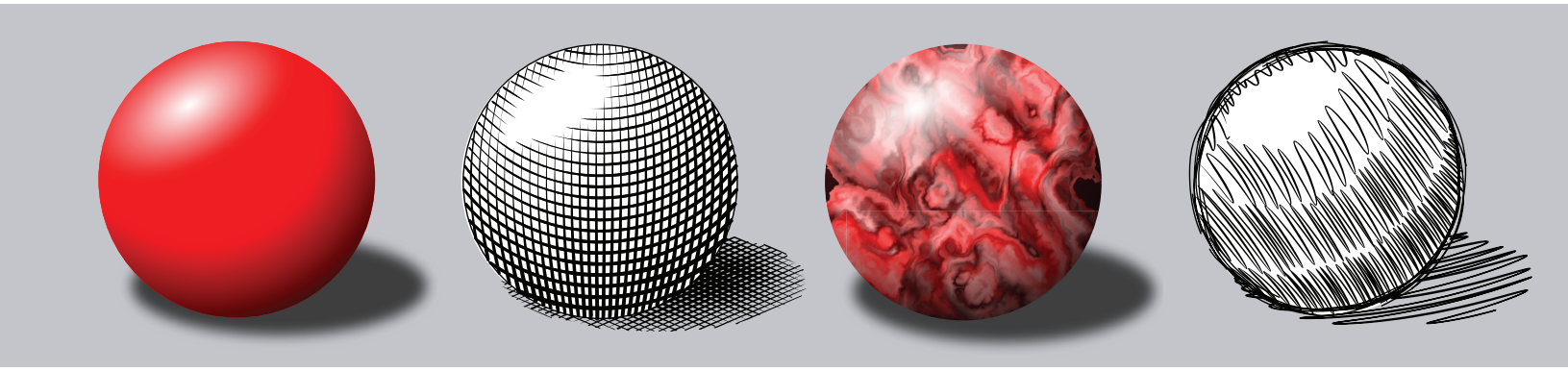
PRENTICE HALL

Inkscape

Guide to a Vector Drawing Program

Third Edition



Tavmjong Bah

sourceforge
Community Press

Library of Congress Control Number 2009938794

ISBN-13: 978-0-13-705173-1
ISBN-10: 0-13-705173-5
Text printed in the United States at Edwards Brothers in Ann Arbor, Michigan.
First printing, November 2009

Introduction

This book serves as both a textbook and a reference for using Inkscape to produce high-quality drawings. It includes a series of tutorials followed by chapters that cover completely each facet of the Inkscape program. The book is full of tips and notes to enable the user to make the best use of the program.

Inkscape is an open source, *SVG*-based[1] vector drawing program. It is useful for drawing:

- Illustrations for the Web
- Graphics for mobile phones
- Simple line drawings
- Cartoons
- Complex works of art
- Figures for articles and books
- Organization charts

The file format that Inkscape uses is compact and quickly transmittable over the Internet. Yet it is powerful and can describe complex drawings that are scalable to any size. Support for the format has been added to web browsers and is already included in many mobile phones.

Inkscape supports the drawing of regular shapes (rectangles, circles, etc.), arbitrary paths, and text. These *objects* can be given a wide variety of attributes such as color, gradient or patterned fills, alpha blending, and markers. Objects can be transformed, cloned, and grouped. Hyperlinks can be added for use in web browsers. The Inkscape program aims to be fully *XML*, *SVG*, and *CSS* compliant.

Inkscape is available prepackaged for the Windows Macintosh, and Linux operating systems. The program and its source code are freely available. They can be obtained from the Inkscape website [http://www.inkscape.org/].

Inkscape is undergoing very rapid development with new features being added and compliance to the *SVG* standard being constantly improved. This manual documents Versions 0.46 and 0.47.

How to Use This Book

Following this introduction, there is a set of tutorials. The tutorials are designed to cover the basics of all the important features found in Inkscape and to lead the reader from the beginning to end of the drawing process.

The bulk of the book is devoted to a detailed discussion of all of Inkscape's features including examples of solving common drawing problems. Both the strengths and weaknesses of Inkscape are pointed out.

Depending on one's background, one may use the book as a reference or read the book from front to back. In general, the more fundamental topics are covered first. Novices are encouraged to work through each of the tutorials sitting in front of their computer. At the end of the book are a few drawing challenges.

Conventions:

- *Click*: Click on icon, object, and so forth with the **Left Mouse** button (unless another mouse button is indicated) with immediate release.

[1] All acronyms are defined in the Glossary.

- *Click-drag*: Click on icon, object, and so forth with the **Left Mouse** button (unless another mouse button is indicated) and hold the button down while moving the mouse.
- Select the option in the pull-down menu. Example: File → Document Properties... (**Shift+Ctrl+D**): Select "Document Preferences..." under the "File" pull-down menu. **Shift**+**Ctrl**+**D** is the keyboard shortcut corresponding to this option.

One-button Mice

Users of one-button mice might want to upgrade to a multi-button mouse. Inkscape makes good use of a three-button mouse with a scroll wheel. (Inkscape also makes good use of graphics tablets.) In the mean time, the button on a one-button mouse corresponds to the **Left Mouse** button.

Icons

The icons used in this book are in general those provided by Inkscape's default icon theme. Some icons, however, are provided by the operating system. It is possible that the icons you see in your version of Inkscape are different depending upon the source of your version. Regardless of what icons are used, the functionality remains the same.

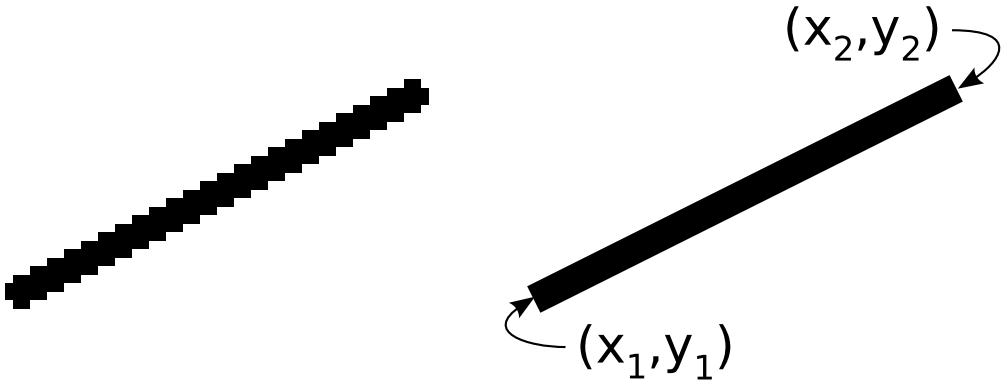
Book Website and Color Addendum

The book has a website [http://tavmjong.free.fr/INKSCAPE/] with some *SVG* examples and tests as well as graphics for use with the tutorials.

Being a drawing program, color is very important in Inkscape. You can download from the website a color addendum, which has many of the book's figures in color. Figures in the book that have a color version in the addendum are marked with the symbol ✪.

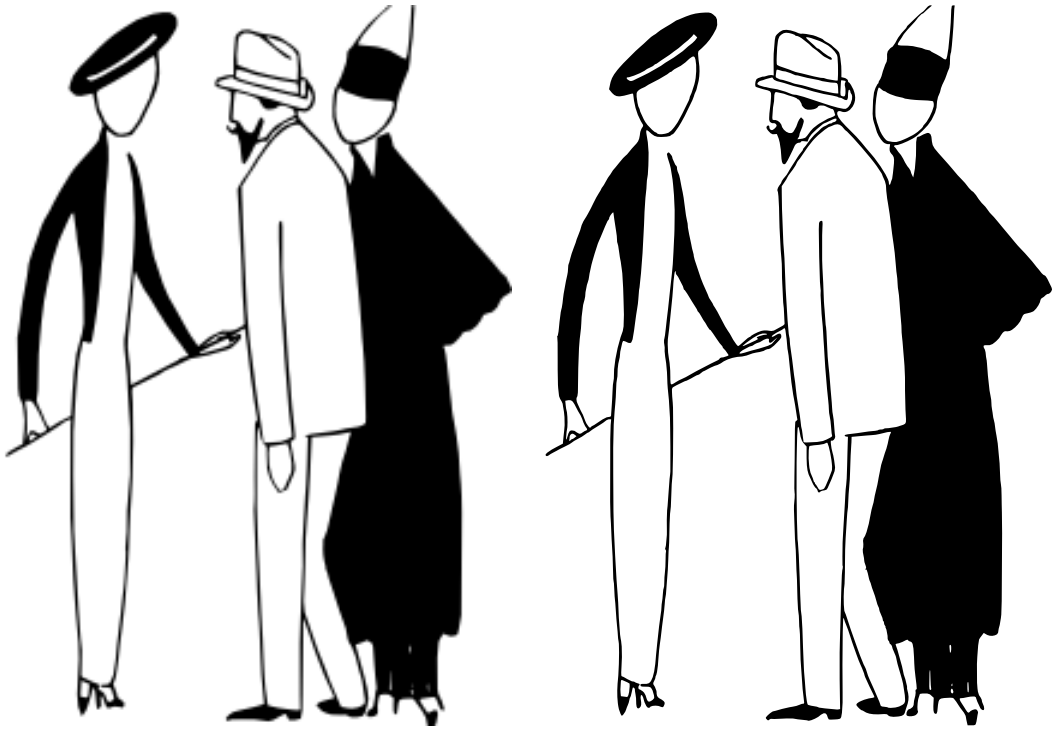
Vector Graphics

There are two basic types of graphic images: *bitmap* (or *raster*) images and *vector* images. In the first case, the image is defined in terms of rows and columns of individual pixels, each with its own color. In the second case, the image is defined in terms of lines, both straight and curved. A single straight line is described in terms of its two end points. The difference in these types of graphic images becomes readily apparent when a drawing is enlarged.



