Introduction to Inkscape

Why you may want to use Inkscape and how to get started.

DMITRY KIRSANOV

When novice artists think of creating artwork, they usually think of programs like The GIMP, Photogenics or some other program with features similar to Adobe's Photoshop. Sometimes these programs are the appropriate tools. Sometimes they aren't. Many people end up working harder than they have to because they are trying to use a program like GIMP to do things that are more appropriate for a drawing tool.

If you have ever used CorelDRAW, you have a good idea of what a drawing program does. Inkscape is something like CorelDRAW. If you haven't used CorelDRAW, here's an example of where a drawing tool like Inkscape would come in handy. Suppose you want to draw a cartoon character without a "sketchy" look to it. You want the outline of the character to be composed of solid lines, not lines that look like pencil sketches. And, you want the lines to be connected perfectly in all the right places so that you can color-fill different parts of your character without the colors leaking through a hole in your outline to another part of your character. The problem is that you don't have the skill to draw the perfect solid-line outline of a cartoon character using the freehand drawing feature in GIMP. Worse, if you draw the character in GIMP and then decide that you have to enlarge it or shrink it, the outline becomes distorted. It may even develop "holes" through which color fills bleed.

Here's what you would like to do, instead.

You'd like to draw the solid lines of your cartoon character with all of its features (shirt, hands and so on) the best you can, and then adjust the shape and position of these lines until you've got it just right. If you discover that you've drawn the character too big or too small, you want to be able to change the size of the cartoon character outline without distorting your hard work.

This is the sort of thing Inkscape does extremely well. In this case, you might want to start with Inkscape and finish your cartoon character in GIMP. It doesn't matter. The point is that Inkscape does some things better than other programs, and you want to use the best tools for the job whenever possible.

VECTOR GRAPHICS

Inkscape makes the drawing task just discussed possible because it is a vector editor. You may have no idea what vector graphics are, but you probably have heard the term SVG by now. More and more programs, Linux distributions and devices are boasting support for SVG (Scalable Vector Graphics).

What's the buzz? What is a vector anyway, and how does it make graphics scalable in such a way that all other graphics are not?

Let's start with a simple thought experiment. Imagine that you have before you a simple image—say, a black disk on white background—and you need to describe it to your friend over the phone, so that she can reproduce it as exactly
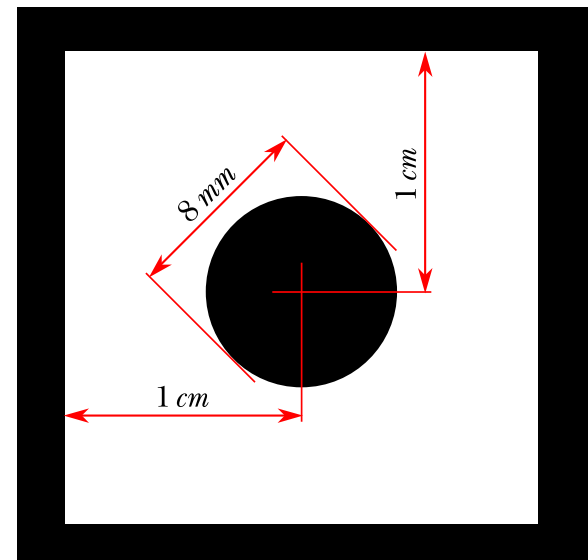

Figure 1. The Perfect Disk Described

as possible. How would you go about that?

Well, you'd probably say something like, "draw a solid disk, colored black, diameter...let me check...eight millimeters precisely, and its center is one centimeter from the top and from the left." That description (Figure 1) indeed gives a complete representation of the image; there's nothing in it that was not mentioned. Such a vector representation of an image is short, easy to understand (at least for humans) and so natural that you may not realize that different approaches to this simple task exist.

The bitmap of the same image (Figure 2) is what a computer would create based on this blueprint.

Programs like GIMP let you enlarge the disk,

Figure 2. A Bitmap Version of the Disk

but the end result is less perfect than the original. You can improve the end result by using anti-aliasing, but even an anti-aliased version can never be as good as the vector-based original disk.

VECTOR SCALABILITY

In the vector blueprint of the black disk image, you can easily replace millimeters with meters and draw the perfectly round disk the size of your backyard with no loss of quality. When you have a bitmap, however, scaling it up inevitably produces jaggies and that annoying pixelated look—because you are, in fact, scaling the pixels, not the image.

The point is that if you stick with vector graphics for your circle, you don't have to redefine the size and start from scratch to get a perfect circle. You can expand the original or shrink it without losing any of the fidelity of the original circle. You do not need to do any anti-aliasing or perform any other tricks. That's why scalable vector graphics are called scalable. They scale up or down without distortion.

