

# generative art

a practical guide  
using processing



  
MANNING

matt pearson  
foreword by marius watz

---

## Numbers

**3D printer, and tangible generative art** 11  
**100 Abandoned Artworks (Matt Pearson)** xxx  
**01010101, Drawing With Particles** xxxix

---

## Symbols

**- operator** 32  
**-- operator** 32  
**-= operator** 32  
**\* operator** 32  
**\*= operator** 32  
**/ operator** 32  
**/= operator** 32  
**% operator** 32  
**+ operator** 32  
**++ operator** 32  
**+= operator** 32

---

## A

**absolute coordinate, definition of** 23  
**abstraction as art** 73  
**add assign operator** 32  
**addition operator** 32  
**Addition/Subtraction, Variant (Robert Hodgkin)** xx  
**Adobe Illustrator, usability** 11  
**Adobe Photoshop, frequent new releases of** 10  
**After Effects, usability** 11  
**agent** 146

- drawing with the ellipse function 68
- human 148
  - and generative art 151
  - challenges of using 152
  - collecting data 152
- humanizing 148
- querying 147

- real-world attributes 147
- turning into a spiral 76
- vs. object 127

**algorithm, definition of** 8  
**algorithm, as real tool underlying generative art** 9  
**Ali, Reza, LORMALIZED** xxxix  
**alpha value, introduction to** 25  
**Android** 46  
**animation**

- basics of 27–33
- noisy 89–91

**ant colony**

- as example of emergence 108
- lack of central organization 108

**Apollo** 17 148  
**append function** 119  
**Apple, annual technology updates** 10  
**array**

- in programming 119
- multidimensional 133

**artificial intelligence, and generative art** 136  
**autonomy**

- definition of 127
- required for generative methodology 6

---

## B

**Beddard, Tom** viii  
**Mandelbulb** 156  
**beginShape function** 75  
**Blake, William, Newton** xxv  
**Blonde Boids (Abandoned Artwork #4)** 109  
**Boids algorithm, realistic flocking simulation** 109  
**boolean** 24  
**break** 39  
**Brian's Brain** 138–141  
**Broken Mirrors 1-4 (Matt Pearson)** xxxviii  
**Brown, Paul, and early generative art** 9  
**byte** 24

---

---

## C

**Cage, John, and generative music** 7  
**Cannon, J. W., finite subdivision rules** 171  
**case study, *Wave Clock*** 76–82  
**cellular automata** 128–144  
    Brian’s Brain *SEE Brian’s Brain*  
    continuous behavior 143  
    definition of 128  
    Game of Life *SEE Game of Life*  
    setting up 130–133  
    Vichniac Vote *SEE Vichniac Vote*  
    waves (averaging) *SEE waves (averaging)*  
**chaos and order** xxv–xxxii  
**char** 24  
**circle**  
    calculating points on circumference of 67  
    drawing  
        with OOP 113  
        with the ellipse function 67–68  
        with trigonometry 67  
    turning into a spiral 69–76  
**class**  
    in programming 114  
    vs. instance 115  
**class keyword** 114  
**Clojure, Processing implementation** 14  
**collaboration, generative artwork as** 6  
**collision correction** 120  
**collision detection** 120–125  
***Colour Economy*, Jeremy Thorp** 149  
**comment, notations** 20  
**comparison operators** 33  
**complexity, and emergence** 123  
**Complexity theory, and emergence** 108  
**Computer Arts Society (CAS)** 170  
**conditional**  
    grouping 33  
    introduction to 33–34  
    nesting 34

**Context Free** 14  
**Conway, John, Game of Life** *SEE Game of Life*  
**cos function** 60  
**cosine curve, drawing** 61  
**cosine, definition of** 67  
**CSS, usability** 11  
**curveVertex(x,y) function** 75  
**C++, usability** 11  
**custom function** 30–31