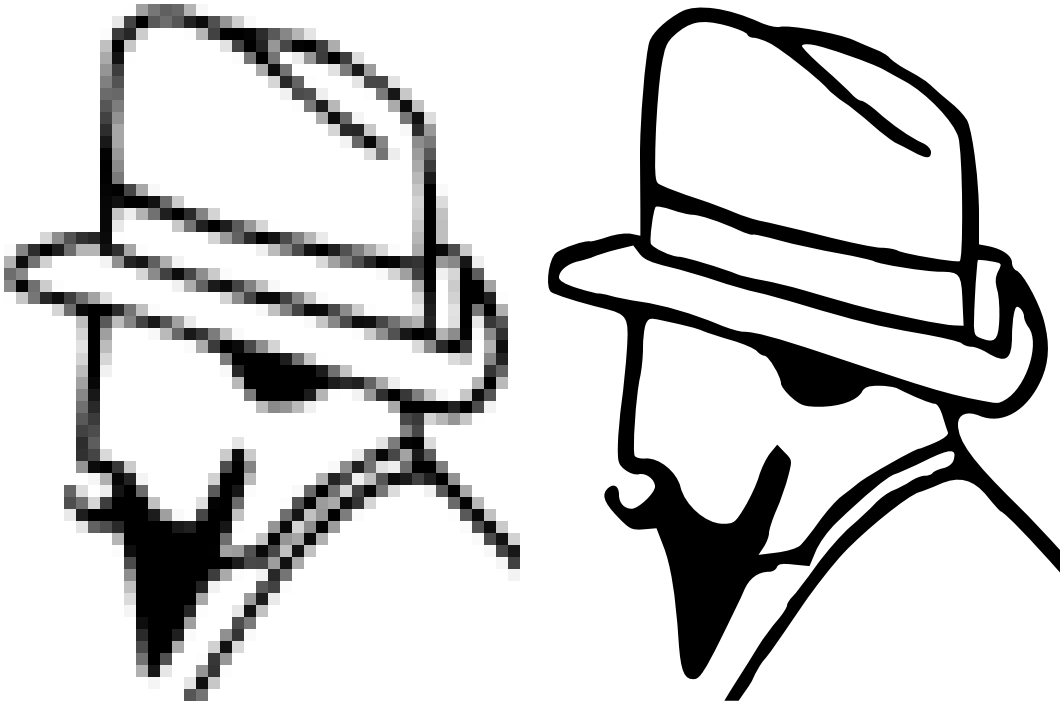
The same line is shown on the left and right. On the left it is displayed as a bitmap image, while on the right it is displayed as a vector. In both cases, the line has been scaled up by a factor of four from its nominal size.

When the bitmap resolution of a drawing matches the display resolution, the objects in the drawing look smooth.



The same drawing, but defined as a bitmap image on the left and a vector image on the right. If the output device has the same resolution as the bitmap image, there is little difference between the appearance of the two images.

If the bitmap resolution is significantly less than the display resolution, the display will show jagged lines.



The head of the gentleman in the above drawings has been scaled up by a factor of five. Now one can see a difference in the quality of the bitmap drawing (left) and the vector drawing (right). Note that the bitmap image uses *anti-aliasing*, a method of using grayscale to attempt to smooth the drawing.

All output devices, with few exceptions, use a raster or bitmap image to display graphics.[2] The real difference between drawing with bitmap graphics and vector graphics is the point at which the image is converted into a bitmap. In the case of vector graphics, this conversion is done at the very last step before display, ensuring that the final image matches exactly the resolution of the output device.

SVG

SVG stands for *Scalable Vector Graphics*. *Scalable* refers to the notion that a drawing can be scaled to an arbitrary size without losing detail.

Scalable also refers to the idea that a drawing can be composed of an unlimited number of smaller parts, parts that can be reused many times.

The *SVG* standard is directed toward a complete description of two-dimensional graphics including animation in an *XML* (eXtensible Markup Language) format. *XML* is an open standard for describing a document in a way that can be easily extended and is resistant to future changes in the document specification. A drawing saved in one version of *SVG* by one version of a drawing program should be viewable, to the full extent possible, by any previous or future version of any drawing program that adheres to the *SVG* standard. If a program doesn't support something in the *SVG* standard, it should just skip over any part of a drawing that uses it, rendering the rest correctly.

SVG files are small, and drawings described by the standard adapt well to different presentation methods. This has led to great interest in the standard. Support is included in many web browsers (Firefox, Chrome, Opera, and Safari), or is available through plug-ins (e.g., Adobe [http://www.adobe.com/svg/viewer/install/], RENESIS [http://www.examotion.com/], Ssrc SVG [http://www.savarese.com/software/svgplugin/], and soon Google [http://code.google.com/p/sgweb/]). Over a dozen companies including Apple (iPhone), Blackberry, LG, Motorola, Nokia, Samsung, and Sony Ericsson produce mobile phones that utilize a subset of the full *SVG* standard that has been tailored for devices with limited resources.

The Inkscape Program

Inkscape has its roots in the program Gill (GNOME Illustrator application) created by Raph Levian [http://www.levien.com/] of Ghostscript fame. This project was expanded on by the Sodipodi [http://sourceforge.net/projects/sodipodi] program. A different set of goals led to the split-off of the current Inkscape development effort.

The goal of the writers of Inkscape is to produce a program that can take full advantage of the *SVG* standard. This is not a small task. A link to the road map for future development can be found on the Inkscape website [http://www.inkscape.org/]. Of course, you are welcome to contribute!

Instructions on installing Inkscape can be found on the Inkscape website. Full functionality of Inkscape requires additional *helper* programs to be installed, especially for *importing* and *exporting* files in different graphic formats. Check the log file `extensions-errors.log` located on Linux at `~/.inkscape/` (v0.46) or `~/.config/-inkscape/` (v0.47) and on Windows at `%userprofile%\Application Data\Inkscape\` for missing programs.

Help

The first place to look for help is under the *Help* menu. Here you will find links to: this book (!), a web page (as of v0.46) containing all the *Keyboard* and *Mouse* commands (Help → Keys and Mouse Reference), tutorials, and a FAQ. Some of the items require a web browser and that you be connected to the Internet.

[2] The few vector output devices include large plotters for engineering and architectural drawings and archaic Tektronix terminals.

If you encounter a problem that is not covered by this book or the other resources under the *Help* menu, here are some other places to look:

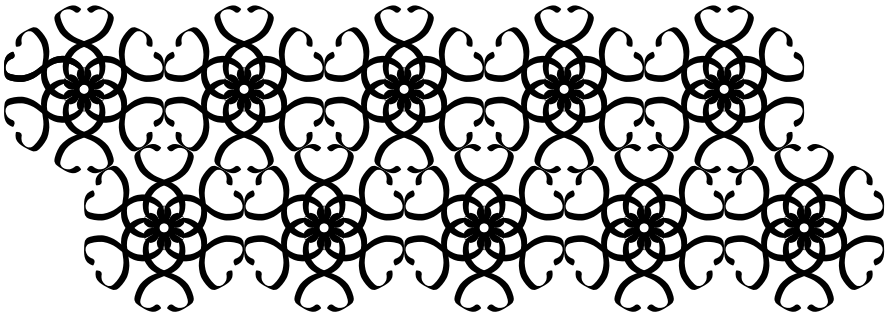
- Inkscape website [http://www.inkscape.org/]. A variety of information is available, but it is not always well organized.
- Inkscape FAQ [http://wiki.inkscape.org/wiki/index.php/FAQ]. A good place to look for answers to common questions.
- Official Mailing Lists [http://www.inkscape.org/mailing_lists.php]. Inkscape has a friendly *Users* list. Lists also exist in a variety of languages including Italian, Spanish, French, and Portuguese.
- The “unofficial” Inkscape Forums [http://www.inkscapeforum.com/].

Chapter 18. Tiling

Tiling or tessellation is the covering of a surface with the repeated use of the same shape tile. A typical example is the tiling in a bathroom. In Inkscape, this concept is expanded to include a multitude of options including progressively changing the tile size, spacing, and orientation.

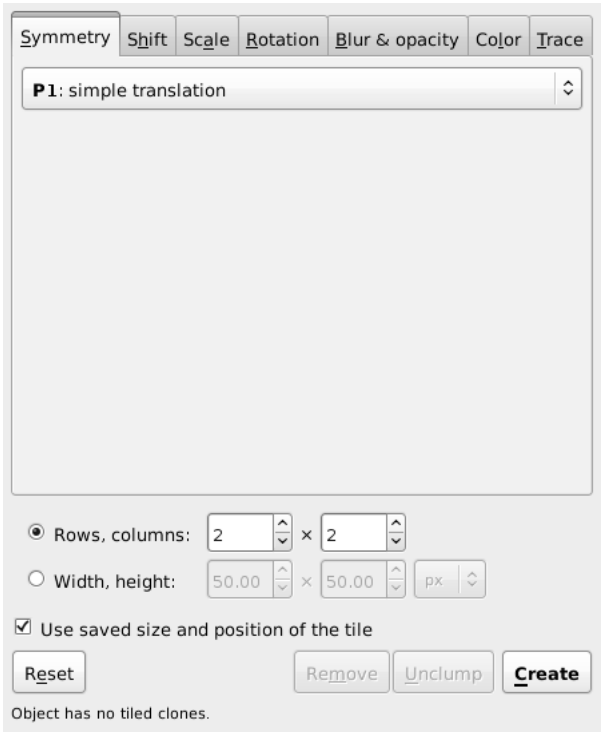
The tiles are in reality just clones of the source tile or object. Thus the same methods that apply to clones apply to tiles. (See the section called *Clones* in Chapter 4, *Editing Basics*.)

While random use of the *Tile Clones* dialog can produce exquisite patterns, it is useful to understand the fundamentals of tessellation in order to have more control over the final design.



An example of using the *Tile Clones* dialog with a simple calligraphic stroke and the *P6M* symmetry group (see text).

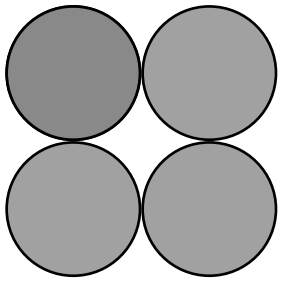
To construct a tiling, open up the *Create Tiled Clones* dialog (Edit → Clone → Create Tiled Clones...).



