



Was ist Processing?

Processing ist eine Open Source-Programmiersprache für die Programmierung von Bildern, Animation und Sound. Es wurde speziell für Studenten, Künstler und Designer entwickelt. Durch den einfachen Aufbau eignet sich Processing gut als Einstieg in die Grundstrukturen des Programmierens. Processing wurde von Künstlern und Designern entwickelt, um eine Alternative zu kommerzieller Software zu bieten. Processing wurde von Ben Fry und Casey Reas lanciert.

Version

Das vorliegende Skript ist aktualisiert für die Processing Version 1.1 von März 2010.

Vergleich mit Flash

Flash und Processing konkurrenzieren sich nicht direkt. Flash ist für GUI (grafische User Interface) Anwendungen die bessere Wahl. Processing ist vor allem für Anbindung externer Hardware (Arduino, Video-Input) geeignet. Für Flash/Actionscript empfehlen wir unsere Plattform <http://www.visualcode.ch>

Download

<http://www.processing.org/download/>

Literatur

Casey Reas, Ben Fry, John Maeda: *Processing: A Programming Handbook for Visual Designers and Artists*, MIT Press, 2007, ISBN-10: 0-262-18262-9, ISBN-13: 978-0-262-18262-1

Ira Greenberg, *Processing: Creative Coding and Computational Art (Foundation)*, Friends of Ed, 2007, ISBN-10: 159059617X, ISBN-13: 978-1590596173

Daniel Shiffman, *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, Morgan Kaufmann, 2008, ISBN-10: 0123736021, ISBN-13: 978-0123736024

Hartmut Bohnacker, Benedikt Groß, Julia Laub: *Generative Gestaltung: Entwerfen. Programmieren. Visualisieren.*, Schmidt Hermann Verlag, 2009, ISBN-10: 3874397599, ISBN-13: 978-3874397599

Beispiele

Zeichnen und Klicken

Custom Tool

<http://www.processing.org/learning/topics/customtool.html>

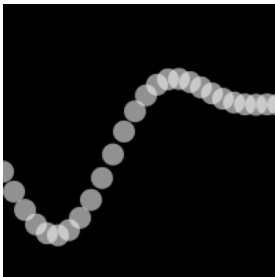
Zeichnen durch Mausbewegung auf dem Bildschirm. Processing kann mehr, als Stift und Pinsel zu imitieren, es können neuartige Zeichenwerkzeuge geschaffen werden.

Pattern

<http://www.processing.org/learning/topics/pattern.html>

Dieses Zeichenwerkzeug reagiert auf die Mausgeschwindigkeit.

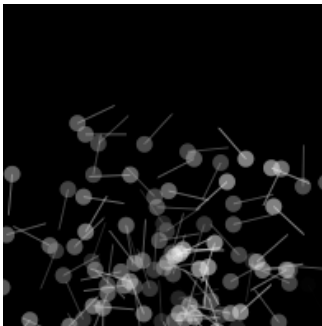
Mathematik



Wellen-Funktion

<http://www.processing.org/learning/basics/additivewave.html>

Simulation



Particles

<http://www.processing.org/learning/topics/multipleparticlesystems.html>

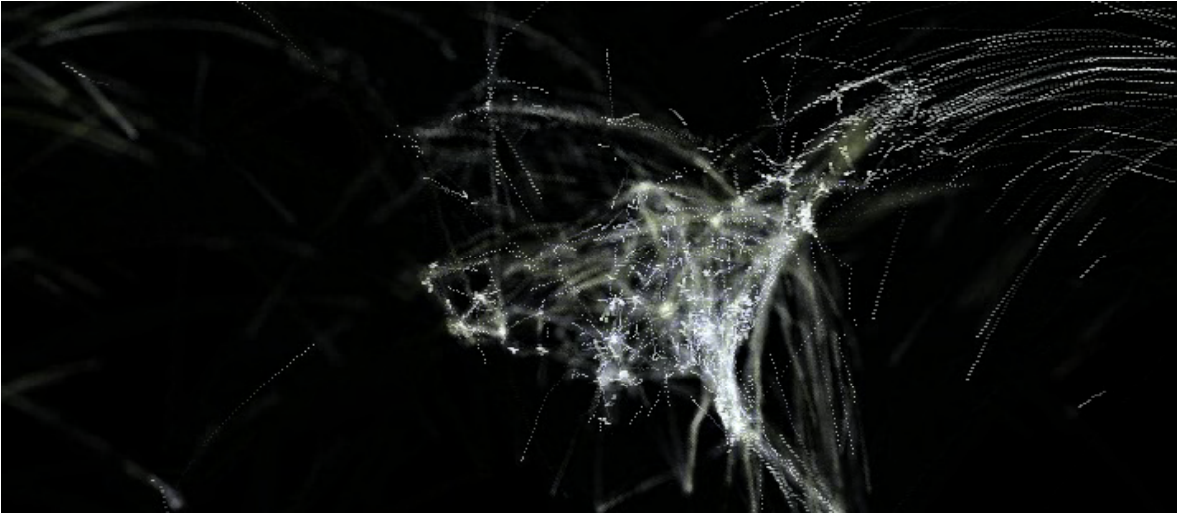
Ein Partikelschwarm, der vom Mausklick ausgelöst wird.

