

The Tabula Rasa

From a thousand words, a picture

by Kurt Cagle

There is something magical about the first issue of any publication. Every writer knows this feeling, every artist. The first issue is the blank page, the empty canvas, the lumps of unshaped clay awaiting the sculptor's hands. That first issue is all potentiality that has not yet broken its symmetry and come into existence. It is perfection, and yet, the moment that the first word gets written down, the first brush stroke smears its color upon the pristine pane, that inhuman perfection crystallizes into a human manifestation of meaning and intent, skill and talent, with luck shaping the world far more than that sterile potential ever could.

When I went to college all too many years ago, I remember working with a vapor discharge terminal and an operating system called Plato that uses an electronic gun to draw brilliant orange images and letters on the screen, one stroke at a time. The system could render photographic images—raster images—at best poorly, so it instead utilized the principle of programming to produce its imagery one instruction at a time:

moveTo 0,0; drawTo 126,16; arcTo 17,192...

These shape instructions became known (somewhat inaccurately) as vectors, and the pictures so formed were vector graphics. That you could use mathematical commands, the very antithesis in many people's minds to the creative spirit of the artist, in order to build a picture may seem counter intuitive, yet it is precisely this that underlies Postscript, the language employed by both printers and applications in order to *render* a page filled with text and graphics. Vector graphics help computers to *objectify* graphics, to turn a drawing into the distinct operations by a set of objects. It is much as if the Sorcerer's Apprentice had chosen not brooms but brushes, buckets of paint, pens, and masks to do his bidding, each engaged in an intricate dance to do what it was designed to do best. The artist ceases to be the hand that wields that brush or pen, but instead becomes the conductor who instructs the brush and pen to do what they must.



Plato, a prehistoric operating system, uses an electronic gun to draw brilliant orange images and letters on the screen one stroke at a time.

Five years ago, a sorcerer's conclave (the Graphics Working Group of the W3C) met to discuss these various pens and brushes, and more importantly, the ownership of the language of the spells which gave these homunculi their orders. The meeting was contentious, because at its heart lay the fundamental question of ownershipwhether the instructions to build these graphics should continue to lay within the province of the tool vendors and hence only be exposed by their effects, or if, instead, the language should be an open standard that anyone could use to produce the tools to create the graphics. In the end, a few of the wizards walked out angrily, including a couple of the mightiest.

The rest, however, took their absence (perhaps slyly) as assent, and proceeded

to build an open language called Scalable Vector Graphics, or SVG. As with many computer names, this involved more than a few ironic entendres—all vector graphics can be intrinsically rescaled without loss of resolution, but the *scalable* here also referred to the ability of the SVG system to scale beyond the bounds of an ordinary graphic.

Suppose, for instance, that I created a graphic that repeats well in two dimensions. With SVG, I can actually create a reference to this graphic as part of a pattern and use that to fill another graphic-even if the original graphic existed on a server a thousand miles away from the image that utilized it. What's more, when fully implemented, both graphics can in turn respond to the actions of the viewer, can change their aspect even as they are being viewed. Like HTML, their thousand words defined not just a simple delineation of actions, but the existence of a complex ecosystem of semi-aware shapes and groups of shapes. The drawing, once drawn, may continue to change and morph and interact with the world around it.

Yet ironically, what was fashioned at that conclave and subsequent ones like it, is remarkable not for this seemingly magic quality of the images that the language makes possible, but the very nature of the language itself. There are any number of such vector formats in existence, but all but one of them are predicated upon the notion that a given vendor controls the specification. They can arbitrarily change the language in order to give them a competitive advantage or to lock out competition, and ultimately there is an uneasy question about whether the graphics so created with the language belong to the artist or to the vendor who supplies the brushes. >> continued on page 8



In this number we have Christopher Lewis and Alastair Fettes talking to us about the Open Source project known as SPARK, SVG Programmers Application Resource Kit, and how to use it.