The *Tile Clones* dialog with no objects selected.

At the bottom of the dialog is a fixed section where you can choose the size of the tiling either by the number of rows and columns or by the width and height of the area you wish to cover. The terms *Rows* and *Columns* are only really appropriate for tiling of rectangular tiles (see below). Checking the “Used saved size and position of the tile” forces the tiling to use the size and position of the base tile at the last time the tile was used in a tiling. This preserves the spacing between tiles if the *bounding box* has changed due to editing the base tile. Clicking on the *Reset* button resets most of the entries under the tabs to their default values. The *Remove* button can be used to undo a tiling when the base tile is selected. The *Unclump* button can be used to spread out the clones in a somewhat random fashion (can be repeated). And, finally, the *Create* button creates the tiling.

With a circle and the default values (P1 symmetry, two rows and two columns), you will get the following tiling:



The simple tiling of a circle. The symmetry is "P1" and there are two rows and two columns.

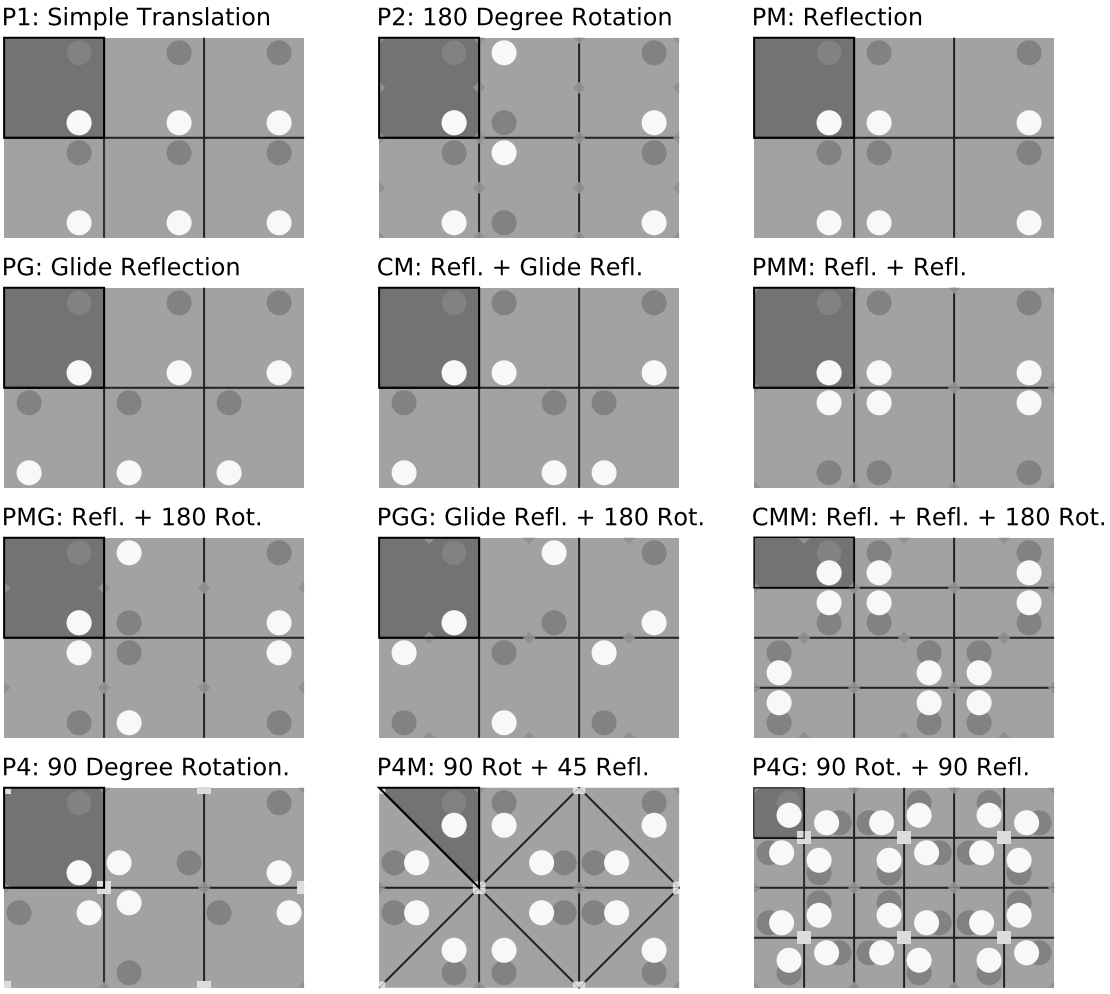
The circle has been replicated four times in two rows and two columns. The original circle is still there, under the top-left cloned circle. The *bounding box* of the circle has been used as the base tile size.

This example is not so interesting, but there are many options under the dialog's tabs that can produce many interesting effects. Each tab will be covered in turn in the following sections.

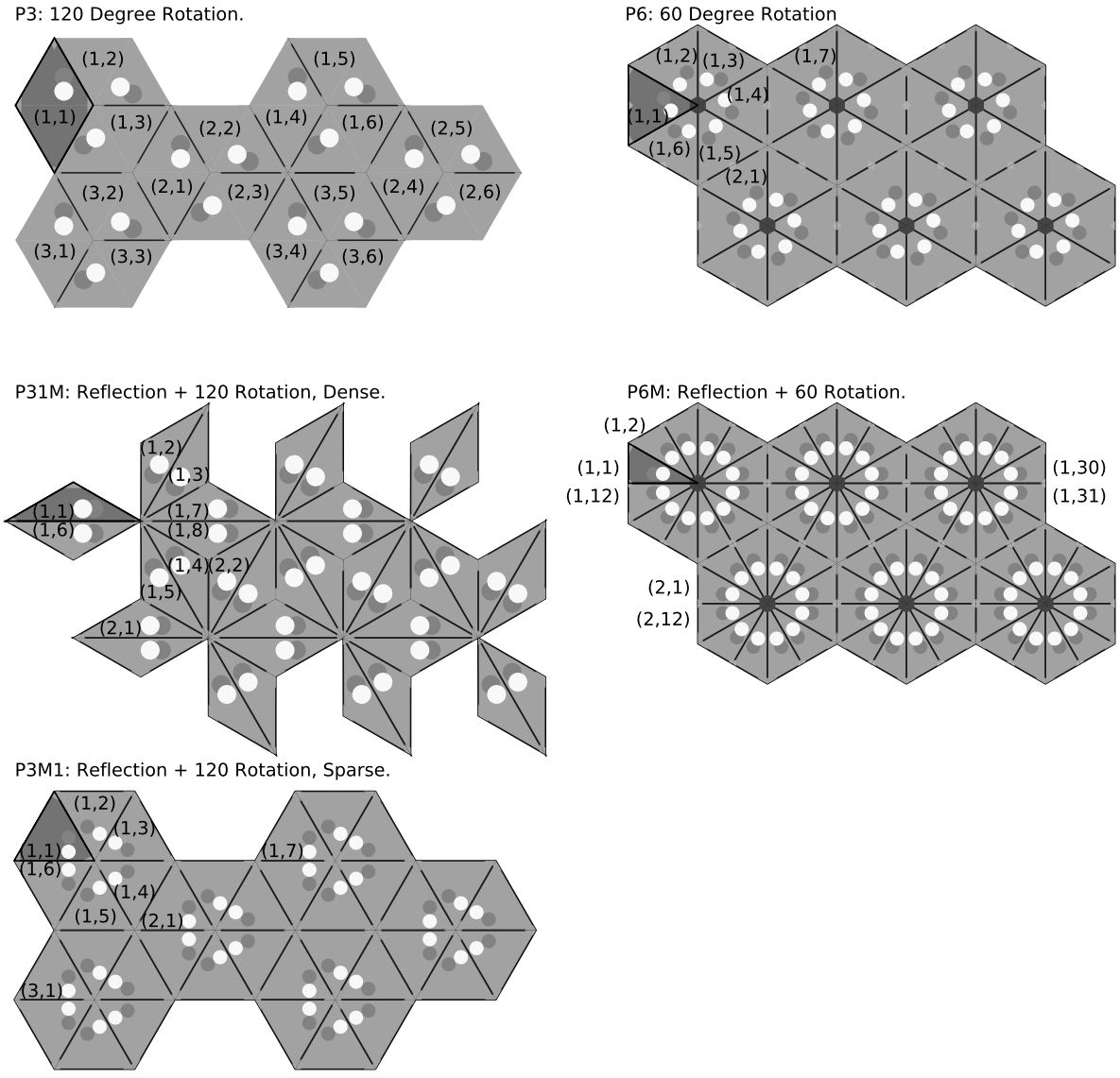
Symmetry Tab

The *Symmetries* tab is at the heart of the tiling process. Understanding the different symmetries is necessary to have full control over the outcome of a tiling. The symmetry of the tiling is selected from the pull-down menu under the *Symmetries* tab (see above figure).

There are three regular geometric shapes that can be replicated to cover a surface completely (without gaps or overlaps). These shapes are: triangles, rectangles (parallelograms), and hexagons. A complete set of tiling symmetries requires taking these shapes and adding rotations and reflections. It is known that there are 17 such tiling symmetries. (See: Wikipedia entry [http://en.wikipedia.org/wiki/Wallpaper_group].) All 17 symmetries are included in the Inkscape *Create Tiled Clones* dialog. The symmetries are shown next.

