

Beyond Applets: Introducing Mathwright Microworlds

Dr James E White
Director of the
Mathwright Library

mathwrig@gte.net

It is my purpose in this short paper to describe a new type of web document that has recently made its appearance at the New Mathwright Library and Café (<<http://www.mathwright.com>>) and to discuss it in two lights: from the point of view of its readers (students of mathematics), and from the point of view of its prospective authors (teachers of mathematics). These documents are called Mathwright Microworlds. You are welcome to visit a few of our Microworlds by downloading our free MathwrightWeb Control at <<http://www.mathwright.com/mathwrightweb.html>>

A Mathwright Microworld is like an applet, with three differences. First, while it runs in Java in the reader's ActiveX enabled browser, it is generally faster than an applet because it uses a mathematics engine that is housed in the MathwrightWeb Control — running on their machine. Second, its object-oriented design and colloquial interface generally present a more versatile and expressive learning environment to the reader than applets can. Microworlds often span several pages, and they can unify and integrate ideas by illustrating them from several points of view, often with colourful and dynamic simulations. And third, authors create these Microworlds with a simple point-and-click interface that is much easier than Java to learn.

A student once made an observation about her algebra course: "Solving equations wouldn't be so bad. But every time you find the value of x , they change it on you." Nothing could be truer in the world of educational computing. Paradigms come and go with the stochastic regularity of late-Summer Hurricanes off the Florida coast. For the bewildered teacher and the beleaguered student, the question for the educational software developers is: "What is constant?" Gleefully, the developers reply: "What is a constant?"

And this only reflects the temperament and the climate of the larger world of computing, in which the beaches of our minds have been battered (invaded?) with wave on wave of new and ever more powerful technology. This is not a bad thing. We have passed from the Spartan "Command Line" to the expressive and colourful "Multimedia" environments of CDs and the Internet in a few short years. But one thing is constant for students of Mathematics. That is the importance of experimental, open-ended, and even gratuitous thinking, through problem solving, to the formation of clear concepts.



Mathwright can be found at:
<http://www.mathwright.com/>

However it is packaged, the content delivered by the new media should first lead the student, not directly to answers, but instead to new questions. With the appearance of the Internet, there arose, almost overnight, the largest repository

of “information” in the history of our culture. But information alone has little value unless we know how to use it. As the Internet became increasingly “interactive” it began to offer us “verbs” in addition to the “nouns” of the information world.

We are now able to “act” as we browse, and to manipulate the “facts” in increasingly useful ways. This is obviously useful for purchasing concert tickets online, and it is useful in a less obvious way for students of Mathematics. If students are to ask questions that are meaningful to them in a virtual environment, then the computer environment should have the flexibility and the “intelligence” to attempt to answer them.

A virtual learning environment is more useful to the extent that students can ask a wider variety of questions within it. Now with the appearance of Java-enabled web browsers, it became possible to put small interactions within the browser itself. These take the form of “Applets.”

But mathematical applets are a somewhat paradoxical solution to the problem: How to bring interactive and live mathematical learning environments to students on the web. They are paradoxical because, while they make use of some of the most advanced and sophisticated technology available to the web, and they harness the power of a truly elegant and powerful language: Java, they have placed the creators of these applets in the prehistoric conditions of the early '80s, where programmers had to build pretty much everything from scratch. So we see a proliferation of dedicated “appliances” on the web, each devoted to the illustration of some particular mathematical point, with very little - if any - communication among them.

It is precisely the possibility of communication among tools that can lead to the integrated experiences for students that can elicit new questions and arouse their curiosity. Of course, it is the virtue of Java that programmers can easily develop reusable class libraries that make possible an object-oriented approach to the development of mathematical applets - building them from components. But very few teachers have the time or the inclination to use Java in this most elegant way to place mathematical explorations on their own web pages. So they (and their programmers) often have to “reinvent the wheel” again and again to do such simple things as parsing input expressions, or drawing graphs.

What is needed to create versatile, “smart,” and expressive virtual environments is a powerful Java class library that can integrate all the mathematical

components (computer algebra style representation, parsing, and simplification of algebraic expressions, rational, complex and decimal number types, functions, vectors, matrices, sets, differential equations, and so on) with the objects that display them (graphing windows that can display sprite animations, data tables, windows for displaying mathematical formulas correctly, as well as matrices, etc.).

And these must of course be tied to built-in programs that handle such tasks as solving equations, differentiating and comparing algebraic expressions, solving differential equations, or inverting matrices, and so on. To be effective and useable, such a class library needs a high level object-oriented language, so that authors can actually turn their ideas into learning environments that students can use easily.

MathwrightWeb is our class library, and MathScript is our high level object-oriented mathematics scripting language for manipulating it. With these tools, we create Mathwright Microworlds, which are multipage interactive web documents that run in the browser. In a sense, MathwrightWeb is like an “Applet” that can “play” an infinite variety of interactive lessons and stories on web pages. But unlike an applet, the language only needs to be downloaded once. It is reused for each story that it tells. And the approach has the virtue, which we have only begun to explore, that the “stories” themselves may be presented in HTML (eventually MathML, perhaps) and can take advantage of the variety of presentation skills that the web itself provides.