Creating SVG widgets with SPARK GUI framework

There have been numerous widgets created by the authors of SVG based web applications and a number of attempts to create standard widget libraries (kevlindev widgets, SVgUI project, CGUI, dSVG). While each of these efforts makes a valuable contribution to the field of SVG GUI, they've required a great deal of duplicated and arguably wasted effort. One of the major hurdles in the creation of a standard SVG GUI library identified by the originators of the SVG Programmers Application Resource Kit (SPARK) was the lack of a published reference framework, which would allow implementors to easily add new widgets to an existing library, and ensure interoperability with future widgets.

With this objective in mind, the SPARK project was initiated, and the SPARK GUI Framework (SPARK-FW) was developed to aid in the development of reusable, interoperable, SVG based widgets. This article introduces the reader to the SPARK-FW. It will show what is required to construct a widget that conforms to the framework, and how that widget can be used in a simple web application.

The SPARK home page can be found at: http://spark.sourceforge.net. The Etch-A-Sketch application developed for this article can be found under *samples* at: http://spark. sourceforge.net/resources/samples/

Figure 1: Screen Shot of Jog Dial. The picture on the left shows the dial in the resting state and the picture on the right shows the dial as it is rotating in the clockwise

cumulative total of events fired.



in the clockwise direction. A text element has been added to the SVG document which displays a value that is incremented as the dial rotates to the right and decremented as the dial rotates to the left. This text element is for illustration purposes only and is not actually present in the final widget; in the illustration the text element displays the

Creating a SPARK compliant widget

A compliant widget consists of two parts: an SVG Document Fragment and an ECMAScript Object. When the document is loaded, the entire document is parsed; elements that describe the widgets are passed to the SPARKFactory, which creates the widget. The FW defines two primitive types from which all widgets inherit: • The *atom* is a self contained widget that will not contain other widgets and may be used as a target for user input.

• The *container* is a widget that may contain other SPARK widgets and will likely not act as a target for user input.

The SPARK framework imposes a few requirements on the SVG Document Fragment describing a widget:

• The fragment will be contained in a top level element whose class specifies the type of the widget.

• This will be of the form "SPARK [atom|container] WidgetName WidgetStyle".

- The SPARK keyword indicates that this is a SPARK-FW compliant element.
- The second indicates that this widget is either an atom or a container.
- The third is the name of the widget (i.e. Button, Slider, Jog Dial).
- The fourth optional entry can be used to provide different widget styles via CSS.
- The SVG must provide a "desc" element describing the widget
- The SVG may provide a metadata element containing initialization parameters for the widget.
- The SVG must include a body that defines the widgets appearance. This should be included in a group element with a meaningful class attribute.

For instance, in this article we will create a Jog Dial—a widget that allows the repeated generation of positive or negative (i.e. up/ down, left/right) events that can be used to control a target. You'll see a Jog Dial on many new keyboards and PDA's, it is also synonymous with the scroll wheel available on most mice. Example 1 presents an SVG Document Fragment that could be used to describe such a widget in the SPARK-FW.

EXAMPLE 1

```
<g id="jog-1" class="SPARK atom JogDial"</pre>
 transform="translate(50,50)">
 <desc>Jog dial widget</desc>
 <metadata eventfreq="250"></metadata>
 <g class="SPARK JogDial base">
   <circle r="40.0" style="fill:#009999" cx="0.0" cy="0.0"/>
   <g>
     <circle r="30.0" style="fill:#99ff66;"</pre>
       cx="0.0" cy="0.0" />
     <path style="stroke:#009999;fill:#99ff66;"</pre>
       d="M -20 -40 Q -10 -20 0 0 10 -20 20 -40"/>
     <animateTransform id="jog-1-rotateRight" ... />
     <animateTransform id="jog-1-rotateLeft" ... />
   </g>
   <rect id="jog-1-dial-left" fill="white"opacity="0.0001"</pre>
     x="-50" y="-50" width="50" height="100"/>
   <rect id="jog-1-dial-right" fill="white"opacity="0.0001"
     x="0" y="-50" width="50" height="100"/>
 </g>
</q>
```

Example 1: SVG document fragment describing SPARK Jog Dial

This SVG Document Fragment provides a circle with an arrow that marks the top of the dial and can be rotated either clockwise or counter-clockwise depending on which animation is started. It was drawn with an available open source WYSIWYG SVG Editor and can be seen in Figure 1 (opposite page).