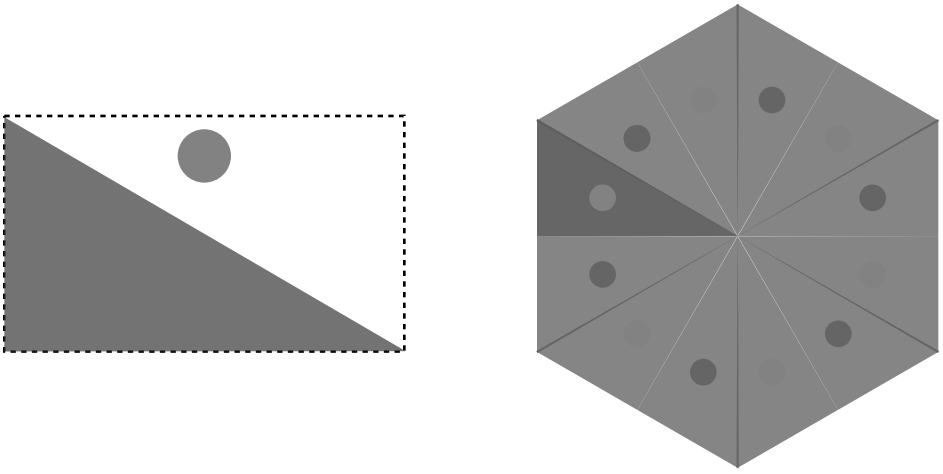


Tilings based on a rectangle tile (or 45-45-90 degree triangle). The outlined dark blue tile is the basic unit. Red and yellow dots show the reflection and rotation symmetries. Points of twofold and fourfold rotational symmetry are shown by pink diamonds and green squares, respectively. The P1 and P2 symmetries also work with parallelograms.



Tiling based on regular subdivisions of a hexagonal. The outlined dark blue tile is the basic unit. All tiling have points of threefold rotational symmetry (orange triangles). Two also have twofold and sixfold rotational symmetries (pink diamonds and purple hexagons). The pairs of numbers indicate the row and column numbers.

The basic tile for each of the 17 symmetries is shown in dark blue in the preceding figures. Inkscape uses the *bounding box* of an object to determine the basic tile size. For rectangular base tiles, the *bounding box* corresponds to the base tile. However, for triangular base tiles, the base tile covers only part of the *bounding box* area. This can result in tiles "overlapping" if an object extends outside the base tile shape (but is still within the *bounding box*) as in the tiling in the introduction to this chapter. Overlapping can also occur if the base tile is altered after the tiles are positioned.



On the left is a triangle and circle that are grouped together. The triangle corresponds to the base tile for a P6M symmetry. Note that the red circle is outside the base triangle but is still within the *bounding box* of the group (and triangle). On the right is a P6M tiling with the triangle and circle. Note how the red circle ends up above some but below other triangles as determined by the order in which the tiling is made.

As of v0.46, Inkscape always uses the *Geometric bounding box* to determine the tile size. This avoids problems when creating a triangular tile with a *Stroke* where the *Visual bounding box* doesn't have the same width to height ratio as the *Geometric bounding box*.

If you need to adjust the base tile size after having creating a tiling, you can use the *XML Editor* dialog to change the parameters “inkscape:tile-h” and “inkscape:tile-w” (these will appear after you have cloned the object and are used only if the *Use saved size and position of the tile* button is checked).

Shift Tab

The *Shift* tab allows one to vary the spacing between tiles. With the default parameters, rectangular tiles are arranged so that their *Geometric bounding boxes* are touching. The following options are available to add or subtract space between the tiles:

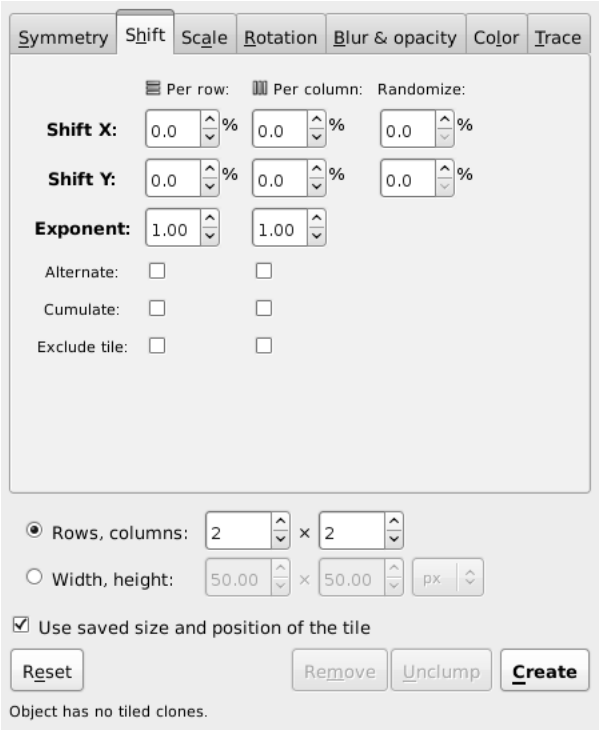
- *Shift X*, *Shift Y*: Adds (or subtracts) to the tile spacing in units of *bounding box* width and height. A random factor can also be added.

- *Exponent*: Changes the exponent factor z so that position of each tile is x (or y) = (1 + “shift”)^z.

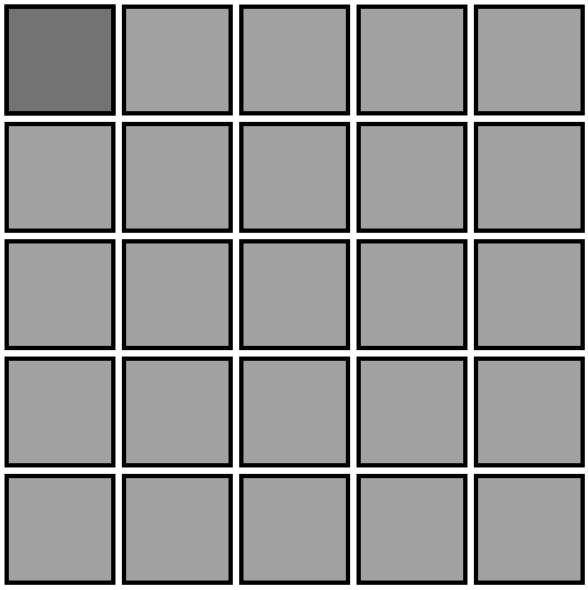
- *Alternate*: The shift alternates between being added and subtracted.

- *Cumulate*: The previous shift is added to the new shift. For example, if there was a *Shift X* of 10%, normally the space between subsequent tiles would be 10%, 20%, 30%, and so on. With this option, the shifts become 10%, (10+20)% (10+20+30)%, etc. This is useful when one is also scaling the tiles to keep the tile spacing constant. (See the *Scale Tab* section for an example.)

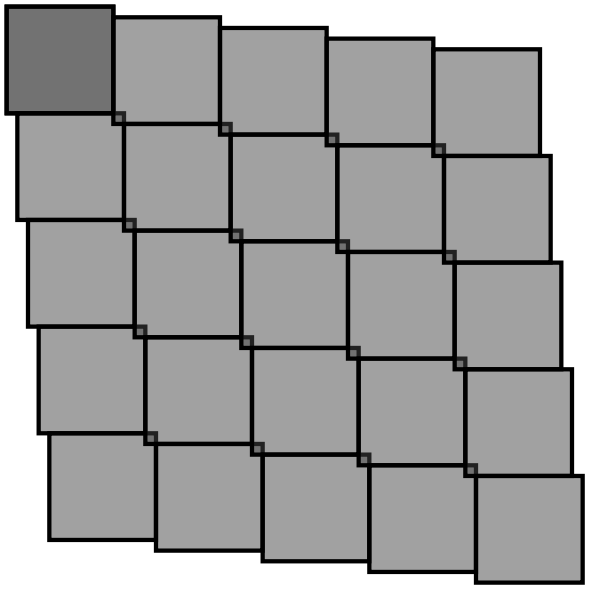
- *Exclude tile*: The tile width or height is excluded in the calculation of tile spacing. This is useful when using the *Rotation* option to put tiles on a circle. In this case, it is a shortcut for specifying a −100% shift.



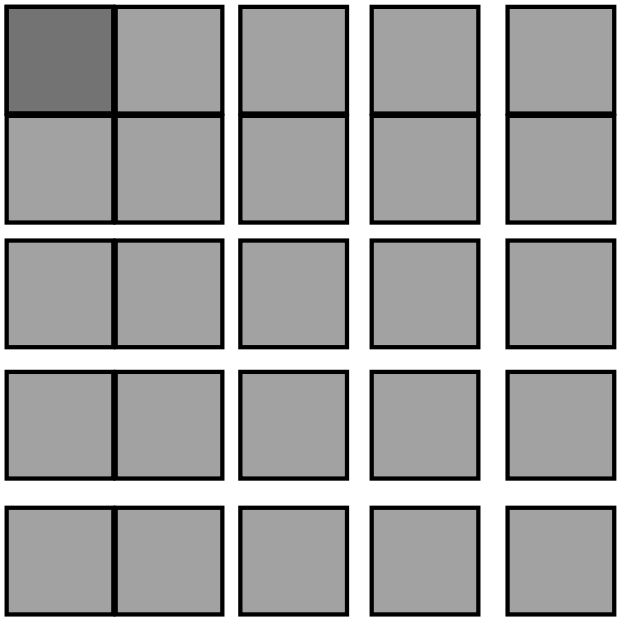
The *Shift* tab of the *Tile Clones* dialog.



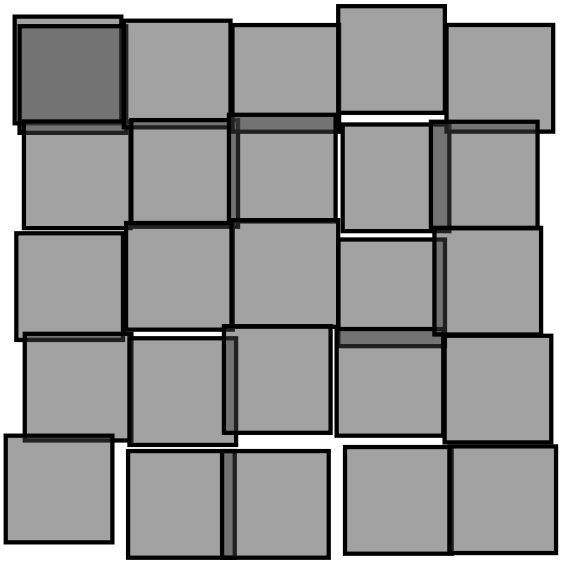
A P1 symmetry tiling with a constant shift of 10% (of the *bounding box*). There is an *x* shift for each column and a *y* shift for each row.



