MicroImage

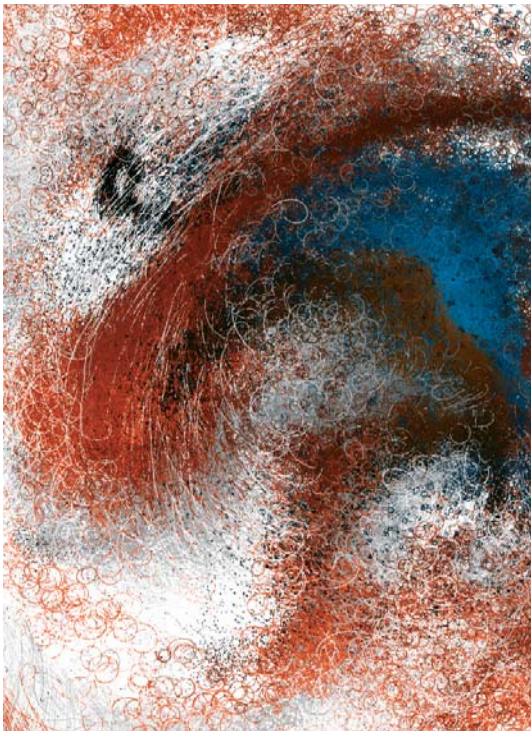
Casey Reas

MicroImage explores the phenomenon of emergence through the medium of software. It is a microworld where thousands of autonomous software *organisms* and a minimal *environment* create a software *ecosystem.* As the environment changes, the organisms aggregate and disperse according to their programmed behavior. They are tightly coupled to the environment and slight changes in the environment create macroscopic changes in the ecosystem. A field of undulating form emerges from the interactions between the environment and the organisms.

In relation to *MicroImage,* the concept of "emergence" refers to the generation of structures that are not specified or programmed. None of the structures produced through interacting with the software is predetermined or planned. Instead of consciously designing the entire structure, simple programs were written to define the interactions between the elements. Programs were written for the four different types of organism and each was cloned in the thousands. Structure emerges from the discreet movements of each organism as it modifies its position in relation to the environment. The structures generated through this process cannot be anticipated and evolve through continual iterations involving alterations to the programs and exploring the changes through interacting with the software. My understanding of emergence was informed by the publications of scientists and journalists including John Holland, Mitchell Resnick, and Kevin Kelly.

MicroImage, like all of my software explorations, has no inherent representation. The core of the project is a responsive structure without visual or spatial form. This structure is continually modified and manifests itself in diverse media and representations. *MicroImage* began as a series of responsive software for desktop computers. It later merged into a series of still images that were recorded during the process of interacting with the software. Enhanced density and physical presence were explored through these vector images. More recently, the software's movements were choreographed and recorded as a collection of short animations. It is currently manifested as a non-interactive triptych displaying the software as a live autonomous system. My preferred patterns of interaction have been encoded into a series of algorithms that control the properties of the organisms' environment. The environment responds to the positions of the organisms and the organisms respond to these changes in the environment. This method explores a balance between dynamic, generative software and controlled authorship.

The formal qualities of *MicroImage* were selected to enable the dynamic structure to be highly visible. Each organism consists of two text files written in the C++ programming language. These files, "micro.cpp" and "micro.h" are respectively 265 and 48 lines long. The files specify the behavior of each organism by defining the rules for how it responds to its simulated environment. After making a range of form explorations, each organism was given the most minimal visual form possible on a computer screen—a pixel. To differentiate the various categories of organisms, each type was assigned a distinct color. *Aggressive* organisms were assigned warm colors and *passive* organisms were assigned cool colors. As a further refinement, the values of the colors were modified to change in relation to the speed of the organism. When the organism is moving at its maximum



speed it is represented with its pure hue, but as it slows down the hue changes along a gradient until it reaches black. I soon realized that representing the organisms with a single pixel placed too much emphasis on their location and not their quality of movement. In the current software, the representation was changed to an extended pixel—a line. Each organism is displayed as a line connecting its current position and its previous twenty positions. Through this visualization, the movement of each organism is seen in both static images and kinematic representations. The linear notation allows the viewer to discern the past and present motion of the organism. The future movement may be imagined through following the degree of curvature in the line.

The core of the *MicroImage* software was written in one day over two years ago. The current version of the software has developed through a gradual evolution. While the base algorithm controlling the movement was constructed in a rational way, subsequent developments were the result of aesthetic judgments constructed through many months of interacting with the software. Through directly manipulating the code, I was able to develop hundreds of quick iterations and make decisions based on analyzing the responsive structures created by the code. This process was more similar to intuitive sketching than rational calculation.

Additional information, images, and video clips for *MicroImage* are available at *http://www.groupc.net*

Terminology

In the description of *MicroImage,* the choice of the words "organism," "ecosystem," "environment," and "clone" are used abstractly. The word "organism" can be replaced by "system" or "machine." The use of terminology grounded in biology is used to emphasize the relation between the synthetic software structures used in the project and biological structures found in the natural world.