

Processing Workshop, january 2008
Documentation for patternGenerator

by Andreas Lahti , final project for the course CAAD Theorie
led by CAAD, Faculty of Architecture, ETH Zürich

Guide with examples

The examples below shows some samples of how different patterns can be produced.

First.

Every time the applet is launched. Some random values assign parameters such as the underlying image that is part of creating the pattern as well as a color scheme. These parameters can not be changed while running the applet. If you're not happy with what you see, simply relaunch the applet.

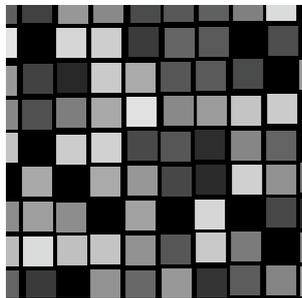
The interface.

If you move the cursor while running the applet you will activate a heads-up display, showing you the basic commands for interacting with the applet.

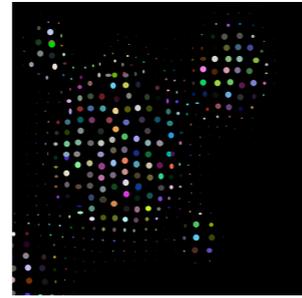
Legend – Commands

- up** (arrow): increases the biggest max size the element can take
- down** (arrow): decreases the biggest max size the element can take
- right** (arrow): increases the biggest minimum size the element can take
- left** (arrow): decreases the biggest minimum size the element can take
- g**: increases the distance the element tries to keep to each-other
- G** (shift+g): decreases the distance the element tries to keep to each-other
- d**: allow elements outside area to die (be removed)
- D**: discards the above command.
- s**: saves a time-stamped pdf-file of the screen.
- i**: displays the underlying image.
- r**: displays the elements as rectangles.
- c**: displays the elements as circles (ellipses to be more accurate).
- o**: displays the elements as outlines.
- f**: displays the elements with color fill.

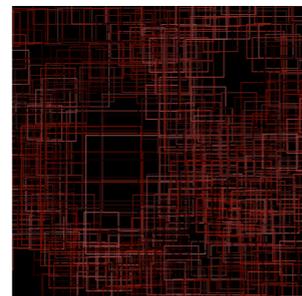
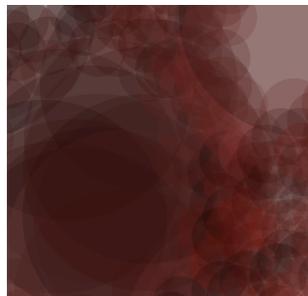
Examples:



The above example has a small difference between max and min size and therefore the main force at play is the elements ideal distance – leading to a grid-pattern as the detection algorithm assumes elements as rectangular.



Here you see the underlying image having a big influence on the pattern – as the elements now have different sizes. The gap value has also forced some elements to change their position. Drawing method here is circle (c).



The above examples uses very extreme values. First a high max size (up) number. Added to that the gap-value is negative. Allowing the elements to overlap. The end result is very much depending on the underlying picture. Also drawing method (outline/fill and rectangle/circle) makes a huge difference in how we perceive the pattern.

About patternGenerato (theoretical description)

This Applet uses mapped image-based information as a parameter for assigning element different sizes. For instance: if an element's center-point is mapped (e.g. it's x and y coordinates) to a light part of the image the element will grow. If it is darker it shrinks.

As a counter-force every element has a parameter set to avoid any other element. This parameter can be either positive or negative. This value sets the "ideal" distance the element strives to have to it's neighbors. A negative value would here allow elements to overlap.

When a neighbor element is within this distance, they will repel each other. Overlaps might still occur as other element also repels one another, a dense situation is very likely to trigger this situation. As a response to this situation each element also will try to change it's proportion slightly.

There are two types of elements: Fixed- and not-fixed- elements. Fixed objects are static and don't react on colors nor other elements. However every other element recognize a fixed-element and responds to these in the same way as any other. The conceptual idea is that a fixed object could represent a window or an opening in a facade if this tool would be used for this purpose.

Main parameters

Starting parameters:

- h-element
- v-element
- gap

These are the starting parameters for creating x number of vertical elements, x horizontal elements and their initial spacing. The objects are then created and layout in a grid

Further there are some presets for color allocation. There is also a preset of images. When starting the applet these pre-sets are chosen on a random basis.

User-interaction

While the applet runs there are several parameter that the user can change:

- max size of element
- min size of element
- gap distance each element tries to keep to neighboring element.
- fill-/outline- of element
- rect-/circle-representation of element

Changing these parameters can change the rendered pattern dramatically. Further the user can at any moment:

- export a pdf-file with a time/date-stamp.
- show mapped image, making the pattern a little more understandable.
- display a Legend for user-interaction.