If you still want to have a bitmap image when you're done, it's easy to convert an SVG image to a bitmap. This is called rasterization. Although it is easy to go from vector to bitmap, it's not easy to go from bitmap to vector.

In a way, the difference between vector and bitmap is similar to that between the source code and a compiled binary of a program. Source is for humans; binary is for computers. Source is the "idea" of the program; binary is a specific "implementation". The entire Open Source movement grows from the premise that you can make this world a better place by sharing ideas, not implementations. Can we apply the same principle to sharing images too?

WHY SVG?

The idea of vector representation of images is not new. In fact, it may be even older than the concept of a bitmap. However, until recently, vectors were used mostly in professional design workflow, whereas the images on the Web and on typical end users' desktops were almost exclusively bitmaps.

One reason for that was the lack of a single, universally applicable and universally accepted, vector graphics standard. PostScript, PDF and Flash are all vector formats, but they are specialized and therefore limited in various ways. Here's where SVG's promise lies: a modern XML-based language, created by the W3C, can certainly give a huge impetus to the use of vector graphics everywhere. SVG (http://www.w3.org/Graphics/SVG) is a powerful (some would say overpowered) language with provisions not only for geometric shapes but lots of other things—gradients, filters (including blur), pattern fills, fonts and even animation.

Of course, SVG can't replace bitmaps outright, if only because not all graphics can be represented as vectors. This works fine for disks and other geometric shapes, but what about photographs, for example? Well, for one thing, a powerful modern vector language such as SVG *can* do amazingly photorealistic graphics. And for another, SVG can *embed* bitmaps into its vector image, treating them just like any other object (for example, you can overlay an embedded bitmap with vector captions and callouts, which is what I did in Figure 3).

But perhaps the more serious obstacle in the way of SVG is users' inertia. As mentioned previously, many people tend to think in bitmap by default, even when vector makes much more sense. Say "computer graphics", and more often than not you'll hear "Photoshop" or "GIMP" in response. Yet among the images that a typical user deals with daily, perhaps only digital photos require strictly bitmap processing. All the rest—

drawings, logos and Web graphics, diagrams and flowcharts, headings and clip art, postcards and wedding invitations, even random scribbles—is much easier and more natural to do in vector.

It's difficult, however, to evangelize about the vector bliss when the only decent vector graphics tools (such as Illustrator, CorelDRAW and Xara X) are limited to Windows and/or Mac, use proprietary formats and are quite expensive. Fortunately, this has changed recently with the advent of Inkscape (http://www.inkscape.org)—a free multiplatform vector editor with SVG as its native format. In its latest version (0.42 at the time of this writing), Inkscape has become seriously powerful and very usable.

INKSCAPE

The interface of Inkscape (Figure 3) is easy for newbies. The big bright buttons in the vertical toolbar on the left are inviting—and, more importantly, they do exactly what you think they do. Click on the blue rectangle, then draw on the canvas—you get blue rectangles. Click on the red ellipse and drag—you get red circles and ellipses. Fun!

But the real fun of vector graphics is not drawing; it's "editing".

