David Hart didn’t really choose to work with computer-generated images. Instead, they chose him. “I don’t remember exactly, but it started pretty early,” he explained. “General interest in computers and videogames fueled my desire to learn some programming, which I think started around age 8, on any computer I could find, usually one of the Atari 400s or Commodore 64s in the neighborhood.”

A few years later, he started dragging his father to computer animation festivals, and that was the clincher. Throw in a master’s thesis in shadowing techniques in the computer graphics program at Cornell University and a year of research for the University of Utah, and Hart found himself working at PDI/DreamWorks for five years. “I learned so much there I really consider that part of my academic background too,” he said.

The rest is about Blots. Taking inspiration from a Siggraph paper by Karl Sims titled “Artificial Evolution for Computer Graphics,” Hart killed time on commuter trains by developing his own generative evolutionary software titled, Blot. Rather than work with stand-alone images, the evolutionary approach just sort of wormed its way into his creative plan of attack.

“Blot is just one experiment among many,” he said. “Most of what I’ve done, for both work and play, has been the direct approach, utilizing more traditional processes. But I do want to see how far an evolution paradigm can go. I want to find out whether this could one day become a traditional process, whether it could be a tool any artist can use.”

Seeding the forms

Hart describes the current state of Blot as “about 42,000 lines of weekend hobby-style C++.” Since he wrote it mainly for his own projects, the current UI is a little on the minimalist side, but this will eventually change, Hart said.

All images here are cross-slices of evolutionary processes that Hart’s software generated. The cover image, Blot 2005/08/20/93, is one example. “When I made this particular image, I was just exploring and having a little fun with hexagonal forms,” he explained. “I seeded my software with the simple expression ‘sin(3*a)’, where ‘a’ represents ‘angle’ in polar coordinates. This produces six radiating wedges. From there I played until something interesting came out.”

Many of Blot’s images start in this fashion, by seeding the evolution with a simple geometric expression. For example, with Figure 1, titled Blot 2006/02/04/67, Hart began by exploring circular forms instead of hexagons. “The underlying basis in math is just part of the fun for me,” he explained. “I can play with simple equations to make a basic geometric shape, and, from there, I’m able to come up with these types of images to feed the math nerd in me.” Figure 2, Blot 2005/03/27/48, and Figure 3, Blot 2005/11/15/31, are also cross-slices of similar generative processes. “Making images using artificial evolution is new and exciting because it’s a very different way of thinking, but it can also be difficult to control,” he said. “It’s easy to explore and play...
with, but as soon as you have a specific idea that you need to make happen, evolution can be like trying to write calligraphy with a fire hose; it’s very powerful but the small details are at best hard to direct.”

Behind the scenes

Hart refers to the process behind the scenes as “interactive evolution” primarily because it’s not autonomous. The artist makes all the evolutionary decisions, and Hart says this matters a lot to him, as artificial evolution is generally used to solve organizational problems without human intervention. He’s not working within a “survival of the fittest” analogy, which he says doesn’t apply to the creation of art anyway. A good chunk of how Blot works is demonstrated by the artist controlling the process directly.

“I start with a very simple equation, like sin(x),” he explained. “I then tell Blot to mutate this equation, and it comes up with 25 random variations, like 1.58 * sin(x - 1) or cos(x) or sqr(sin(x)). These are all one or two simple steps away from the original equation. All the variations are plotted so I can see them. I don’t generally even look at the math expressions at all, just the plots. I pick a small number of them to ‘survive,’ typically only one or two. Then the survivors are mated and mutated to produce a new generation of variations. If I just keep picking the ones I like for long enough then, the theory goes, I can evolve aesthetics into the images.”

Hart also hopes that understanding the math is not automatically a requirement for the viewer to appreciate the images. “Despite the underlying process being mathematical, which is interesting, it fails as art if you have to understand the math,” he said. “I have, however, noticed that people who understand math or programming tend to appreciate my pictures a little more often than people who don’t—I think because it’s easier for them to understand the story behind this work and relate to me.”

Further growth

Hart declares that Blot will by no means stop evolving; he’s recently been cleaning up the whole shooting match and making it more user friendly. For experimenters who want to get their hands dirty, the source code is now available at http://blot.sourceforge.net/.

“I think an implementation that evolves shader programs for today’s graphics hardware would be fun to make and play with,” he noted. “It would also be fun to do another animation, perhaps even try to make something story-based and less abstract.”

Since he’s also starting to make forays into the realms of children’s software and educational toys, Hart feels that now is also the right moment to concentrate on overhauling the UI, since interactive evolution is designed to hide complex goings-on behind simple picking interfaces any-