A P1 symmetry tiling with a constant shift of 10% (of the *bounding box*). There is a *y* shift for each column and an *x* shift for each row.

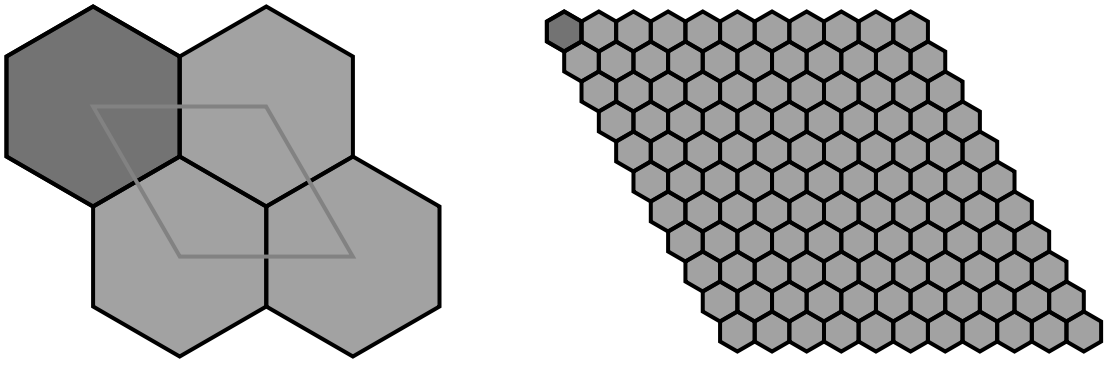


A P1 symmetry tiling with an exponential shift of 1.1 (2% shift in *x* and *y*).



A P1 symmetry tiling with a random shift of 10% (of the *bounding box*) in both *x* and *y*.

Question: What is the symmetry of closely packed hexagons? The answer is P1 as can be seen below. One can use this fact to trivially generate the board for the game Hex [http://en.wikipedia.org/wiki/Hex_%28board_game%29] invented independently by the mathematicians Piet Hein and John Nash.

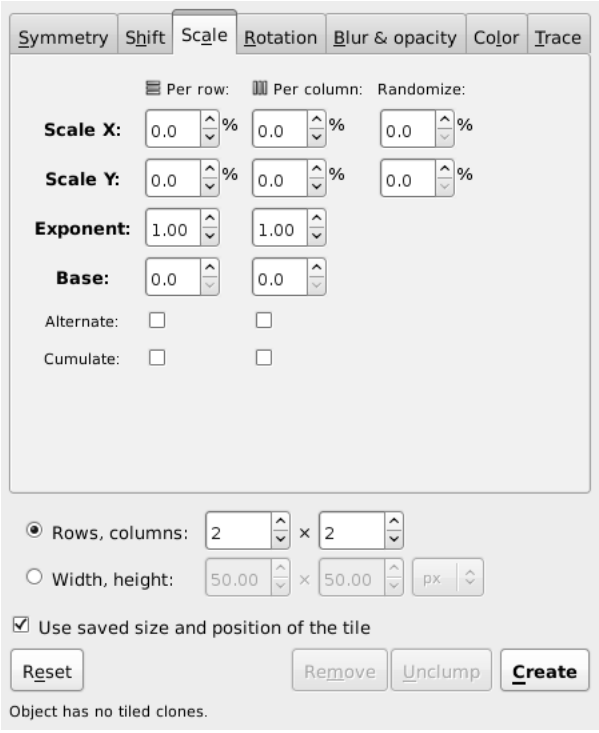


Closely packed hexagons have a P1 symmetry tiling as shown on the left. On the right is the board for the game Hex. To generate both tilings, a hexagon was tiled using a shift in *x* of 50% and a shift in *y* of −25% per row.

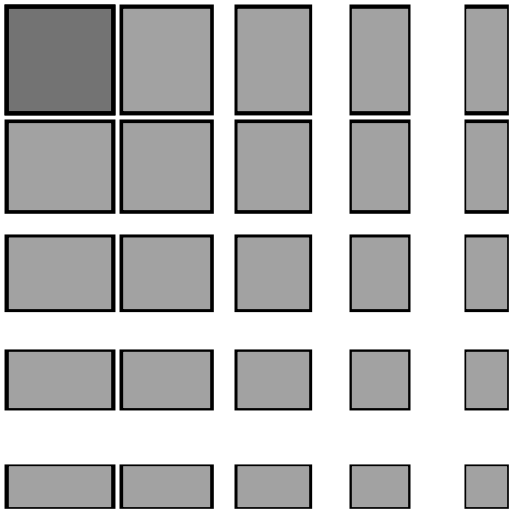
Scale Tab

The *Scale* tab allows one to increase or decrease the size of the tiles depending on the row and column position. The following options are available to scale tiles:

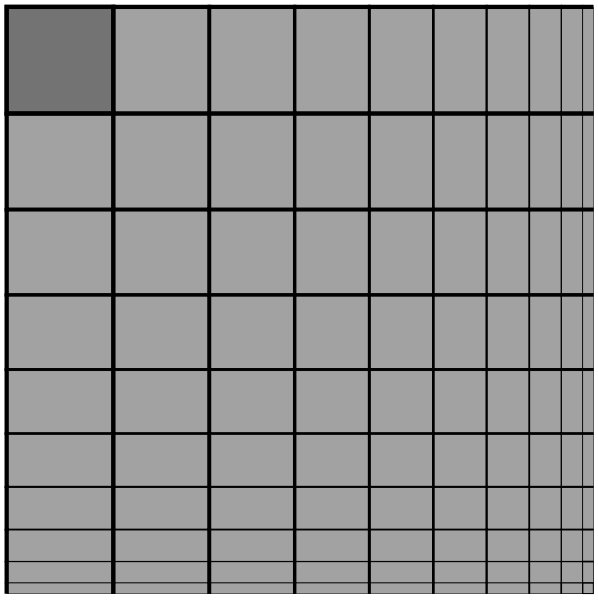
- *Scale X*, *Scale Y*: Scales each tile in terms of percentage. A random factor can be added.
- *Exponent*: Scale each tile with an exponential factor. The nominal scaling S becomes $S^{exponent}$.
- *Base*: Used to create a logarithmic spiral along with *Rotation*. The nominal scaling S becomes $base^{S-1}$ unless *base* is one in which case scaling remains unchanged. Use a value less than one for a converging spiral and a value greater than one for a diverging spiral. A true logarithmic spiral would use a base of $e = 2.718$ (or $1/e = 0.368$). See the *Tile Tricks* section for examples.
- *Alternate*: Alternate scaling up and scaling down tiles.
- *Cumulate*: Scaling is cumulative.



The *Scale* tab of the *Tile Clones* dialog.



A P1 symmetry tiling with a negative scaling. There is an −15% *x* scaling for each column and a −15% *y* scaling for each row. The scaling is a percentage of the base tile *bounding box*. The spacing between the center of adjacent tiles remains fixed.

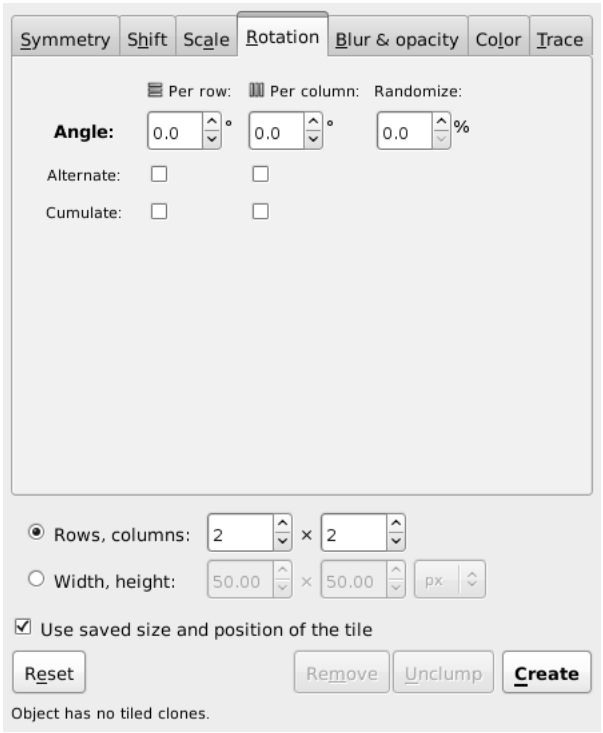


A P1 symmetry tiling with a cumulative negative scaling. There is a −10% *x* scaling for each column and a −10% *y* scaling for each row. There is also a −5% *x* shift for each column and a −5% *y* shift for each row. The *Cumulate* box is checked for both *x* and *y*. A general rule is that to keep scaled tiles just touching, specify a cumulative shift that is half of the scaling (in percent).

