GeoGebra goes Mobile

Gábor Ancsin
gabor@geogebra.org
Pápay Endre Voc. School
Makó
6900
Hungary

Markus Hohenwarter
markus@geogebra.org
Dept. of Math. Education
J. Kepler University, Linz
4040
Austria

Zoltán Kovács
zoltan@geogebra.org
Dept. of Analysis
University of Szeged
6701
Hungary

Abstract

The introduction of smartphones with broadband Internet access allows students to access educational materials from almost everywhere. While the open source software GeoGebra is widely used on desktop and laptop computers, it is currently not available for the majority of mobile touchscreen devices like Apple's iPhone/iPad or Google's Android devices. In this paper we describe the project GeoGebraMobile which aims to overcome this limitation by making GeoGebra applets accessible to students on a wide range of mobile devices.

1 Introduction

With the open source educational mathematics software GeoGebra [1] it is possible to create web pages with embedded interactive applets that can be used in a web browser. One of the main goals of GeoGebra is to make interactive mathematics materials available to students everywhere. This focus on availability concerns the following aspects:

- Costs: software and educational materials should be available free of charge for students and teachers from the project's website.
- Languages: the software's user interface should be available in the local language of students and teachers. GeoGebra is currently available in 50 languages.
- Platforms: the software should run on a large variety of platforms and operating systems to allow students to use the interactive materials on any computer or mobile device.

This paper focuses on the platforms supported by GeoGebra. We will first explain the current situation and then present the GeoGebraMobile project which aims to allow the use of interactive GeoGebra materials on mobile phones and tablet computers like Apple's iPhone/iPod/iPad, Google's Android devices, and others.

2 Differences between Desktop Computers and Mobile Devices

Currently, the software GeoGebra and its interactive online materials work on computers that have the Java Standard Edition (JavaSE) with its web browser plugin installed [2]. This technology is available free of charge for all major desktop operating systems including Microsoft Windows, MacOS X, and major Linux distributions (using the OpenJDK [3] variant). Today, Java is (pre-)installed on many desktop computers out of the box, so students can use the GeoGebra software and its interactive online materials immediately. Thus, the use of GeoGebra is possible on all desktop and laptop computers but may require the installation of Java.

The situation for mobile devices is somewhat different. Older mobile phones had support for Java Micro Edition (JavaME) which was used for games and mobile applications like email [4]. Unfortunately, standard JavaSE software like GeoGebra cannot be used easily on JavaME phones. We had expected that JavaSE would eventually become available with the introduction of more powerful newer mobile phones. However, this did not happen. Instead, the incompatibility among mobile phone platforms has increased. Google has developed its own Android [5] platform for mobile devices just like Apple did for its iPhone. Both platforms neither support JavaME nor JavaSE applications.

Today, we think that JavaSE will not be widely supported by mobile phones and tablets anytime soon (if at all). However, touchscreen phones and tablet computers are ideal devices to interact with dynamic mathematics software like GeoGebra as they allow dragging of e.g. points and sliders in a very natural way with your fingers.

As sales of mobile touchscreen phones and tablet computers are increasing quickly [6], the GeoGebra developer group was facing an important question: How can we get GeoGebra applets to run on these new mobile devices?

3 GeoGebraMobile

The project GeoGebraMobile [7] was started in 2009 with the goal to find a way to allow the use of existing interactive GeoGebra materials [8] on mobile phones and tablet computers. There are two possible solutions to achieve this: use an existing system that can display GeoGebra files or port GeoGebra to a platform that is supported by these devices.

3.1 Using an Existing System: JSXGraph

JSXGraph [9] is an open source library that allows dynamic mathematics applets to be embedded into web pages. The remarkable thing about this software is the fact that it works in modern web browsers without the need of a plugin software like JavaSE. The only requirement is that the web browser needs to support vector graphics in the form of SVG [10] (like Mozilla Firefox, Google Chrome, Opera, Safari) or VML [11] (for Microsoft Internet Explorer). Today, mobile phones typically come with web browsers that already support these features.

Both GeoGebra and JSXGraph are partners in the European project Intergeo [12]. One of the goals of this project is to allow the exchange of construction files between different interactive geometry systems. JSXGraph thus allows to load construction files created with GeoGebra and display them as interactive applets embedded in a web page. So, using JSXGraph is a viable possibility to get GeoGebra files to work interactively both on mobile phones as well as desktop or laptop computers.