In this case we've decided that the Jog Dial will be a SPARK atom. Therefore the ECMAScript object extends Atom (not shown). It implements mousedown and mouseup handlers to begin and end the animation of the Jog Dial and to start and stop a timer which increments or decrements the value of the Jog Dial. Example 2 shows the portion of the mousedown event responsible for starting the clockwise rotation of the dial.

Example 2: Method that starts the clockwise rotation of the Jog Dial

The closure is necessary to ensure that the increment (incr) method is called on this instance of the Jog Dial by the timer which is started on by the mousedown event. This is so that you can hold down the mousebutton

EXAMPLE 2

```
JogDial.prototype.mousedown = function(evt) {
 if(SPARK.requestMouseFocus(this) {
   var thisObj = this;
   if(evt.target.getAttribute("id") == this.ID + "-dial-left"){
     this.animate = evt.target.ownerDocument.getElementById -
       (this.ID + "-rotateLeft");
     this.animate.beginElement();
     JogDial.active = function() {
      thisObj.incr();
      JogDial.timeout = setTimeout("JogDial.active()", →
       thisObj. eventFreq);
   };
     JogDial.timeout = setTimeout("JogDial.active()", 40);
   }
   evt.stopPropagation();
 }
}
```

and have the events continue to fire. The frequency with which events are fired is determined by the "eventFreq" member, which is extracted from the metadata specified in the SVG Document Fragment.

You'll notice that, in addition to starting the animation, the mousedown handler calls the "incr" method. This method is responsible for triggering a positive event from the Jog Dial. This is accomplished by giving the Jog Dial a positive state via a call to the setState method, which in turn notifies any observers that the state of the widget changed and runs any commands associated with the widget (Example 4). The notifyObserver method is part of the Observable interface which is implemented by all SPARK widgets. The Observer design pattern allows other objects implementing the pattern to be notified of changes in the state of the Observed object. The runCommands method is part of the CommandHolder interface which is implemented by all SPARK Atoms. The Command pattern allows functionality to be added and removed from otherwise generic components.

Example 3: The increment and setState methods

When the increment method is called, it updates the state of the widget, which in turn runs any associated commands and notifies any observers that the widget has been updated.

To demonstrate the use of the Jog Dial, we produced a simple Etch-a-sketch application (Figure 2, this page). This application consists of two Jog Dials that control a cursor on the screen. As the cursor is moved, a line is drawn on the screen. The first Jog Dial controls vertical movement and the second controls horizontal movement. *>> continued on page 7*





Our publisher Michael Bolger recently travelled to Xamlon, Inc., a fascinating new company located in La Jolla, California, to talk with founder and CEO Paul Colton. Paul is the creator of Live Software and JRun, the latter now owned by Macromedia (Adobe). Xamlon's eponymous product focuses primarily on Microsoft and Macromedia technologies, but does include limited SVG support, though SVG looms larger in Xamlon's future via the efforts of Xamlon developer Kevin Lindsay and others.

MB> I would just think that in Semantic Web applications, SVG is going to be a major player.

PC> Yes, I would think also. I want to make clear, we are about targeting whatever works, we are not tightly bound to Flash, Flash is just a good target now, the problem is that browsers are all different. If there could be a standard out there that I know that I can generate, let's say SVG, XHTML, fine, all we are after is unifying the developing experience. The developer does not want to learn yet another API, let us do that work for you, we're just going to take your code and spit it out to the very latest and greatest technologies available in that platform. Unfortunately every browser is different, including same browser with different versions, it's a nightmare for developers. You need a compelling unified experience.