Create some rectangles and disks as described above. Click on the topmost button (with an arrow) on the vertical toolbar—it's called the Selector tool—and then click on any of the objects (rectangles or ellipses) you just created. Immediately that object becomes selected—you see it framed in a dashed box, and eight handles appear around it. A selection can contain any number of objects; use Shift-click or drag objects around with the Selector tool and watch the status bar at the bottom of the window, which tells you how many objects and of what kind you have in selection.
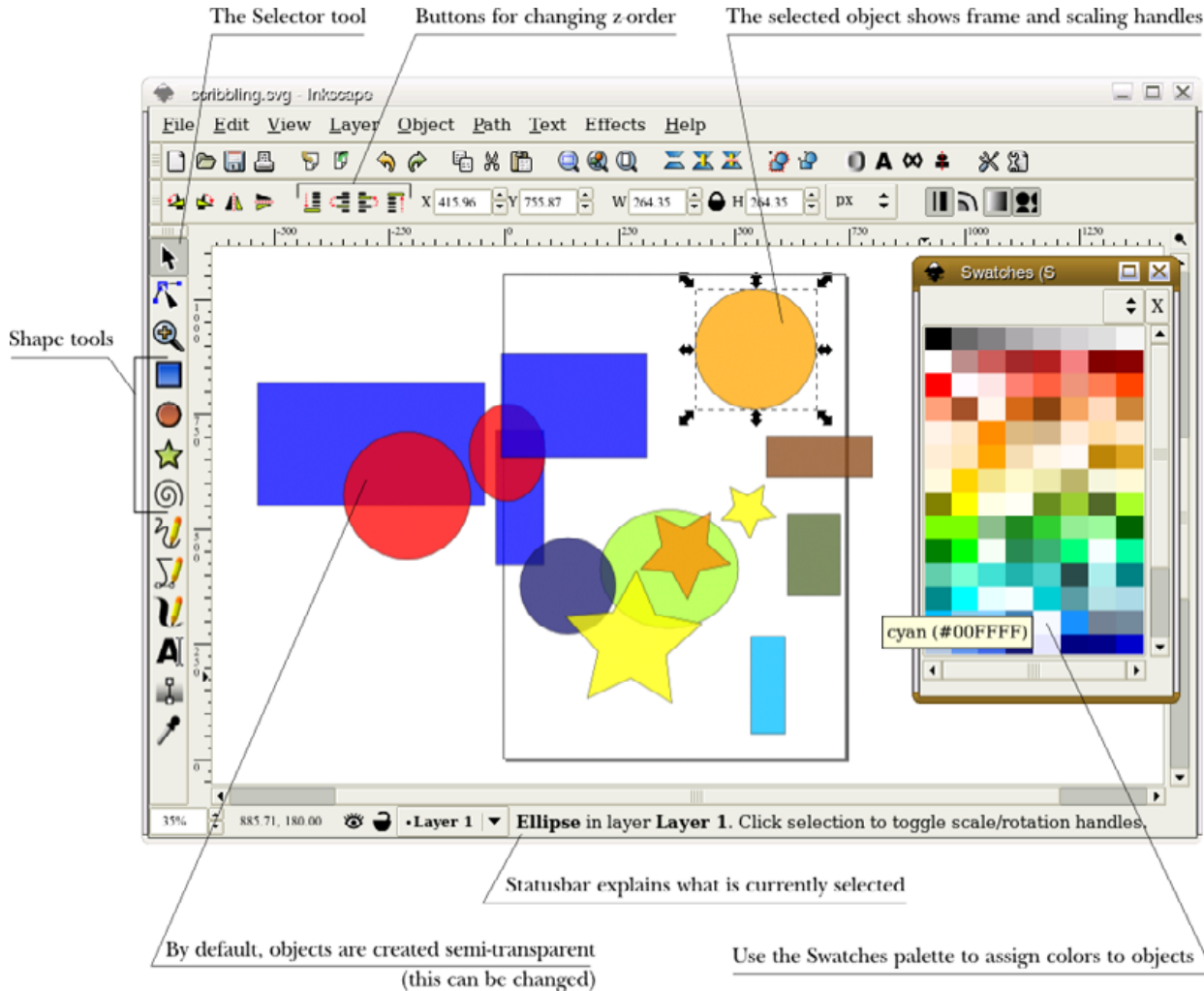
Think of what you've done. If you've ever had

The Selector tool — Buttons for changing z-order — The selected object shows frame and scaling handles

Shape tools

By default, objects are created semi-transparent (this can be changed)

Statusbar explains what is currently selected

Use the Swatches palette to assign colors to objects

Figure 3. The Inkscape Interface

Figure 4. Stretch your objects easily.

Figure 5. Rotating an Object

some experience with bitmap editors such as Photoshop, you may know how difficult it often is to separate something (for example, a human silhouette) from the background, and how poor the results can be. With vectors, all you need to do (in most cases) is a single click. Objects in a vector image are never merged or flattened; the drawing always remembers what it consists of. It's like putting each single stroke and shape onto a layer of its own, except that it's done automatically for you.
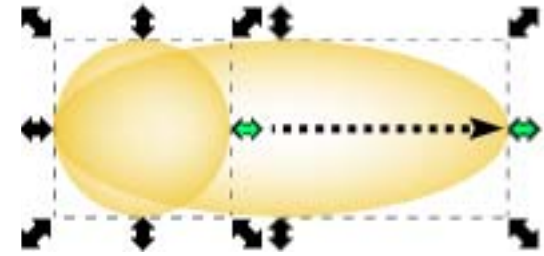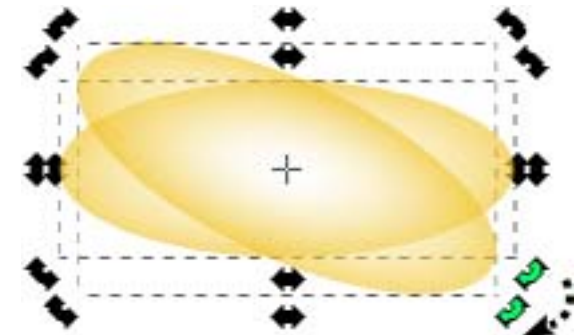
So what can you do to a selected object? A lot. You easily can drag it around to wherever you like, and you can stretch or scale it to any size by dragging the handles (Figure 4).

If you click the object again, the handles change and you can rotate the object (Figure 5). You can also easily skew the object.