When GeoGebra users export a web page with an interactive applet, they expect the applet's construction in the web page to work in exactly the same way as it did within the GeoGebra desktop application. This is currently the case for GeoGebra's JavaSE applets on desktop and laptop computers as the underlying software is identical. Using another system like JSXGraph introduces three important differences compared to GeoGebra's current JavaSE applets.

1. Different features: JSXGraph offers a wide range of features. However, there are commands and object types in GeoGebra like conic sections, implicit polynomials, and others that are not supported directly by JSXGraph at the moment.

2. Different look: even if a feature is common to both systems, the result may look different on screen simply because the two software applications may use slightly different drawing algorithms internally. For example, a function plot of $\lfloor x \rfloor$ may look different when plotted by JSXGraph or by GeoGebra.

3. Different dynamic behaviour: even if a construction looks exactly the same after loading in JSXGraph or GeoGebra, it may behave differently when it is changed dynamically by e.g. dragging a point. That can happen when some of the internal computation algorithms work differently, e.g. they may handle the order of the four intersection points of two ellipses differently.

These differences are simply due to the fact that JSXGraph and GeoGebra are two different pieces of software. In principle, it is possible to work around most of these differences by adapting GeoGebra's export routines or adding new features to JSXGraph. Unfortunately, this would mean that a lot of time would have to be spent to make the two systems compatible both for existing features as well as for each new command added to GeoGebra—with the possibility that some of the differences can not be overcome at all.

Thus, while we initially intended to use JSXGraph as the new standard way to export GeoGebra web applets, we started to look for alternatives that would allow us to find a better maintainable solution for fully compatible GeoGebra applets on mobile devices.

3.2 Porting GeoGebra to JavaScript

Creating native ports of GeoGebra for mobile devices (e.g. iPhone App, Android App) or for other web browser plugin technologies (e.g. Adobe Flash, Microsoft Silverlight) could solve the compatibility problem of supporting all GeoGebra files on mobile devices. However, it would essentially mean to start from scratch and rewrite GeoGebra; possibly multiple times. For an open source project with a small team of developers this is not a realistic option.

From JSXGraph we had learned that it is possible to create well-performing interactive graphics applets that will work in any modern web browser both on mobile devices as well as computers. The technology behind JSXGraph is JavaScript [13], a scripting language for web pages not to be confused with GeoGebra's programming language Java.

So, we started to look into ways how GeoGebra could be ported to JavaScript in order to ensure maximum compatibility between the current Java desktop application and exported GeoGebra web applets based on JavaScript. While this still involves a significant amount of work, it would only require one single port of GeoGebra to support standard web browsers on a wide range of devices instead of targeting each mobile platform separately.

The following sections will describe some of the technical details of our GeoGebraMobile project where we decided to port GeoGebra from Java to JavaScript.

4 JavaScript: the rediscovered technology

The origin of JavaScript dates back into the 1990s when the Internet was young and dominated by the Netscape [14] web browser. Netscape's programmers developed a scripting language called Livescript which was later renamed to JavaScript, in the hope that the name similarity to the well-known Java programming language would help its popularity. Due to the widespread success of JavaScript as a client-side scripting language for web pages, Microsoft developed a compatible dialect of the language, naming it JScript to avoid trademark issues. [15]

In the early days, JavaScript was often (ab)used to create visual effects on web pages, earning it the reputation of a 'non-serious' programming language. One of the first widely accepted 'serious' usages of JavaScript was Google's GMail web application in 2004 [16]. With its revolutionary use of asynchronous JavaScript and XML (AJAX [17]) to load web server contents in the background without the need of a page refresh, it pushed JavaScript back into the focus of web developers. Leading programmers began to rethink JavaScript's capabilities and many books were written about the AJAX technology that has drastically been changing the Internet during the last years.

Today's web applications have to be responsive and quick, even on narrow bandwidth or high server load. Meanwhile, the upcoming HTML5 standard [18] also embraced JavaScript as the language of its programmable features. In particular, JavaScript is used with the new canvas element, a 2D graphics panel which has become widely available in modern web browsers since 2009.