MB> With the pending Adobe-Macromedia merger, will they support Mobile SVG? PC> They might, SVG Tiny, they might, I am all for it, that would be great, all we want is consistency, we want to write once and deploy to all mobile devices, unless any of those is the exact same, everywhere, it won't really work, so the one player, whoever, it might be SVG right, whoever can be the most consistent across all mobile devices is the winner.

MB> With all the XML features in .net 2.0, how exciting is that getting to be?

PC> What we're saying is, still use .net 2.0 and all the great tools, as a development language but disconnected from the deployment target. What we're trying to accomplish here is use as much of .net as you can, maximize as much of .net you can then throw it up to the web—flash—including web services and everything, essentially one IDE, right now Visual studio, we are planning on Java, Eclipse.

MB> What about Mono?

PC> We are considering it, we're not bound to XAML, we don't have to be our code, of course whole pieces are specific to Microsoft stuff, we purposely made sure we could target any other type of markup. **MB>** How about SVG in the work flow?

PC> We have SVG converters in, we probably want to do some of them out. Clean SVG is, because of the complexities, it's hard to out a clean beautiful SVG or to suck in someone's hand written SVG, mung with it and get it out with it still beautiful, that's hard with any XML.

MB> The X windowing people are working on SVG for the desktop.

PC> Okay, so that would be a competitor to XAML. My philosophy is getting all this stuff to work together as best it can, of course you're building tools that are going to make it a reality, you know we have, it's a simple one, Kevin wrote it, fairly effective Adobe SVG to XAML converter, we would like anything to come in, Native SVG, go for it, we get it into the final product. Our ultimate goal to unify, best case wouldn't it be great, I like Java, I like C#, done, Eclipse or Visual Studio done, ultimately SVG.

Xamlon is one of new companies that we will be talking about more in the future. For our next issue, join Michael as he bites deep into Apple to find about an emerging and important partnership between Apple and KDE over Scalable Vector Graphics.



Sensorial expressions and emotion by Domenico Strazzullo

Boethius (480-524 ad), Latin scientist, philosopher and mathematician, was certainly the first digital artist. According to this remarkable thinker (De Institutione *Musica*), by the divine reason, all things were established in harmony after the order of the numbers. This order existed in the intelligence of the Creator. Thus, the number is the principle of all things, and the music according to Boethius is nothing else than the science of numbers that govern the world. During the Roman and Gothic epochs the word "music" designated harmony in its broader sense, and was even defined as "science of all proportion, whatever it may be" (Edgar de Bruyne, Etudes d'esthétique Médiévale 1946). There we have official recognition of digital arts that backdates 1500 years!

> "It is important to accept that the canons of beauty are not arbitrary."

Much later, electronic scientific instruments, appliances and programming logic made possible to scientifically corroborate Boethius' digital visions and this ultimately led to the flourishing of computer art, not only as a medium for "analog" art reproduction but as an original tool for artistic expression. Among Computer Arts disciplines we are particularly interested here in graphic arts. Like with other forms of art, one important distinction can be made for graphic arts: decorative versus fine arts. This series will attempt to help young artists to get insight as to the fundamental meaning of that distinction, to help establish the notion of the self as "the creator", to discover and develop inner vision.

A good place to start is to try to define and understand what "aesthetics" stands for. According to the dictionary, aesthetics is:

"1. The branch of philosophy dealing with such notions as the beautiful, the ugly, the sublime, the comic, etc., as applicable to the fine arts, with a view to establishing the meaning and validity of critical judgment concerning works of art, and the principles underlying or justifying such judgments 2. The study of the mind and emotions in relation to the sense of beauty."

Um...perhaps we had never suspected it was a branch of philosophy. The first definition fits perfectly what an art critic does and we may feel uncomfortable with the notion that aesthetics, as a science, has "...a view to establishing...the principles..." and we might show some resistance: "what principles? I'm making new ones tomorrow!". This attitude is legitimate and even encouraged, at some point and to some degree. Let's put the first definition on the side temporarily.