---

## D

**data type**  
    boolean 24  
    byte 24  
    char 24  
    composite 24  
    float 24  
    int 24  
    most common 24  
    String 24  
**data visualization** 153  
**Dawkins, Richard, *The Blind Watchmaker*** 4  
**decrement operator** 32  
**digital artwork, fragility of** 11  
**digital tools** 9–12  
    learning curve 11  
    rapid obsolescence of 9, 11  
**dist function** 120  
**divide assign operator** 32  
**division operator** 32  
**draw() function block** 27  
**drawing**  
    basics of 19–27  
    in 3D space 93–95  
    rotational *SEE rotational drawing*  
***Drawing With Particles* (01010101)** xxxix  
**Dublin Trinity College, School of  
    Computer Science** 52

---

## E

**efficiency, in code, importance of** 23

**ellipse function** 18

drawing a circle with 67–68

**emergence** 108–111

and complexity 125

and Complexity theory 108

and consciousness 111

ant colony as example of 108

as a way of generating organic artworks 111

as organizing principle 111

definition of 108

patterns in interactions 125

political connotations of 111

sociological perspective 110–111

**Emergence (John Holland)** 108

**Emergence (Steven Johnson)** 108

**emergent complexity** 125, 128–144

**emergent networks** 110

**endShape function** 75

**Enlightenment** xxv

**Eno, Brian**

and generative music 7

*Discreet Music* 7

---

## F

**fill**

introduction to 25

turning off 27

**fill function** 75

**Final Cut Pro, usability** 11

**Flash, usability** 11

**flicker-fusion** 45

**Flight Patterns (Aaron Koblin)** xxxiv, 150

**float** 24

**flocking simulation, rule for** 109

**Floyd, W. J., finite subdivision rules** 171

**for loop** 39–40, 54

**fractal**

and infinite recursion **SEE** *infinite recursion*

case study, Sutcliffe Pentagons 170–187

angle step, varying 184

animating 181

constructing 171–180

description of 171

example 172

experimenting with 180–187

incomplete revolution 184

number of sides, varying 181

strut length, varying 180

exponential growth 165–170

introduction to 157

self-similarity, coding **SEE** *self-similarity, coding*

space clockwork 165, 168

tree

animating 163

beginnings of 162

**frame loop** 27–30, 89

and noise 89

**frameRate function** 27

**frame rate, standards** 45

**Frosti (Matt Pearson)** xxvii, 102

**Fry, Benjamin** 14

Visualizing Data 153

**function**

custom **SEE** *custom function*

introduction to 18

**function block**

definition of 27

---

## G

**Galanter, Philip, definition of generative art** 3

**Game of Life** 128, 134–137

and generative art 137

code 134

online examples 136

rules 134

**GarageBand, usability** 11

**generative art**

aim of xxxii

algorithms as tools to produce 9

and artificial intelligence 136  
and human beings 151  
and the Game of Life 137  
as chaos art xl  
as collaboration 6  
history of 6–9  
introduction to xviii  
order and chaos xxv–xxxii  
simplicity of xxii–xxiii  
tangible, via 3D printer 11  
what it is 12  
what it isn't 4–6