Don't like the color? Use the Fill and Stroke dialog to change it. You can bring up this dialog by right-clicking on the object and selecting Fill and Stroke from the pop-up menu. You should see a dialog like the one in Figure 6. There are several ways to change the color. You can play with the sliders in the default tab, or try some of the other tabbed tools.

The Fill and Stroke dialog also lets you change the opacity of the selected object. You can change the stacking order (or z-order as it's usually called)
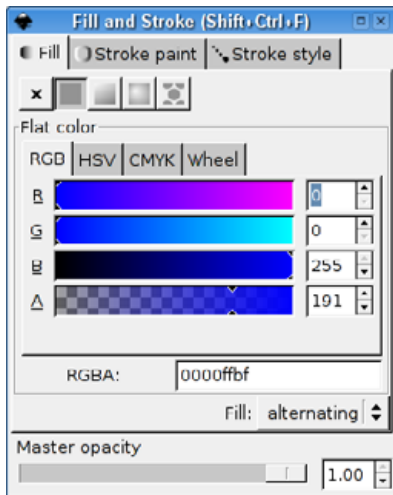
Figure 6. Fill and Stroke Dialog

of objects as well; use the Raise/Lower commands in the Object menu. You can combine several objects into a group—a single composite object. Or, simply press the delete key and the objects you have selected are no more.

DRAWING WITH INKSCAPE

Of course, you're not limited to geometric figures. You also can do freehand drawing (Figure 7).

Here, too, every stroke is an object in its own right, selectable and modifiable separately from others. For example, you can use the node tool (the tooltip says "Edit path nodes or control handles"), which is the icon right below the arrow selection tool. You also can press F2. This tool lets you reshape a line you have drawn. Simply grab (click on) one of the nodes that
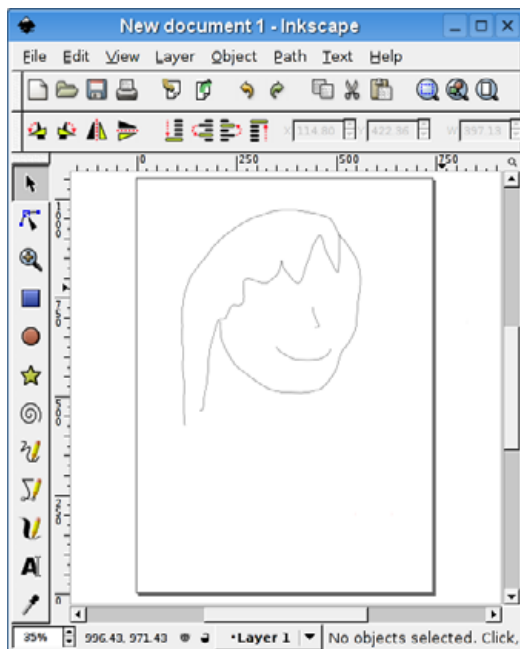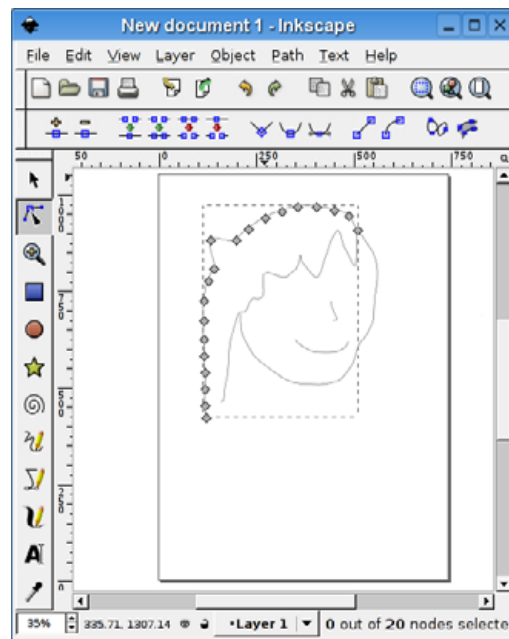
appears, and pull it in any direction to modify the line (Figure 8).

A very nifty Inkscape tool is the Calligraphic pen, which is very convenient for making drawings—either technical or artistic. You can create sketches where every stroke is an object (Figure 9). There are 1,116 individual lines in this sketch.

This article should give you a general idea of what Inkscape is for and what it can do. I hope you will take it from here and explore the program on your own.■

Figure 7. The Beginning of a Face

Figure 8. You can adjust lines without erasing or starting over.

Figure 9. An Artistic Sketch with the Calligraphic Pen

Dmitry Kirsanov (http://www.kirsanov.com) is an independent author, designer, artist and consultant in the areas of XML, XSLT and SVG. His most recent book is *XSLT 2.0 Web Development* (ISBN 0131406353).