

The Art and Craft of Open Source

Code-based art (or “software art”) really has little in common with what the general public calls “software.” Software is something that allows users to do things with their computers. Code-based art is more about enabling computers to do something creative, artistic. Although many assume the computer to be the medium, it's actually the code itself that becomes the medium. Therefore, the concept of “open-source” in code-based arts has an additional and very different function than “open-source” in the software development world.

One of the things that seems to be missing from the code-based art community, is any serious dialog regarding code as medium. We don't see many open and active discussions regarding actual processes and methods. Perhaps this is due to the fact that the processes and methods aren't visible. Code-based art isn't like painting where you can take a close look at the work and see how the artist lays down the strokes. We don't get to see how the artist handles the medium. It's like seeing an instructional piece by Sol Lewitt without the instructions. One of the first steps towards meaningful discussion would be the recognition of code as medium rather than computer as medium. To say that the computer is the medium is like saying that a painters canvas is the medium.

One of the things that is frequently done in code-based arts, is to convert to audio or visuals a source of data. More often than not, this data must be manipulated or transformed in some way before it can fit into the framework of the artists presentation of it. How might this transformation affect the concepts involved in the work? What portions of the data affect what aspects of the presentation and why? Let's say we see a piece that creates visuals based on the wind speeds in New York City, London and Paris. Is the fact that the piece always has a reddish tint to it due to the data itself or is it because the artist intended it to be that way? In fact, how much of the visual representation is due to conscious construction on the artist's part and how much is due to the data itself? We don't know but it could tell us a lot about the work. We could even derive entirely new and meaningful concepts from the work.

Of course, it's easy to dismiss all this by saying, but the people viewing this art can't understand the code anyway. But I believe that if the artists, simply by making the code available, make it relevant to an understanding of the work, those interested will make the effort to extract some kind of meaning from it. Before you know it, we'll be hearing about "boldly executed while loops" and "delicate tints of 0xEE88FF" as well as "powerfully executed decrements juxtaposed with a lazy random generator."

This work is licensed under a Creative Commons License.



<http://creativecommons.org/licenses/by/2.5/>

Pall Thayer

2005

<http://www.this.is/pallit>