Cellular Automata

<http://www.processing.org/learning/topics/conway.html>

Eine einfache Version von Conway's Game of Life. Ein Punkt wird schwarz, wenn er weniger als zwei, oder mehr als 3 Nachbarn hat. Ein schwarzer Punkt wird aktiviert, wenn er genau drei Nachbarn hat.

Anwendungen und Programme



Flight patterns

<http://www.aaronkoblin.com/work/faa/>

Shadowmonsters

<http://www.worthersoriginal.com/wiki/#page=shadowmonsters>

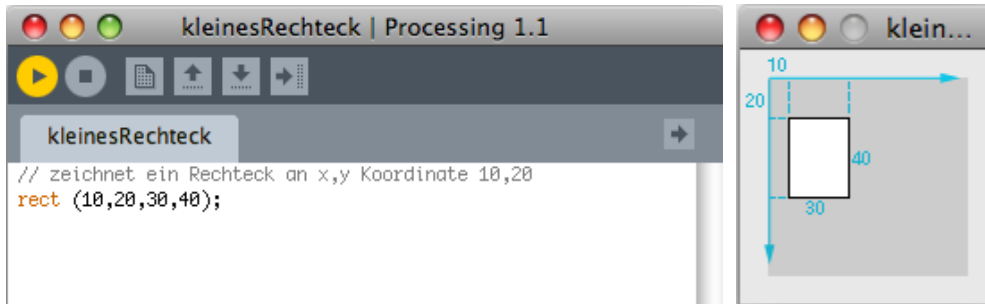
Khronos Projector

http://www.k2.t.u-tokyo.ac.jp/members/alvaro/Khronos/Khronos_P5/Khronos_Applets.htm

Grundaufbau Processing

Umgebung

Die Processing-Umgebung besteht nur aus einem textbasierten Code-Editor. Die wichtigsten Funktionen Run, Stop, New, Open, Save und Export sind auch gleich als Icons im Titelbalken vorhanden.



Beim Klicken des Symbols "Run" wird der Programmcode kompiliert und in einem neuen Fenster ausgeführt (rechts). Die Koordinaten werden von der linken oberen Ecke aus gemessen. Gemäss der üblichen Konvention wird die horizontale x-Koordinate (10 Pixel von links) zuerst geschrieben, dann folgt die vertikale y-Koordinate (20 Pixel von oben). Ähnlich wird beim hier gezeichneten Rechteck zuerst die (horizontale) Breite 30 Pixel angegeben, und dann die (vertikale) Höhe 40 Pixel.

Sketchbooks

Die Programme sind als Textdateien in sogenannten Sketchbooks abgelegt (Dateien mit der Endung .pde, müssen in einem gleichnamigen Ordner liegen). Mit "Run" werden sie gestartet, und mit Export in Form eines Java-Applets exportiert. Dieses kann in den meisten gängigen Browsern betrachtet werden.

Beispiele und Hilfe

Unter Examples können vordefinierte Programme zu den meisten Standardfunktionen geladen werden. Programmdokumentation gibt's unter <http://www.processing.org/reference/index.html>

Code-Referenz

Spezialzeichen

<code>//</code>	Kommentare
<code>;</code>	schliesst eine Code-Zeile ab, muss am Ende jeder Zeile stehen
<code>{}</code>	Begrenzung eines Anweisungsblocks, z.B. einer Prozedur

Initialisierung

<code>size(w,h);</code>	Bühnengrösse, w: Breite, h: Höhe
<code>background(r,g,b);</code>	Hintergrundfarbe in rot, grün, blau – Werten
<code>smooth();</code>	alle Formen werden anti-aliased dargestellt

Zeichnen

<code>point(x,y)</code>	setzt einen Pixel
<code>line(x,y,z,w);</code>	Linie von x,y nach z,w
<code>rect(x, y, w, h);</code>	Rechteck an der Stelle x,y, mit Dimensionen w, h
<code>ellipse(x, y, w, h);</code>	Ellipse an Stelle x,y mit Dimensionen w, h
<code>ellipseMode(CENTER);</code>	besagt, dass x und y der Ellipse von der Mitte aus definiert sind
<code>triangle(a,b,c,d,e,f);</code>	Dreieck

Eigenschaften

<code>stroke(r,g,b);</code>	Linienfarbe
<code>noStroke();</code>	Formen sollen ohne Linien gezeichnet werden
<code>fill(r,g,b);</code>	Füllfarbe
<code>noFill();</code>	Formen sollen ohne Füllung gezeichnet werden
<code>StrokeWeight();</code>	Liniendicke

Strukturen

<code>{}</code>	Begrenzung eines Anweisungsblocks, z.B. einer Prozedur
<code>void setup() {</code> <code>}</code>	Initialisierungs-Prozedur
<code>void draw() {</code> <code>}</code>	zur Laufzeit immer wiederholt

Maus-Interaktion

<code>mouseX, mouseY</code>	Maus-Position, horizontal und vertikal
<code>pmouseX, pmouseY</code>	letzte Mausposition, horizontal und vertikal

Mathematische Funktion

<code>random(n)</code>	liefert eine zufällige Zahl zwischen 0 und n
------------------------	--