**generative methodology**  
autonomy of 6  
rules of defining 6  
unpredictability of 6

**generative music 7**

**Gibson, William 57**

**global variable**  
definition of 29  
underline as preface for 29

**Gödel, Escher, Bach (Douglas Hofstadter) 136**

**Greenberg, Ira, Processing tutorial 17**

**Grid Distortion 02D 0018 (Marius Watz) xxxii**

---

## H

**halting problem, of Alan Turing 158**

**Happy Place (Jared Tarbell) xl**

**HasCanvas 47**

**Haunted Fishtank 99**

**Hello World 17–19**

**hexadecimal, using to denote color values 22**

**Hodgin, Robert**  
*Addition/Subtraction, Variant* xx  
*Jelly (Magnetosphere)* xxi  
*Magnetic Ink* xxi

**Hofstadter, Douglas, Gödel, Escher, Bach 136**

**Holland, John, Emergence 108**

**HTML, usability 11**

**human beings, and generative systems 151**

**hypotenuse, definition of 67**

---

**if 33**

**if ... else 33**

***Illuminations B (Marius Watz) xxxiii***

**Illustrator, usability 11**

**increment operator 32**

**infinite loop 35, 158**

**infinite recursion 158**  
simulating in code 158

**infinity, computer's inability to process 158**

**Information Is Beautiful 153**

**instance**

creating 115  
in programming 114  
self-awareness 120  
vs. class 115

**int 24**

**interaction between objects, and complexity 125**

**internet, as an example of emergence 110**

**Introduction to Circle Packing  
(Kenneth Stephenson) 171**

**iPad, developing for 46**

**iPhone, developing for 46**

**iProcessing 47**

**iterative variance 54–56**  
vs. randomness 56

---

## J

**Java**

must be installed in browser 42  
usability 11

**JAVA2D rendering engine 93**

**Java Micro Edition (Java ME) 46**

***Jelly (Magnetosphere) (Robert Hodgin) xxi***

**Jerome Saint-Clair SEE 01010101**

**Johnson, Steven, Emergence 108**

---

## K

**KBGD01E 0012 (Marius Watz)** xxxii  
**keyPressed()** function 43  
**Koblin, Aaron, *Flight Patterns*** xxxiv, 151

---

## L

**Lady Quark (Manfred Mohr)** 8  
**Lee-Delisle, Seb, *Lunar Landing Trails*** 152  
**Lewes, George Henry, and emergence** 108  
***Life in 2050* (Matt Pearson)** xxvi  
**line function, vs. translate function** 88  
**loading.gif** 44  
**local variable, definition of** 29  
**logic operators** 33  
**loop**  
    for *SEE for loop*  
    terminating 39  
    while *SEE while loop*  
**looping, introduction to** 34–40  
**LORMALIZED (Reza Ali)** xxxix  
**Lunar Lander arcade game** 152  
***Lunar Lander Trails* (Seb Lee-Delisle)** 152

---

## M

***Magnetic Ink* (Robert Hodgkin)** xxi  
**Mandelbrot** 156  
***Mandelbulb* (Tom Beddard)** 156  
**mathematical transformations, and line drawing** 63  
**Maya, usability** 11  
**methodology, generative** *SEE generative methodology*  
**micro-behavior vs. macro-knowledge** 111  
**mobile phone, developing for** 46  
**Mobile Processing** 46  
**MODE, specifying rendering engine with** 92  
**modulo operator** 32

## Mohr, Manfred

    and early generative art 9  
    *Artificiata I* 9  
    *Lady Quark* 8

**Molnar, Vera, and early generative art** 9

***Mondrian Architecture* (Matt Pearson)** xxxvi

**Moore's Law** 11

**morphogenesis** 110

**Mozart, as generative musician** 7

**Mozilla, Processing for the Web** 14

**multiplication operator** 32

**multiply assign operator** 32

**music, generative** *SEE generative music*

---

## N

**Nake, Frieder, and early generative art** 9

**naturalistic variance** 57–63, 73–76

**Nees, George, and early generative art** 9

**negation operator** 32

***Neuromancer*** 57

**new keyword** 115

**Newton, Isaac, world view of** xxv

***Newton* (William Blake)** xxv

***New York Times*, API** 153

**NodeBox** 14

**noFill function** 27

**noise**

    and animation 89–92  
    generating 60, 73–76  
    three-dimensional 95–98  
    two-dimensional 84–89  
    visualizations 85–89

**noise cube** 97

**noise function** 57

    as variance in alpha values 85  
    custom 73–76

**noise grid** 84–85

    on frame loop 89  
    with noise variance 91  
    written as function blocks 85

---

## O

### object

- in Processing 114
- modeling in programming 114
- vs. agent 127

### object-oriented programming 112–125

- array 119
- class 114
- classes and instances 113–120
- drawing circles with 113
- instance 114
- interactions between behaviors 125
- introduction to 112
- objects 114