The second definition attracts our attention in that we feel affectively concerned. Since we are artists and, aggravatingly enough, of the inquisitive type, we need to know more about the sense of beauty and its dependency to emotiveness, both of the creator and of his public. Our five senses serve, or served, primarily for survival needs. In the microcosms of the human brain, as an epitome of the world, like in that of other superior animals, we cannot say if capability of intelligence was an implemented feature or if it was triggered accidentally by playful activities; but we can safely admit that the senses being used more and more in conjunction with communication, leisure and playful activities, was an expression of growing intelligence. The caveman coming back from a hunting trip, having seen some dangerous animal, would run away, then back in his cave relaxing, he would sit and draw what he had seen, for others to know.



A manuscript page from Boethius' "De Artithmetica", circa 520. His thoughts were of great influence throughout the Middle Ages.

> That was probably the first example of a newsletter. He must have found some pleasure in that drawing activity since we are still doing it, telling stories. The first aesthetics rules were gradually in the making with tacit consentience, that is, before scholars were around, probably based on simple observation of natural shapes with their colors and shadows. The degree of sophistication of the concepts that make up aesthetics has become very complex since, although the emotional exchange, one of the ends of artistic expression, has remained in its basic meaning.

> It is important to accept that the canons of beauty are not arbitrary. In respect to mankind as a whole, they may be so, but certainly not in respect to individuals. The collection of aesthetics rules with the addition of aberrations generated by the artistic thought results in a delicate balance which may respond to two contradicting feelings: the need for security, with its attachment to nature, to the known, to the safe, and the need for discovery. This can be compared to those very singular human activities, that of the scholars who spend most of their time codifying and normalizing, and that of the creators who spend most of their time breaking the rules. These two activities are in fact in the scheme of a feedback trial and error system, the aberrations being in >> continued on page 7

SVG (for lack of a better name)

Protect your image by Peter Schonefeld

As the name suggests, the Scalable Vector Graphics format will allow you to build vector graphics and, if you wish, scale them. But if we look at the draft specification for SVG1.2 in the very first sentence we find "SVG is a language for describing two-dimensional graphics and graphical applications". OK, we get the first bit...but what's with "graphical applications"??

The truth is SVG is much more than just a medium for vector graphics. SVG also has capabilities that relate to raster images and other non-vector media. Throw in animation, pixel based filter mechanisms and its uses for building graphic user interfaces for computer programs and we have something that's much more than just SVG, perhaps it should be called Scalable, Scriptable Vector and Raster Graphics, Sound and Animation (or SSVRGSAA for short). Now I'm just being silly.

It's clear that we can look at SVG as something much more than just a vector graphics format, but don't take my word for it. In this and hopefully following articles, I'd like to show some of the uses for SVG that will add a great deal of value to graphics on the web. First cab off the rank is the following blurb that shows how we can embed copyright information into a web graphic using the Creative Commons License. The two left images *(below)* both appear to be raster, not vector, images. Correct. However, while the first image is a jpeg, the second is an SVG document with a jpeg image embedded within it.

> "It's clear that we can look at SVG as something much more than just a vector graphics format..."

As an amateur artist, I don't really stand to lose much if my image is copied and used against my intentions (of course you do know my intentions for use of this image...just by looking at it, don't you?). However, through Internet piracy, some professionals do stand to incur loss to their pocket and/or reputation if work is used in a way that they don't intend. At the end of the day, amateur or pro, it doesn't matter who you are: it's your right to protect your work.

Unfortunately, today on the web if you're an image pirate you might use the defence, "Well I thought that because it was on the web, it was free. Nothing told me otherwise."

I may not be in a position to analyze the law around breach of copyright, but I can show you how to better protect your images by using the tools provided at www.creativecommons.org.