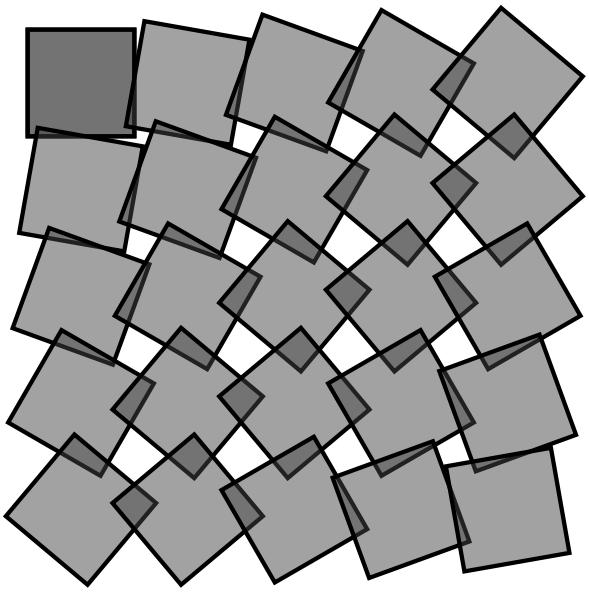
Rotation Tab

The *Rotation* tab allows one to rotate the tiles depending on the row and column position. As of v0.46, the *Rotation center* is used as the center of rotation. See the *Tile Tricks* section for examples of using a shifted *Rotation center*. The rotation is specified in degrees. The following options are available:

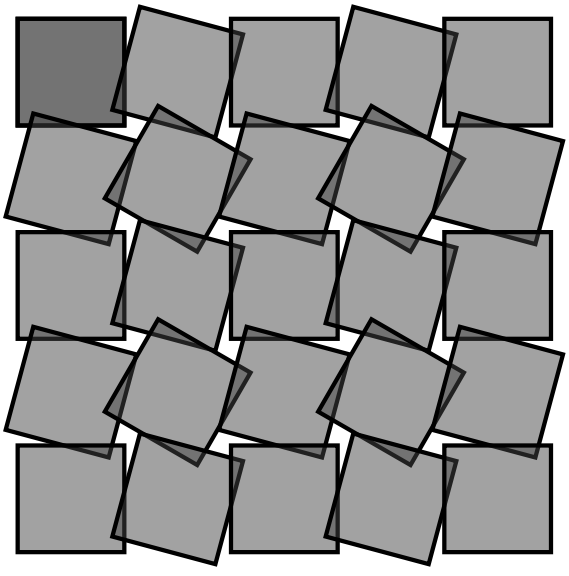
- *Angle*: Rotate by this amount around the *Rotation center*. A random factor can also be added.
- *Alternate*: The rotation alternates between being added and subtracted.
- *Cumulate*: Rotation is cumulative.



The *Rotation* tab of the *Tile Clones* dialog.



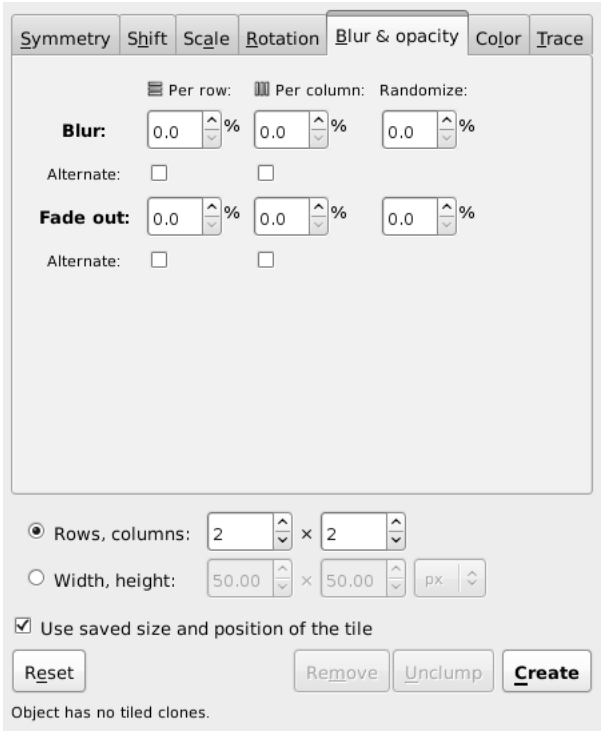
A P1 symmetry tiling with a 10° rotation for each row and column.



A P1 symmetry tiling with a 15° alternating rotation for each row and column.

Blur and Opacity Tab

The *Blur and opacity* tab allows one to change the *blur* and/or *transparency* of each tile depending on the row and column position.

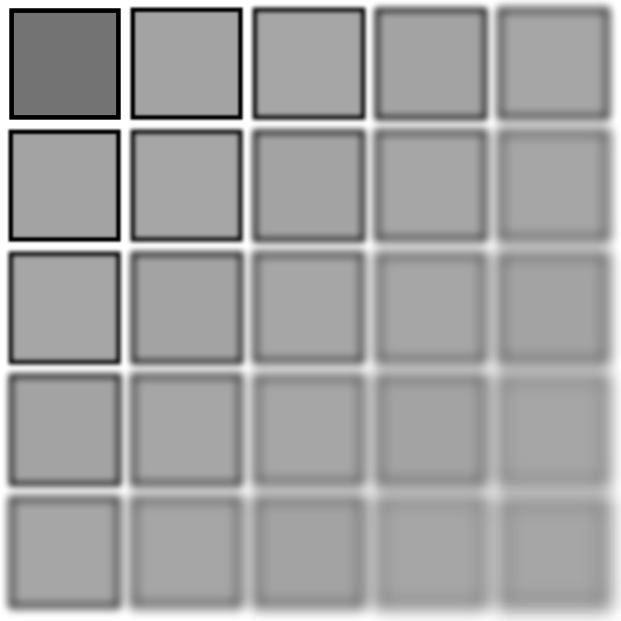


The *Blur and opacity* tab of the *Tile Clones* dialog.

Blur

A *Gaussian Blur* filter can be applied to each clone with different blurring values.

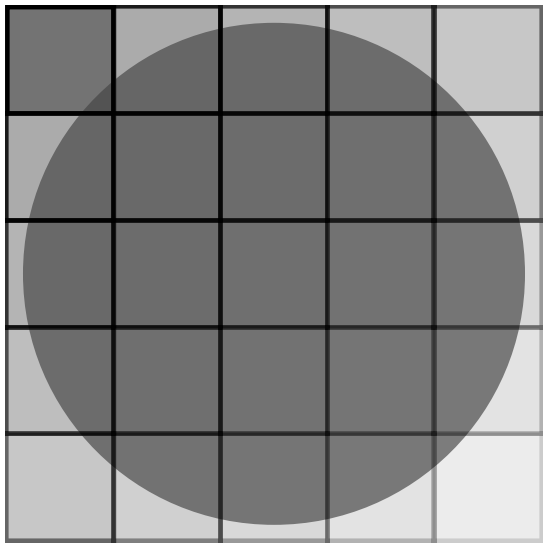
The blur change is specified in percent. The change in blur can be specified to *Alternate* between a positive and negative value; however, a negative blur value can be entered in the *Per row* and *Per column* boxes. A *Randomizer* factor can also be specified.



A P1 symmetry tiling with a 2% increase in blur for each row and column.

Opacity

The opacity change is specified in percent. The change in opacity can be specified to *Alternate* between a positive and negative value. A *Randomizer* factor can also be specified.



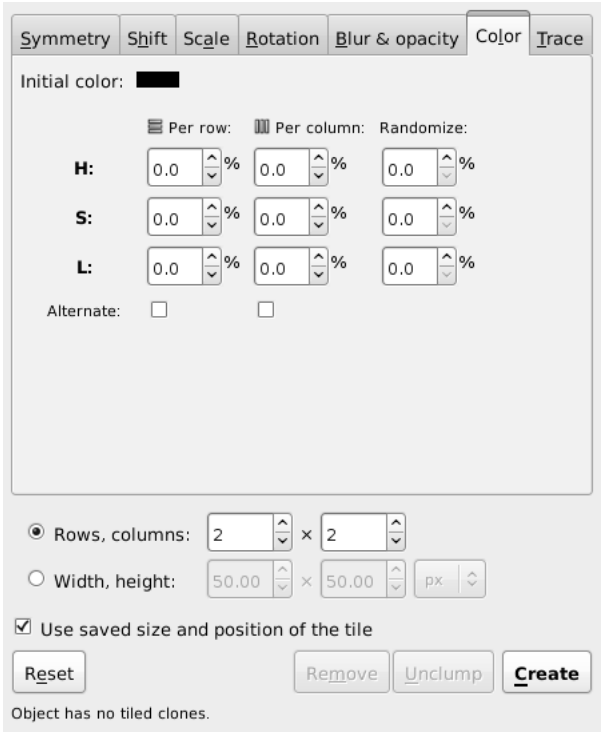
A P1 symmetry tiling with a 10% decrease in opacity for each row and column.
A red circle has been placed under the tiling to illustrate the changes in opacity.

Color Tab

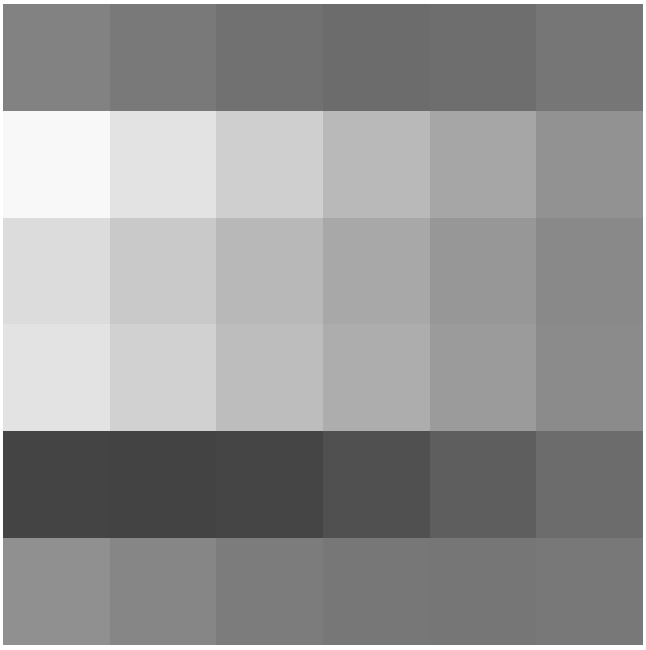
The *Color* tab allows one to change the *Color* of each tile depending on the row and column position. The color change is specified in percent for each of the three components of a color specified with the *HSL* standard (see the section called *HSL*). The *Hue* repeats itself after a change of 100%. The full scale for *Saturation* and *Lightness* components are each 100%. The changes in the three parameters can be specified to *Alternate* between a positive and negative change. A *Randomizer* factor can also be specified.

Two key points: First, the *Fill* and/or *Stroke paint* must be specified as *Unset* (**?**) (see the section called *Fill and Stroke Paint*). Second, an *Initial color* must be specified by using the *Initial color of tiled clones* dialog accessible by clicking on the color button next to the *Initial Color* label.

