

The Nature of Technology

John Maeda

There is a tendency to perceive technology with distaste because it is not natural.

Consider how when a writer introduces herself as a “writer” to strangers. An immediate moment of recognition by her audience occurs—an image in the mind forms of the completely detached virtuoso frantically hovering over the keys on her typewriter, with many balls of crumpled paper and huge stacks of typed pages that surround her desk. A nod of respect to the writer is given to show respect to her craft and creativity.

Consider how when a painter introduces herself as a “painter” to strangers. An immediate moment of recognition by her audience occurs—an image in the mind forms of an unkempt genius in mental battle with her canvas, with the strong smell of drying pigments and visual disarray in spats of color on the walls and floor. A nod of respect to the painter is given to show respect to her craft and creativity.

Then consider how when a programmer introduces herself as a “programmer” to strangers, there is no immediate moment of recognition by her audience. “What does a programmer do?” There is no cultural reference to this act of programming by the majority of people. There are no feature films that romanticize the creator of computer codes as some kind of artistic genius. No, instead the programmer is depicted as some kind of evil and unfriendly genius. “Ah!” her audience vaguely recognizes this term “programmer” and musters the only conclusion that can be ascertained from general culture, “You are a programmer ...” and ends on a tone of indifference and will likely wish to change the topic of conversation.

Programming is generally not widely understood as a creative activity. We do not usually learn this skill in school along with other creative disciplines such as writing and painting. And when it is learned today in schools as “computer science,” usually the subject matter is on the “how to use” a computer versus “how to make” a computer line of thinking. This is no surprise, as we do not have many teachers in the world that can teach programming as a creative discipline. Should programming be taught as a creative discipline? I used to think so. Strongly so.

In 1999 I created a system called “Design By Numbers (DBN)” as a gentle means to introduce computer programming to visual artists as a simple language for drawing in a 100-pixel square box. The system is freely available and covers the key aspects of computer code in a day’s worth of rigorous exercises. Two of my students went on to create a much more advanced system called “Processing” which has enjoyed recent success for its handling of all the visualization features that any Flash master might be aching to explore. As many art and design schools have begun to adopt programming of the DBN, Processing, or ActionScript variety into their curriculum I see that there is an increasing interest in tapping into the added power of creating codes.

However, today I find myself increasingly less interested in the idea of programming, and more in the idea of ideas. Great programming does not necessarily beget a good idea;

however the reverse is always successful. A good idea goes a long way. Whether if that idea is made into a painting, crafted into a poem, or even rendered as a computer algorithm. I think that when we stop worrying about the technology and focus more on the ideas, we are approach a needed universality in appreciation.

Ten years ago I was at UCLA talking to a senior faculty person about a recent graduation critique. The professor complained to me about how a student was showing his multimedia work, and then halfway through the presentation he stopped and began to explain to the faculty committee how difficult it was for the student to create his piece. Of course the professor thought this to be ridiculous—“Who cares about the technology? All that matters is the idea!” he said. I quickly took the student’s position and explained to the professor, “Naturally the student was upset—you, the ‘faculty,’ had no idea as to how his construction was made. If the student were to have woven a tapestry (which was that particular professor’s lifelong expertise), he would not have had to bother to demonstrate that he had achieved mastery. On the other hand, the student was so beyond his own faculty that he was simply expressing his frustration.” The professor thought for a moment, and vaguely acknowledged that I might be right, but in the end I knew he didn’t really get it.

I think many people “get it” now, as evidenced by the unique array of work assembled for this exhibition “The Domino Effect.” As many more dominoes fall, we can expect even more people to emerge and to collectively appreciate the new media not for its newness, but for its ability to add to the wonderful spectrum of expressions capable by the human mind.