From their web site: "Creative Commons has developed a Web application that helps people dedicate their creative works to the public domain—or retain their copyright while licensing them as free for certain uses, on certain conditions".

Now that's what we want! Typically you add a cc license to your web site, but that to me is still classed as fine print...let's be clear about what we want (or don't want) done with our images.

Notice the creative commons watermark on the image. That "watermark" is not actually a watermark in the traditional sense of the term. It sits above the image, like a layer. We can interact with the cc "watermark" so let's click on it....

As you can see in the third image, it is now clear what people can do with the image and even who made it. Further, when your image is used at a web site other than your own, if it's embedded in an SVG document, your license information (or whatever info you like) can travel too! For something like this to work, it needs to be simple to do and it helps if many people use it, so that the function of cc becomes common knowledge. The good news is that it is easy to do. A template page with the SVG and code that's generated from creativecommons.org can be found with the source code for this article. If you were to examine the SVG code for this document,

The first image on the left is a JPEG image. The second shows an SVG document with a JPEG image embedded within it. The third image on the right shows the same SVG document after clicking on the "CC" watermark.



you can see SVG is an XML based format just like XHTML. As for people using it: Did you know that this works in Firefox without installing any other program? It'll also work in IE with the Adobe SVG viewer.

Perhaps as individuals we can do something to make protecting our rights even easier. Contact the vendor of your favorite graphics package and ask them to include "Save as SVG (with CreativeCommons)" as an option in their programs. So, I hope from this short blurb you can see that SVG is useful, for more than vector images alone, even if it's called that for lack of a better name.

The source code for this article can be found at: http://www.svgf.org/code/peter

Jog Dial and those required to use the widgets

in an application. This widget can be used

together with the existing SPARK widgets

and should be usable with any forthcoming SPARK widgets. Using the widget in the

sample application is as simple as attaching

a command to the widget at the time that it

is created. 🚯



A new web application solutions company providing superior application development services for enterprise clients. We are leading experts in the field of rich standards-compliant web applications.

Creating SVG widgets >> continued from page 3

Cursor movement is started by a Command attached to the Jog Dial. This Command is attached to the Jog Dial via the SPARKDecorator when the widget is created in the SPARKFactory. The details of the Decorator and Command can be seen in the Etch-A-Sketch source which is available on the SPARK project page.

This article presented the main steps required to create a SPARK-FW compliant

Sensorial expressions >> continued from page 5

turn codified and normalized, eventually. Originality is incompatible with the first activity and essential to the second.

"It don't mean a thing if it ain't got that swing"

As artists we are particularly concerned with establishing the meaning of the second definition of aesthetics: "The study of the mind and emotions" in the sense of the metaphysical *what* and *why* we seek. This will help us understand the meaning of poetry. In a broad sense we can call poetry the emotiveness engendered by any form of communication, that thing without which any expression is sterile or even boring (but not necessarily meaningless), failing to awaken an emotive reaction in the spectator. Meaning that if the creation lacks or fails to convey emotion, it doesn't fulfill one of its basic purposes of taking the spectator for a trip out of his daily occupations and preoccupations, feeding his soul with exciting discovery or, at least, entertainment.

In the next article we will explore more deeply the impact of physiological perceptions on the emotional construct.

By the way, Boethius had to sustain torture and execution. (?)



www.vectoreal.com

Scalable Vector Graphics Enschede, The Netherlands August 15 - 18

506 Open 2005

SVG Open 2005 conference University of Twente Enschede (the Netherlands) August 15-18

4th Annual Conference on





SVG Open 2005 PO Box 6 7500AA Enschede The Netherlands SVG Open 2005 is organised by: University of Twente Elemente Sponsors Vectoreal