> The canvas element is currently supported by the latest versions of Mozilla Firefox, Google Chrome, Safari, and Opera. It is not natively implemented by Internet Explorer (IE) as of version 8, though support is in development for Internet Explorer 9, however many of the Canvas element's features can be supported in IE, for example by using Google or Mozilla plugins, JavaScript libraries and either Adobe Flash or IE's proprietary VML. [19]

As a side note, JavaScript programs are sometimes said to run much slower than for example Java software. While this may have been true several years ago, due to the increasing importance of JavaScript for web programming, its implementations in modern web browsers have been improved significantly during the last years. Various organizations are currently competing in the informal 'JavaScript engine race' [20]: the Mozilla Foundation (TraceMonkey and JägerMonkey), Google (V8), Opera Software (Caracan and Futhark) and Apple (Nitro).

5 Google Web Toolkit

For the reasons mentioned above, we decided to develop GeoGebraMobile as a web application from scratch instead of using another tool like JSXGraph. In this way, we hope to be able to achieve full compatibility to GeoGebra's desktop Java application by using the calculation algorithms of the GeoGebra Java kernel. This will allow us to reuse thousands of existing GeoGebra files and make them available for mobile devices.

However, this approach means that we need to translate GeoGebra's source code from Java to JavaScript, two quite different programming languages. A manual translation would be extremely daunting, given the fact that GeoGebra's source code is more than one hundred thousand lines long and would take years to be rewritten in JavaScript. Furthermore, errors in the manual translation to JavaScript would very likely introduce new bugs and incompatibilities compared to the original Java version.

Thus, we started to experiment with various tools that could help to automatically convert our source code from Java to JavaScript (e.g. [21], [22]). Finally, we decided to use Google's Java-to-JavaScript compiler, a component of Google's Web Toolkit (GWT [23]) that has been developed for creating rich web applications (like GMail) using only the Java programming language. With GWT, Java programmers can use their favorite developer tools and a large set of available libraries in Java. GWT will then perform a sophisticated translation from Java code to highly optimized JavaScript code which runs in all HTML5 compliant web browsers.

5.1 From Java to JavaScript

In this technical subsection we give a brief overview of how we are creating the JavaScript web application GeoGebraMobile from GeoGebra's Java code using the Google Web Toolkit. Currently, the goal of GeoGebraMobile is to allow the use of GeoGebra applets that only show the interactive graphics view in a standard web browsers both on computers as well as on mobile devices. In the future, we may also add other user interface elements like toolbars or other views like GeoGebra's spreadsheet view to GeoGebraMobile.

In order to create the GeoGebraMobile web application we are following these steps:

1. Standard Java source code: We start from GeoGebra's Java source code written for Java Standard Edition, version 5.

2. GWT Java source code: As GWT does not support all features of JavaSE, we have to manually change parts of the original Java sources to get Java code that is compatible with the Google Web Toolkit. These manual changes mostly affect the way objects are drawn on the screen and can be kept minimal by using add-on packages for GWT that substitute missing JavaSE packages used in GeoGebra.

3. JavaScript application: Finally, the GWT compatible Java source code of GeoGebraMobile is compiled using the GWT Java-to-JavaScript compiler. The compiler compresses the javascript code, so the resulting optimized JavaScript application can be thought of as the 'binary' of GeoGebraMobile that runs on any standard HTML5 web browser.

So, all programming is done in Java, allowing us to merge updates from GeoGebra's Java desktop application into GeoGebraMobile using standard source code versioning and merging procedures with minimal manual interventions. In this way, we only need to maintain one core Java codebase of GeoGebra that is used to create both a JavaSE desktop application as well as a fully compatible JavaScript web application.

5.2 Some Implementation Notes

GeoGebra's software architecture uses the Model-View-Controller [24] design pattern where the mathematical kernel (model), graphics view (view), and user input handling (controller) are separated into different packages. GeoGebraMobile follows the same design pattern and package hierarchy as the original Java desktop application.

Most of the mathematical kernel of GeoGebra is compatible to the GWT application programming interface and can be translated to JavaScript automatically. Some missing features are added by using additional libraries like gwt-java-math [25] to support e.g. BigDecimals. In many cases the manual changes can be done by simple changes of package import statements.

The graphics view of GeoGebra is implemented using the Java2D drawing features of JavaSE which are not available in GWT, so we are using graphics libraries that extend the capabilities of GWT in this area:

- gwt-geom [26] is a port of the OpenJDK java.awt.geom package to add support for e.g. Color and AffineTransformation objects to GWT.

- GWTCanvas [27] is used to create and draw to the HTML5 canvas element. We decided to choose this library among many other GWT based canvas implementations because it is already part of the GWT Incubator. This means that GWTCanvas will eventually become a standard part of the Google Web Toolkit and will be supported and improved in the future by Google's developer team.