### obsolescence, rapid, of digital tools 10

### openFrameworks 14

### OpenGL library, importing 94

### OPENGL rendering engine 93

### operators

- grouping 31
- introduction to 31–32
- list of 32

### *Opiamas Trangelo* (Matt Pearson) xix

### *Orbitals* (Matt Pearson) xxxvii

### order and chaos xxv–xxxii

### order of execution

- and layering 25
- and styles 25

### organic, creating using the mechanical xviii

### O'Rourke, Robert, HasCanvas 47

### output window 18

---

## P

### parameter, introduction to 20

### Parry, W. R., finite subdivision rules 171

### PDF rendering engine 93

### Pearson, Matt

- 100 Abandoned Artworks* xxx
- Broken Mirrors 1–4* xxxviii
- Frosti* xxvii, 102
- Life in 2050* xxvi

*Mondrian Architecture* xxxvi

*Opiamas Trangelo* xix

*Orbitals* xxxvii

Perth Arts Festival Branding xxiv

*Spiral Stairs* 72

*Tube Clock* 5

*Twill* xxviii

*Wave Clock* 77, 81

### Perlin, Ken 57

### Perlin noise 57

- 2D, from a 3D perspective 96
- case study 76–82
- in Processing 57
- visualized as alpha values 85
- visualized as clouds 89
- visualized as squares of varying sizes 87
- visualized using rotation 86

### Perth Arts Festival Branding (Matt Pearson) xxiv

### Photoshop, usability 11

### popMatrix function 88

### Process 14 / Image 4 (Casey Reas) xxix

### Processing

- as tool to create generative art 6
- introduction to 14–19
- learning curve 15
- Mobile *SEE Mobile Processing*
- Perlin noise in *SEE Perlin noise*
- resources 15
- state-based 25
- tutorials 17
- website 15

### Processing 2D (P2D) rendering engine 93

### Processing 3D (P3D) rendering engine 93

### Processing for Android 46

### Processing for the Web 14

### Processing.js 14

- and iPhone 47
- downloading 47

### programming

- as poetry xxxv–xxxviii
- introduction to 19–40
- syntax rules xxxviii

---

## programming language

- as means of producing 2D generative art 6
- as tool to create generative art 6

**pseudo-randomness** 52

**pushMatrix** function 88

---

## R

**radians vs. degrees** 67

**random function** 52–56

- custom 63
- drawing a line with 53

## randomness

- and not-so-randomness 52–54
- and Perlin noise 57
- computer-generated 52
- generating in a controlled fashion 57
- random.org project 52
- vs. random variance 56

**random number, generating** 52

**random.org project** 52

**Reas, Casey** 14

*Process 14 / Image 4* xxix

**recursion, infinite** SEE *infinite recursion*

**relative coordinate, definition of** 23

## rendering engine

- JAVA2D 93
- list of 93
- OPENGL 93
- P2D 93
- P3D 93
- PDF 93
- specifying with MODE 92

**Resig, John** 14

**Reynolds, Craig, Boids algorithm** 109

**RGB, introduction to** 20

**Romanticism** xxv

**rotational drawing** 66–76

**rotation, performing with translate function** 88

**Ruby, Processing implementation** 14

**Rucker, Rudy** 105

---

## S

**Saint-Clair, Jerome** SEE *01010101*

**Save As** 42

**saveFrame** function 43

**saving work** 41

**Scala, Processing implementation** 14

**screen dimension** 45

## script

- exporting 41
- generating application from 42
- saving 41

## self-similarity

- coding 159–170
- exponential growth 165–170
- introduction to 157
- recursive depth, limiting 159
- tree structure 159
  - animating 163
  - branches 161
  - level and index 162

**semicolon** 19

**setup() function block** 27

**Shiffman, Dan, Processing tutorial** 17

Boids implementation 109

**simulation and visualization** 146

**sine curve, drawing** 60

**sine, definition of** 67

**sin function** 60

**size function, MODE parameter** 92

**sketch, definition of** 41

**smoothing, introduction to** 20

**speed, as code limitation** 23

## sphere

- drawing with sphere function 93
- drawing with trigonometry 100–103
- spiral drawn around 101