 Tabula Rasa is published by The SVG Foundation,

 949-257-0011, 23016 Lake Forest Drive, A472,

 Laguna Hills, CA 92653

 http://tabularasa.svgf.org

 © 2005 The SVG Foundation

WC WORLD WIDE WEB

Executive Publisher Art I

Art Director Jayne De Sesa

Production Manager Sara J. Porter

Associate Editors Peter Schonefeld Ruud Steltenpool Ronan Oger

The Tabula Rasa >> continued from page 1

SVG is an Open Standard, in the same way that HTML and XML are. An open standard is much like a contract, in that it specifies what you can and can't do with it. You can create graphics without having to pay a license fee for that permission; create tools that will help others create those graphics without having to pay a license or fearing that someone can revoke your right to work with the standard. You can see (and to a great extent participate in the development of) the standards as they are being created, and can know that changes that occur in those standards are stable once the development process is complete. What you can't do is prevent other people from also utilizing the standards, nor impose a penalty upon them if they happen to be a competitor of yours. An open standard is ultimately a level playing field.

Not surprisingly, certain companies that are perfectly happy with playing fields that have hills and valleys (especially where they command the hills) have not been thrilled to see SVG. At first they chose to ignore it, then as more and more small companies and developers began to see the benefit of the language and built tools around it, they began to deprecate it. Meanwhile the tools moved from specialized viewers to editors, and most recently SVG has begun to scale the ramparts of the browser, the platform it was originally intended for. In the withering barrage of criticism and scorn heaped upon it by its detractors, the role and purpose of the language has become more clearly refined. Curiously enough, every time those same detractors shake their heads and say that SVG is not suited for this or that purpose, lo and behold some enterprising soul proves them wrong.

The last five years, however, have simply set the stage. We have only just reached a point where scalable vector graphics are now available to the average non-programmer, yet most people are not aware of the power which is at their grasp. This is the role of both the SVG Foundation and this magazine. Over the ensuing weeks and months and years we hope to inform and highlight and educate (and maybe, just maybe, to entertain). We hope to promote SVG and related technologies, to recognize those people who follow a similar mission, and to act as a bridge between the SVG community, the standard setters and the tool makers.

If you have a story to tell, a work of art to show, a tool to promote, or even just a thought to express, please contact us at editors@svgf.org. The canvas is primed let's go paint a masterpiece.

SIGGRAPH VISITORS

The year 2005 will likely be seen in retrospect as the year that SVG hit the big time. If you are a graphic artist, a developer, a content manager, a cartographer or anyone who needs to deal with graphics as being more than just pretty pictures, you will have one of the best opportunities to learn more: on August 15-18, the SVG Open 2005, the 4th Edition of the International Conference (and exhibition) on Scalable Vector Graphics will be held in Enschede, the Netherlands, a conference where the cutting edge artists and developers get a chance to immerse themselves in the frontiers of 2D computer graphics.

In the last year a great deal has happened in the world of SVG. All of the big mobile phone companies now offer many phones with SVG (-Tiny) pre-installed, indeed some have models left that don't support it. This year as well is seeing SVG move into the browser market, with five of the six major browsers in use today promising SVG support by year's end.

SVG Open 2005 promises to offer a smorgasbord of new technologies and implementations for artist, programmer, and manager alike. The program includes courses and workshops on topics ranging from SVG animation to who's doing what with SVG to using Scalable Vector Graphics within browsers and cell phones. Keynote speakers will include Simon Kendall, CTO of multi-national car insurance company EurotaxGlass that uses an SVG system for all its damage reports, XML guru Kurt Cagle, talking on the future of web graphics, W3C SVG WorkGroup leader Chris Lilley, and many others. You'll also get a chance to talk to vendors and developers of SVG services on the convention floor.

Please take a look at http://www.svgopen.org/2005 for more information, including the full schedule, registration form, and sponsors. SVG Open 2005 is organized by the University of Twente, Telematica Instituut and ITC, The International Institute for Geo-Information Science and Earth Observation, and is affiliated with the W3C. We look forward to seeing you there.

-Ruud Steltenpool, Chairman, SVG Open 2005

1

Manaaina Editor

Domenico Strazzullo