Some parts of GeoGebraMobile had to be coded by hand. This includes both the basic application framework that initializes an applet as well as the loading of constructions. GeoGebra uses an XML file format [28] and the XML parsing class had to be rewritten because the original routines were not compatible with GWT. GeoGebraMobile currently supports worksheets where the construction is given as a base64 encoded XML string using JSON and AJAX technology. In the future we also plan to support construction loading from standard GeoGebra (.ggb) files. Previous work of the JSXGraph project on loading GeoGebra files will be very helpful here.

The first attempt to translate GeoGebra's Java source code to JavaScript resulted in about 6000 compiler errors of the Google Web Toolkit. We thus decided to start small by only converting a few classes needed for a minimal GeoGebra web application that draws two points and the line between them.

For example, implementing GeoGebra's Point object in the kernel and graphics view required conversions which are mostly automated by the GWT compiler. Some manual coding was required to add missing graphics packages and adapt the original source code for them. To make such tasks easier, we are using the integrated development environment Eclipse [29] that speeds up finding incompatible parts of the original code and GWT and also helps in identifying ways to work around such problems.

6 Current Status and Browser Compatibility

A working prototype of GeoGebraMobile showing the currently supported features can be found at [7]. As of summer 2010, GeoGebraMobile applets already support points, segments, lines, conic sections and most commands for these object types. The implementation progress of GeoGebraMobile

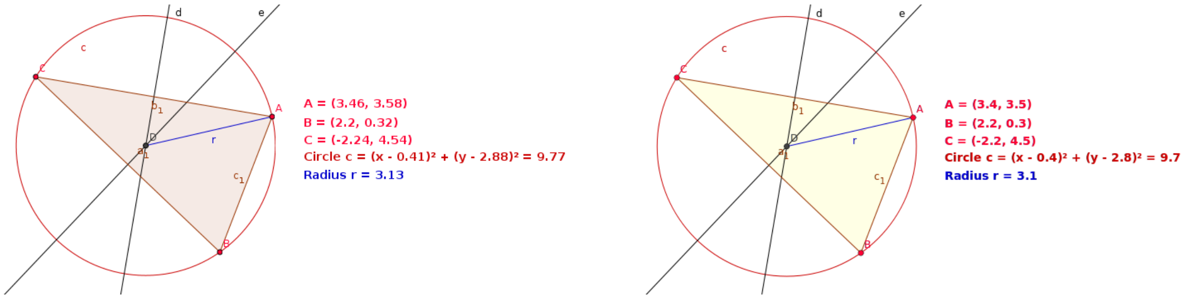

Figure 1: GeoGebra as a Java applet and a JavaScript application

is documented in our development blog [30] and the source code can be found at [31]. During the last year, the GeoGebra code has been merged several times into the GeoGebraMobile code branch without any conflicts. Thus, we are able to keep the development of the GeoGebra desktop application and GeoGebraMobile applets synchronized efficiently.

On desktop computers, GeoGebraMobile applets currently work well with the latest versions of the following web browsers: Mozilla Firefox, Safari, Google Chrome, Opera, and other browsers with the same rendering engine. It does not work however with Microsoft Internet Explorer 8 as this browser doesn't implement the HTML5 canvas element yet. This situation will change with Internet Explorer 9 which will support HTML5 better. As mentioned above, there are several ways to work around this limitation of Internet Explorer 8 which have been successfully applied by JSXGraph and could be included with GeoGebraMobile as well. For GeoGebra, this situation is not crucial as we can still use JavaSE applets with Internet Explorer 8 or older on desktop computers.

Concerning mobile devices, GeoGebraMobile has been successfully tested with the latest web browsers on Apple's iPhone, iPod Touch, and iPad devices as well as on Google Android phones. For iPad, iPhone, iPod, and Android we have implemented touch support that handles one finger events for dragging and selecting as well as two finger zooming. As GWT doesn't support touch directly, we are using its JavaScript Native Interface to embed JavaScript code into our GWT Java code. So far, touch support is the only place where we are using manually written JavaScript code as a bridge to GeoGebraMobile's Java code.

7 Applet Startup on Computers

While GeoGebraMobile is really about bringing GeoGebra applets to mobile devices, it is also interesting to compare the user experience when loading a web page with a GeoGebra Java applet compared to a GeoGebraMobile JavaScript applet in a web browsers on a desktop computer.