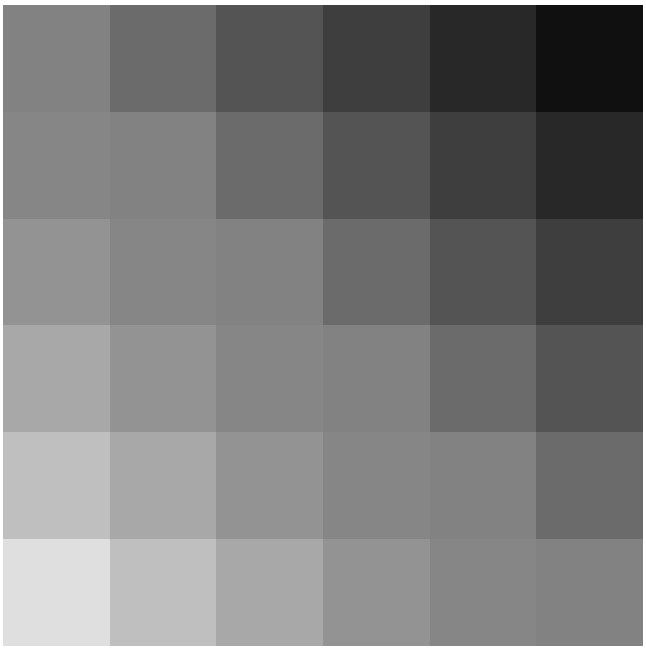
Note that it is meaningless to have only a shift in *Hue* with a starting color of black or white. This is like trying to walk east from the north pole.



The *Color* tab of the *Tile Clones* dialog.



A P1 symmetry tiling with a 16.7% change in *Hue* per row and a −16.7% change in *Saturation* per column. The starting color is a red with 100% *Saturation* and 50% *Lightness*.✪



A P1 symmetry tiling with an 8.3% change in *Lightness* per row and a −8.3% change in *Lightness* per column. The starting color is a red with 100% *Saturation* and 50% *Lightness*.

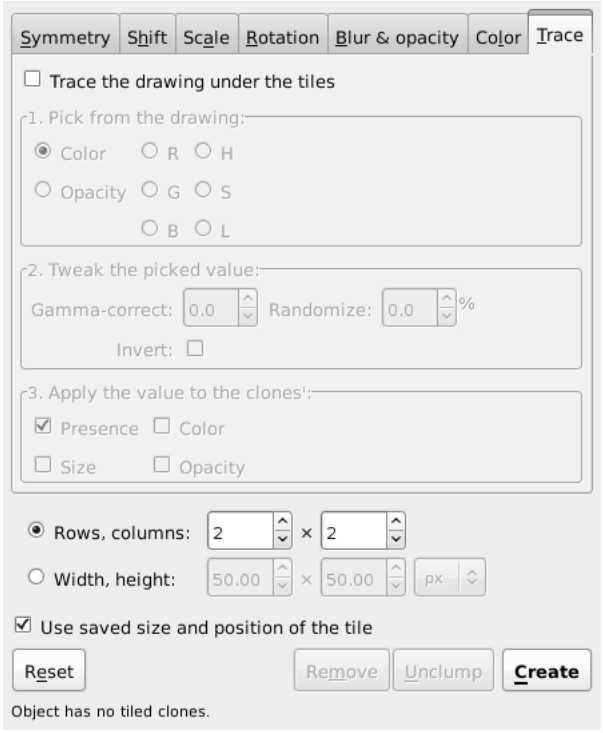
Trace

The *Trace* tab allows one to set the color, size, and transparency of the tiles by the color or transparency of the objects (including bitmaps) that are placed under the location of the tiling. To enable this feature, the *Trace the drawing under the tiles* box must be checked.

The *Trace* tab has three sections. At the top is a section for specifying what property of the underlying drawing should be used for input. Options include the color, one of the *RGB* components, or one of the *HSL* components. There is also the option to use the *Opacity*, which is the sum of the opacities (*Alpha*) of all objects under the tile.

In the middle of the tab is a section to modify the input value. One can specify a *Gamma*[1] correction or add a randomization factor to the input. One can also invert the input.

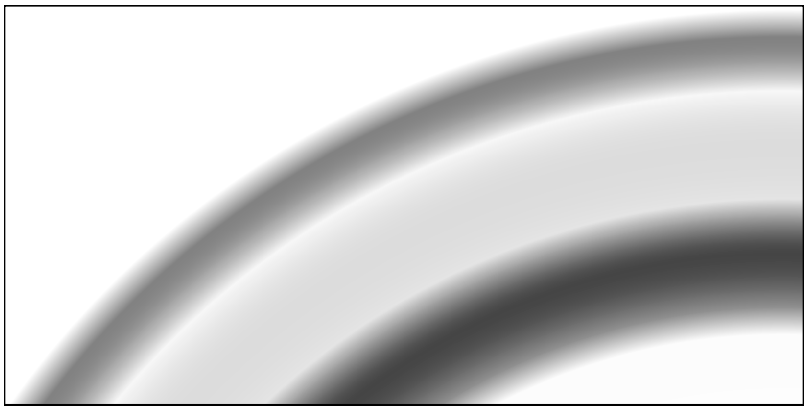
The bottom section is for specifying what should be affected by the input. Options include *Presence* (the probability that a given tile will be drawn), color, size, and opacity. The color will only be changed for regions of the base tile that have *Unset* fill.



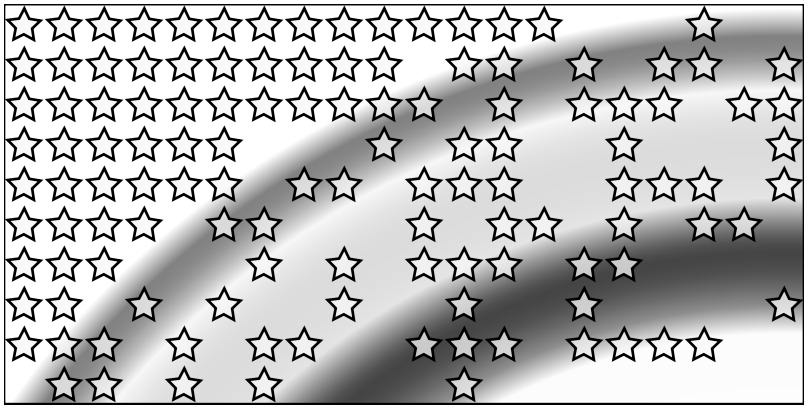
The *Trace* tab of the *Tile Clones* dialog.

The following figures show the effect of some of the possible combinations of input and output options. All the figures use the first rainbow figure as the input drawing. The rainbow is a *radial gradient* with multiple stops. The inside of the rainbow is defined as a white gradient stop with zero *Alpha*. The last outside stop is defined with a red color and with zero *Alpha*. For most figures, a star inside an unfilled rectangle is used as the base tile. The star has been given an *Unset* fill when color is selected in the output.

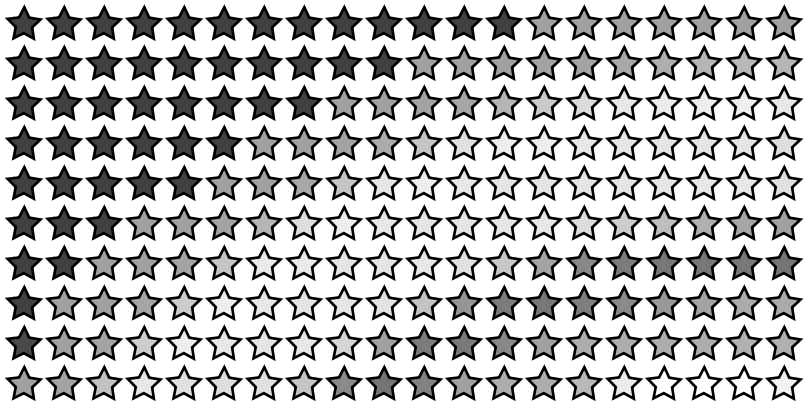
[1] See appendix for definition of *Gamma*.



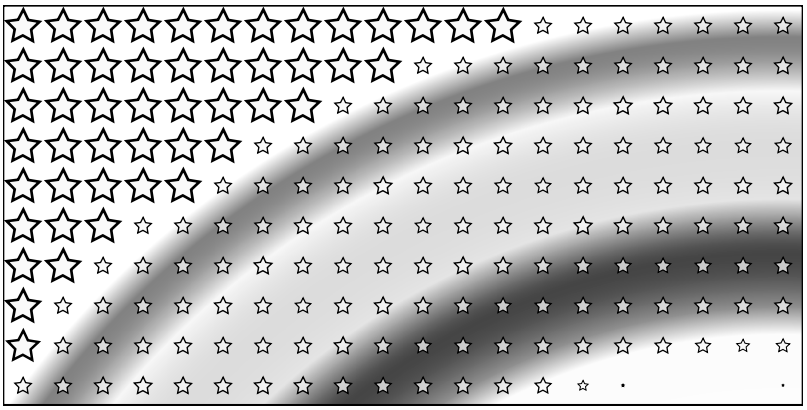
The rainbow pattern used for the background (a radial gradient).



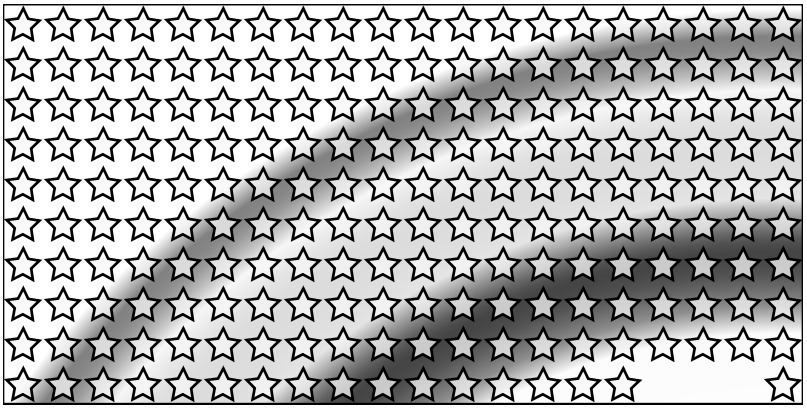
Input: Color. Output: Presence.



