or clicking the button calls a function named setView. The script for the function setView collects four values from document.ViewForm and uses them to construct a setCoordinateSystem command to document.ggbApplet.

The second example in the handout pages shows how to pass information in both directions. Depending on the button pushed, you can either send the definition of a function to the applet or retrieve the definition of the function from the applet.

The third example given uses a drop down menu to let the user select from a list of pre-set examples. It is noteworthy that choosing an example sends more than one command to the applet, changing both viewing window and the key function of the applet.

More information on the commands that an applet will accept from the webpage via javascript can be found at:
http://www.geogebra.org/source/program/applet/geogebra_applet_javascript.html

References:

[1] http://www.slu.edu/classes/maymk/GeoGebra/pdfs/ICTCMHandouts.pdf - Handouts for the minicourse on GeoGebra at ICTCM XX
[2] http://www.slu.edu/classes/maymk/GeoGebra - Web location of the collection of applets discussed in this paper.
[3] http://www.geogebra.org/en/wiki/index.php/Main_Page - The largest repository of other GeoGebra material.