Thus, we have performed some experiments on various desktop computer environments (Windows, Mac OS X, Ubuntu Linux) with broadband Internet connection where we have cleared the Java cache (Java 6, OpenJDK 6) and web browser cache (Safari 5, Firefox 3.6). Loading an online web page with a simple GeoGebra Java applet from the GeoGebra server took between 5-15 seconds on different machines while loading a page with a similar GeoGebraMobile JavaScript applet took only about 1-2 seconds on the same machines.

After closing the web browser, restarting it and revisiting the same pages, all the files are loaded from the Java cache or the web browser cache. In this case, the Java applet needs between 4 to 14 seconds to display while the JavaScript applet page shows instantaneously.

The size of the Java applet files and GeoGebraMobile's JavaScript files are about the same with approximately 250 KB each, so download size cannot explain the different loading times on the first visit. This is also confirmed by the cached timings on the second visit of the pages where nothing needs to be downloaded and all files come from the hard disk cache. The big time difference is due to the loading time of the Java plugin when the first Java applet is used in a web browser. Subsequent pages with other Java applets are then also loaded instantaneously.

To work around this slow startup of the Java plugin, the GeoGebra website uses Java applet preloading [32] where an invisible Java applet is loaded in the background, making subsequent GeoGebra applet loading instantaneous as the Java plugin has already been started and the applet files are cached. However, this approach fails in our experiments above when a Java applet web page is opened directly at a time where the Java plugin has not been started yet.

As a consequence, a student or teacher waiting more than 5 seconds for a web page with a GeoGebra Java applet to load may think that the page doesn't work and just leave it. GeoGebraMobile applets clearly improve this situation: web pages with its JavaScript applets are loaded in a few seconds without the need for workarounds like Java applet preloading. In addition, GeoGebraMobile applets don't require any browser plugins as they rely only on JavaScript and HTML5 which is part of all modern web browsers.

8 Conclusion

The project GeoGebraMobile is under active development since 2009 and will soon allow the use of GeoGebra applets in modern web browsers both on computers and new mobile phones (e.g. iPhone, Android) and tablets (e.g. iPad). The software is fully compatible with the GeoGebra desktop application as they share the same Java code base. This will allow students and teachers to use GeoGebra's large pool of dynamic mathematics materials on virtually all devices with a web browser.

In the future, GeoGebraMobile will be the standard way of exporting interactive web pages from GeoGebra. Test versions are already available on the project website [7] and the first stable release is planned with GeoGebra 4 in summer 2011. Future work will strive to support all features related to GeoGebra's graphics view including equation rendering and computer algebra support. Furthermore, the door is now open to create a fully fledged GeoGebra web application that doesn't need any browser plugins or desktop installation.

With GeoGebraMobile, teachers and students using GeoGebra can now be sure that their materials created with our software will be (re)usable on a wide range of devices ranging from desktop computers to mobile phones for many years to come.

References

[1] GeoGebra – free mathematics software for learning and teaching

[2] Java Standard Edition overview

[3] OpenJDK

[4] Java Micro Edition overview

[5] Android

[6] Gartner Newsroom: Touchscreen Mobile Device Sales Will Grow 97 Percent in 2010

[7] GeoGebraMobile

[8] International GeoGebraWiki

[9] JSXGraph – Dynamic Mathematics with JavaScript

[10] JSXGraph and SVG

[11] JSXGraph browser support

[12] Intergeo – Interoperable Interactive Geometry for Europe

[13] JavaScript (Wikipedia article)

[14] Netscape Navigator (Wikipedia article)

[15] ECMAScript (Wikipedia article)

[16] GMail (Wikipedia article)

[17] B. W. Perry, Ajax Hacks, Tips & Tools for Creating Responsive Web Sites, O'Reilly Media, 2006

[18] HTML5 (Wikipedia article)

[19] Canvas element (Wikipedia article)

[20] JavaScript Engine: Performance evolution (Wikipedia article)

[21] j2s (SourceForge project page)

[22] yui4java: A New Java-to-JavaScript Translator Using YUI 2

[23] Google Web Toolkit (Wikipedia article)

[24] Model-view-controller (Wikipedia article)

[25] gwt-math (Google Code project page)

[26] gwt-geom

[27] GWTCanvas (Google Code project page wiki)

[28] GeoGebra XML Format (GeoGebraWiki article)

[29] Eclipse

[30] GeoGebraMobile blog

[31] ggjsviewer Subversion repository

[32] Unsigned GeoGebra Applet (GeoGebraWiki article)