←

REAS.com is a database for Casey REAS

- **REAS**
- INDEX
- TEXT
- INFORMATION

REAS

- INDEX,
- TEXT,
- INFORMATION

Process Compendium,
Casey Reas (2004 - present)
Updated 20 August 2010

//

An Element is a simple machine that is comprised of a Form and one or more Behaviors. A Process defines an environment for Elements and determines how the relationships between the Elements are visualized. For instance, Element 1 takes the form of a circle and one of its behaviors moves it along a straight line at a constant speed. Process 4 fills a surface with Element 1 and draws a line between elements while they overlap. Each Process is a short text that defines a space to explore through multiple interpretations. A Process interpretation in software is a kinetic drawing machine with a beginning but no defined end. It proceeds one step at a time, and at each discrete step, every Element modifies itself according to its behaviors. The corresponding visual forms emerge as the Elements change; each adjustment adds to the previously drawn shapes. During the last seven years, I have continuously refined the system of Forms, Behaviors, Elements, and Processes. The phenomenon of emergence is the core of the exploration and each artwork builds on previous works and informs the next. The system is idiosyncratic and pseudo-scientific, containing references ranging from the history of mathematics to the generation of artificial life. The Process Compendium 2004-2010 consists of two editions of fifteen prints that document Process 4 to Process 18. In each case, two static images have been selected from an infinite number of potential variations. This book documents these prints, alternate software interpretations, and the emergence of the forms through a series of time-lapse images.

//

The most important element of Process <NUMBER> is the text. The text is the Process described in English, written with the intent to translate its content into software. The Elements, Forms, and Behaviors referenced within the Process text are defined in the Library. The software interpretation is secondary to the text. The text leaves many decisions open to the programmer, decisions that must be made using personal judgment. The act of translating the Process from English into a machine language interprets the text. Process <NUMBER> was implemented by <NAME> into the <LANGUAGE> language in <YEAR>. The hardware is inconsequential and in time it will fail. It was selected to be robust, but electronic devices are fragile. If a component of the hardware fails, it may be replaced without diminishing the work. Eventually, compatible components won't be available. When this happens, a new hardware system must be acquired and the software modified for the new platform.

//

The Elements, Forms, and Behaviors referenced within the Processes are defined in the Library:

Forms
F1: Circle
F2: Line

Behaviors
B1: Move in a straight line
B2: Constrain to surface
B3: Change direction while touching another Element
B4: Move away from an overlapping Element
B5: Enter from the opposite edge after moving off the surface
B6: Orient toward the direction of an Element that is touching
B7: Deviate from the current direction

Elements
E1: F1 + B1 + B2 + B3 + B4
E2: F1 + B1 + B5
E3: F2 + B1 + B3 + B5
E4: F1 + B1 + B2 + B3
E5: F2 + B1 + B5 + B6 + B7

//

Process 18

A rectangular surface filled with instances of Element 5, each with a different size and gray value. Draw a quadrilateral connecting the endpoints of each pair of Elements that are touching. Increase the opacity of the quadrilateral while the Elements are touching and decrease while they are not.

Implemented by Casey Reas
Summer 2007, Winter 2008
Processing 125 / 135

Process 17

A rectangular surface filled with instances of Element 5, each with a different size and gray value. Draw a transparent circle at the midpoint of each Element. Increase a circle?s size and opacity while its Element is touching another Element and decrease while it is not.

Implemented by Casey Reas
Summer 2010
Processing 1.1

Process 16

A rectangular surface filled with instances of Element 3, each with a different size and gray value. Draw a tiny, transparent circle at the midpoint of each Element. Increase a circle?s size and opacity while its Element is touching another Element and decrease while it is not.

Implemented by Casey Reas
Fall 2006
Processing 121

Process 15

A rectangular surface filled with instances of Element 3, each with a different size and gray value. Draw a small, transparent circle at the midpoint of each Element. Increase the circle?s opacity while its Element is touching another Element and decrease while it is not.

Implemented by Casey Reas
Fall 2006
Processing 121

Process 14

A rectangular surface filled with instances of Element 4, each with a different size and direction. Display the intersections by drawing a circle at each point of contact. Set the size of each circle relative to the distance between the centers of the overlapping Elements. Draw the smallest possible circle as white and the largest as black, with varying grays representing sizes in between.