**sphereDetail** function 94

**sphere** function 93

## spiral

- adding noise to 69–76
- drawing 69–76

**Spiral Stairs (Matt Pearson)** 72

---

starlings, murmating 109  
**Stephenson, Kenneth, Introduction to  
Circle Packing** 171  
still  
  creating 43  
  using as \alt\ image 44  
  using to create video 44  
**Stockhausen, Karlheinz, and generative music** 7  
**Strange Symmetry 2 (Frederik Vanhoutte)** xxxv  
**String** 24  
stroke  
  introduction to 22  
  turning off 27  
**stroke function** 25  
**strokeWeight function** 25  
**Structure Synth** 14  
**subtract assign operator** 32  
**subtraction operator** 32  
**Sutcliffe, Alan**  
  Sutcliffe Pentagons 170  
  *White Heat Cold Logic* 170  
syntax, importance of 19

---

## T

**Tarbell, Jared, *Happy Place*** xl  
**Thorp, Jeremy**  
  *Colour Economy* 149  
  data visualization 153  
  use of *New York Times* API 153  
tools, digital *SEE digital tools*  
**Toop, David, *Haunted Weather*** 7  
**translate function** 88  
  and drawing in 3D 94  
  vs. line function 88  
**TRON** 57  
**Tube Clock (Matt Pearson)** 5  
**Turing, Alan**  
  morphogenesis 110  
  halting problem 158  
**Twill (Matt Pearson)** xxviii  
**Twitter, and emergence** 110

## U

underline, as preface for global variables 29  
unpredictability, required for generative  
  methodology 6

---

## V

**Vanhoutte, Frederik, *Strange Symmetry 2*** xxxv  
variable  
  global *SEE global variable*  
  introduction to 23–24  
  local *SEE local variable*  
variable scope, definition of 29  
variance  
  iterative *SEE iterative variance*  
  naturalistic *SEE naturalistic variance*  
version control 42–43  
**Vichniac, Gerard** 137  
**Vichniac Vote** 137–138  
video  
  creating 44–45  
  dedicated machine for rendering 44  
  frame rates 45  
  screen dimensions 45  
**Vimeo, screen dimensions** 45  
**Visual Complexity** 153  
**Visualizing Data (Ben Fry)** 153  
vvvv 14

---

## W

**Watts, Alan** xviii  
**Watz, Marius**  
  *Grid Distortion 02D 0018* xxxii  
  *Illuminations B* xxxiii  
  *KBGD01E 0012* xxix  
**Wave Clock, case study** 76–82  
waves 144  
while loop 35–36

**White Heat Cold Logic (Alan Sutcliffe)** 170

**wireframe, definition of** 45

**wireless toolkit (WTK)** 46

---

X

**xnoise** 92

**xstartNoise** 92

---

Y

**ynoise** 92

**YouTube, screen dimensions** 45

**ystartNoise** 92

---

Z

**z value** 93

**ZX Spectrum** 10

---

# generative art

pearson

Artists have always explored new media, and computer-based artists are no exception. Generative art, a technique where the artist creates print or onscreen images by using computer algorithms, finds the artistic intersection of programming, computer graphics, and individual expression.

**Generative Art** presents both the techniques and the beauty of algorithmic art. In it, you'll find dozens of high-quality examples of generative art, along with the specific steps the author followed to create each unique piece using the Processing programming language. The book includes concise tutorials for each of the technical components required to create the book's images, and it offers countless suggestions for how you can combine and reuse the various techniques to create your own works.

## What's inside

- The principles of algorithmic art
- A Processing language tutorial
- Using organic, pseudo-random, emergent, and fractal processes

**Matt Pearson** is an artist, coder, and award-winning blogger.

FREE  
eBook

SEE INSERT

For access to the book's forum and a free ebook for owners of this book, go to [manning.com/GenerativeArt](http://manning.com/GenerativeArt)

“Perfectly placed at the intersection of code and creative thinking.”

From the Foreword by  
**Marius Watz**  
Founder of Generator.x

“Succeeds in teaching without lecturing ... turns dull programming concepts into fun explorations.”

**Frederik Vanhoutte**  
Generative Artist, wblut.com

“If you're interested in generative art, this is the place to start.”

**Robert O'Rourke**  
Founder of HasCanvas

US \$39.99 / Can \$45.99 (including eBook)

ISBN-13: 978-1-935182-62-7  
ISBN-10: 1-935182-62-5