One of the essential skills that is present in each Microworld is a basic understanding of the conventions of algebra — the ability to simplify and compare algebraic expressions, and to transform them in meaningful ways. A Microworld can represent its answers to student questions in a form that students can understand, usually without any special attention by the author. This includes graphical, geometric and symbolic representation. Part of the reason for the versatility of our virtual environments is the fact that our “machine language” is LISP written in Java. It has long been known that LISP is an ideal language for the artificial intelligence techniques that underly computer algebra. What is perhaps not so well known is that LISP also lends itself nicely to the object-oriented design of our Microworlds themselves by our authors.

The collection of over 160 interactive WorkBooks at the New Mathwright Library and Café is currently being translated to Microworld format, which is more colourful, more versatile, and mathematically more perspicuous than our old 16-bit format. These Microworlds can

provide unlimited opportunities for the reader to explore in the browser nearly any topics in the Primary, Secondary, or Undergraduate Mathematics Curriculum, from the exact arithmetic of fractions, to geometry and combinatorics, to abstract algebra.

Among the capabilities that LISP provides to our Microworlds are:

- Exact rational arithmetic and algebra, e.g. the ability to compare variant forms of algebraically equivalent expressions.
- Function definitions (functions of one or several variables, single or vector valued) and variable assignments
- Six number types, including rational, complex and quaternionic, and unlimited precision decimals or integers
- Expansion of algebraic expressions, and comparison of different forms
- Derivative calculations, including iterated partial derivatives to any order
- Vectors and Matrices, vector and matrix manipulation, vector-valued functions, and differential operators
- Solving differential equations interactively, drawing tangent fields
- Automatic equation solving with explanations (linear, quadratics, cubics, etc.)
- Rewriting rules (created by authors) that apply functional identities to simplify expressions (trigonometric, exponential, and logarithmic, for example) with explanations at each step.
- Scripted explorations with sprites and graphics in which gadgets such as buttons, scroll bars and checkboxes control interactions on the screen.
- Graphing functions and curves with 16 million colours, arbitrary pen width. Interactive zoom-in, zoom-out, and tracing.
- Definition of “geometric” sprites that may be transformed by arbitrary affine transformations as they move.
- Wallpaper background for Graph2D windows.
- Factoring integers and polynomial functions of one variable with rational coefficients.
- Forward and backward chaining with Unification Pattern Matching.

Finally, we realize that a browser is not necessarily an ideal place to think hard about mathematics. So we give students the opportunity to visit any Microworld on the web with MathwrightWeb, and to create what we call a “Disk Version” of the Microworld on their machine. This is usually much faster than visiting the Microworld

in applet mode anyway, because the Microworld comes across in compressed form and is automatically decompressed on the client machine.

Once this is done, the student may continue to read the Microworld at any later time with our free Mathwright32 Reader, which is an application that runs independently of the browser. Thus, they may later read these Microworlds offline if they choose to, without having to be connected to the web at all! They need only go to the web once, to get the Microworld in the first place.

Mathwright32 Reader is actually quite a bit faster than MathwrightWeb. In this mode, using Mathwright32 is essentially the same as using the Mathwright 2000 classic with our WorkBooks. Students may create their own documents, programs, notes and reports and share them among themselves and with their teachers.

Along these lines, our Mathwright32 Author program that is used to create these Microworlds finally brings to web authors a point-and-click WYSIWYG interface that, together with our object-oriented mathematical scripting language, reduces the time required to design interactive mathematical web pages from months to days, and with experience, sometimes to a few hours. While it creates Java Microworlds, it is much easier than Java to learn. That program will be available at the Library in November.

It is one thing to read about them, and quite another to see them. We invite you to visit the Library and do just that. Among the Microworlds already available at the Library to be viewed in your browser are: Exploring Quadratic Functions, <http://www.mathwright.com/book_pgs/book502.html> (a 9-page book that gives step-by-step symbolic and graphic solutions of quadratic equations and inequalities that the reader may supply), Cardano <http://www.mathwright.com/book_pgs/book512.html> (a 13-page book that uses both abstract algebra and graphical experiments to illustrate Cardano’s approach to solving cubic equations from a novel point of view), Mathwright Logo Playground <http://www.mathwright.com/book_pgs/book501.html> (a Logo programming environment with multiple turtles in which young readers can create and save sophisticated Logo interactions among the turtles).

Like an applet, a Mathwright Microworld is designed to elicit and invite reader questions by presenting a simple User Interface. But a Microworld has the power and range that a generic applet can seldom achieve and this makes possible a wide variety of reader experiences. It invites students to play. Take a look, see what our authors, all over the world, are up to.