Implemented by Casey Reas
Fall 2006, Winter 2008
Processing 122, 135

Process 13

Bisect a rectangular surface and define the dividing line as the origin for a large group of Element 1. When each Element moves beyond the surface, move its position back to the origin. Draw a line from the centers of Elements that are touching. Set the value of the shortest possible line to black and the longest to white, with varying grays representing values in between.

Implemented by Casey Reas
Fall 2009, Summer 2010
Processing 1.0

Process 12

A rectangular surface filled with instances of Element 3, each with a different size and gray value. Draw a dot at the center of each line. Draw a quadrilateral connecting the endpoints of each pair of Elements that are touching. Increase the opacity of the dot and quadrilateral while the Elements are touching and decrease while they are not.

Implemented by Casey Reas
Fall 2006

Processing 121

Process 11

A rectangular surface filled with instances of Element 2, each with a different size, speed, and direction. Display the intersections by drawing a circle at each point of contact. Set the size of one circle to be relative to the distance between the centers of the overlapping Elements and make the other circle tiny. Draw the smallest possible circle as white and largest as black, with varying grays representing sizes in between.

Implented by Casey Reas
Summer, Fall 2006
Processing 115

Process 10

Position a circle at the center of a rectangular surface. Set the center of the circle as the origin for a large group of Element 1. When an Element moves beyond the edge of its circle, return to the origin. Draw a line from the centers of Elements that are touching. Set the value of the shortest possible line to black and the longest to white, with varying grays representing values in between.

Implemented by Casey Reas
Fall 2005
C++/OpenGL, Processing 115

Process 9

Position three large circles on a rectangular surface. Set the center of each circle as the origin for a large group of Element 2. When an Element moves beyond the edge of its circle, return to the origin. Display the intersections by drawing a circle at each point of contact. Set the size of each circle relative to the distance between the centers of the overlapping Elements. Draw the smallest possible circle as white and largest as black, with varying grays representing sizes in between.

Implemented by Casey Reas
Fall 2005
C++/OpenGL, Processing 115

Process 8

A rectangular surface densely filled with instances of Element 2, each with a different size, speed, and direction. Display the intersections by drawing a circle at each point of contact. Set the size of each circle relative to the distance between the centers of the overlapping Elements. Draw the smallest possible circle as black and largest as white, with varying grays representing sizes in between.

Implemented by Casey Reas
Summer 2005
C++/OpenGL, Processing 115

Process 7

A rectangular surface filled with varying sizes of Element 1. Draw a line from the centers of Elements that are touching. Set the value of the shortest possible line to white and the longest to black, with varying grays representing values in between. Draw the perimeter of each Element as a black line and the center as a white dot.

Implemented by Casey Reas
Summer 2005
C++/OpenGL, Processing 115

Process 6

Position three large circles on a rectangular surface. Set the center of each circle as the origin for a large group of Element 1. When an Element moves beyond the edge of its circle, return to the origin. Draw a line from the centers of Elements that are touching. Set the value of the shortest possible line to black and the longest to white, with varying grays representing values in between.

Implemented by Casey Reas
Winter 2005
C++/OpenGL, Processing 115

Process 5

A rectangular surface filled with varying sizes of Element 1. Draw the perimeter of each Element as a black line and the center as a white dot. Draw a gray line from the centers of Elements that are touching.

Implemented by Casey Reas
Winter 2005
C++/OpenGL, Processing 115

Process 4

A rectangular surface filled with varying sizes of Element 1. Draw a line from the centers of Elements that are touching. Set the value of the shortest possible line to black and the longest to white, with varying grays representing values in between.

Implemented by Casey Reas
Winter 2005
C++/OpenGL, Processing 115

Process 3, 2, 1

The first three Process works are named Structure 1, 2, and 3. They were created for the {Software} Structures project, commissioned by the Whitney Museum of American Art's ARTPORT in 2004. Although the Process works originated from the Structure works, they are separate projects. There is more information at the URL: http://artport.whitney.org