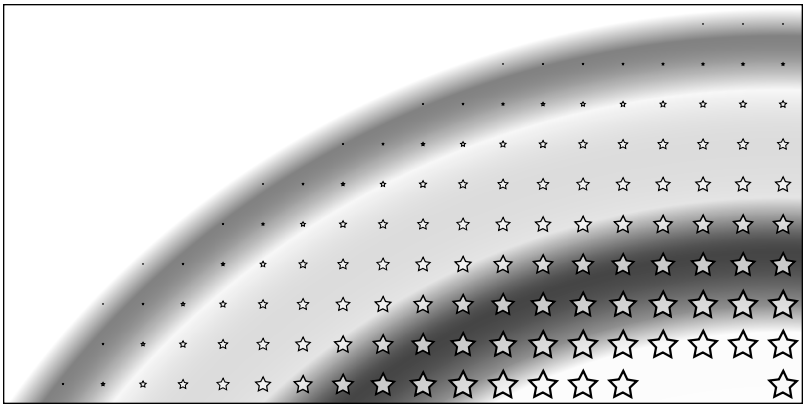
Input: Color. Output: Color. Background rainbow has been removed.



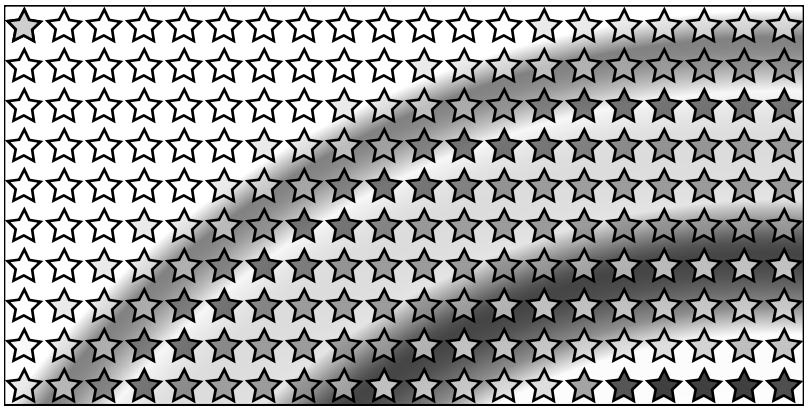
Input: Color. Output: Size.



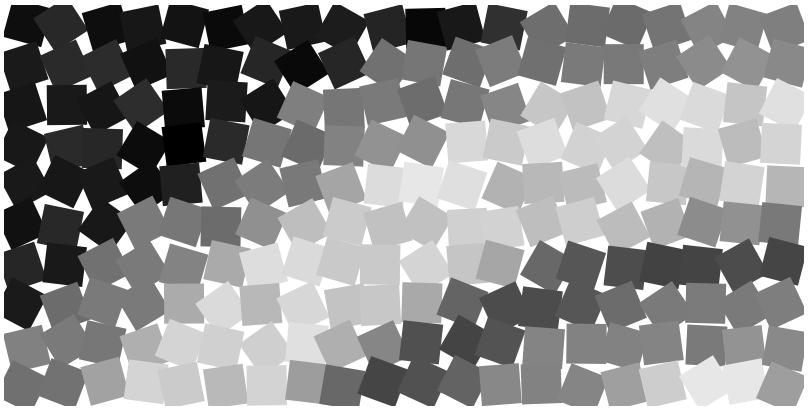
Input: Color. Output: Opacity.



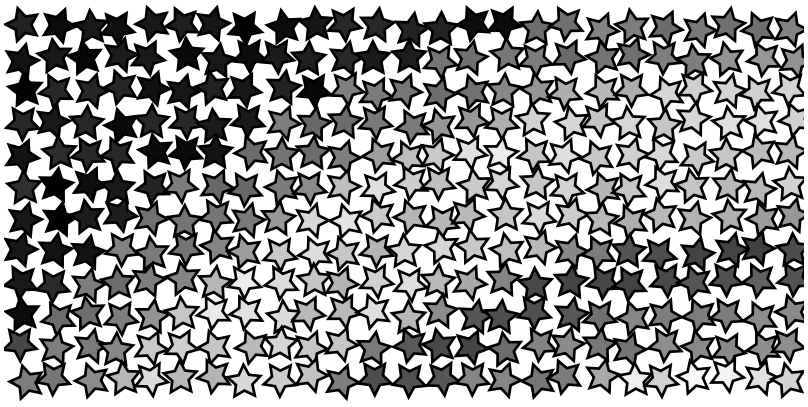
Input: Hue. Output: Size. Note how the red has a hue of zero and purple has the maximum value.



Input: Hue, inverted. Output: Color.



Input: Color, 10% random gamma. Output Color. Changes made to other tabs: Shift: random 10%, Rotation: random 20%. A square base tile with *Unset* fill has been used. The background rainbow has been deleted.

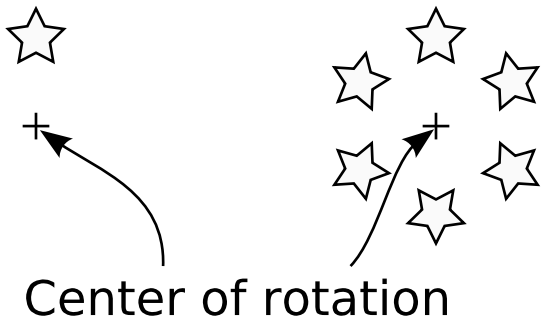


Input: Color, 10% random gamma. Output Color. Changes made to other tabs: Shift: −20%, random 10%, Rotation: random 20%. The number of rows and columns has been increased to compensate for the shift. The background rainbow has been deleted.✪

Tricks

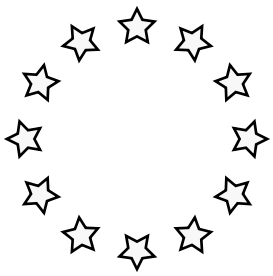
It is possible to exploit the *Tiling* dialog to produce a number of useful effects. The most interesting is placing tiles along an arc or spiral.

To put a tile along an arc use the P1 symmetry with one row of tiles. One use to have to resort to using a *Group* to put tiles along a curve. As of v0.46, the *Rotation center* is used as the center of rotation. One also used to have to specify a shift of −100%. Now one can just check the *Exclude tile* box.



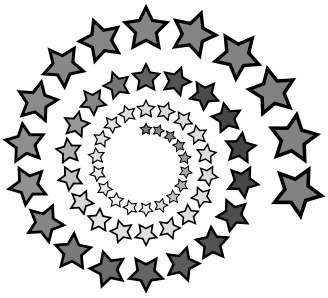
The base tile is drawn on the left, showing the *Rotation center* of the tile. On the right is after a P1 tiling with a per column shift removed by checking the *Exclude tile* box and with a rotation of 60%.

The next figure shows how 12 stars can be put in a circle. This would have been an alternative way of placing the stars in the European Union flag if the stars did not need to be placed with one of their points straight up.

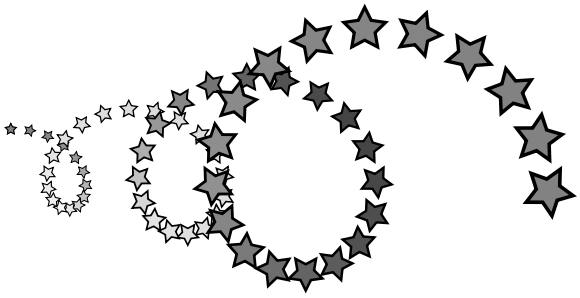


Twelve stars in a circle.

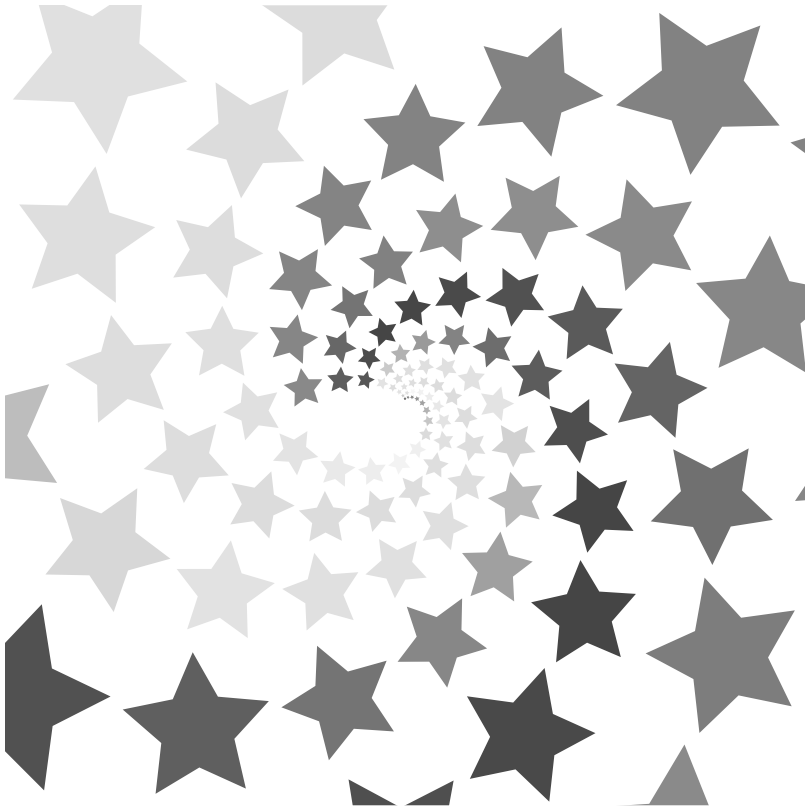
This trick can also place objects along a spiral by specifying that the tile should get larger with each column. As of v0.46, one can put the stars on a logarithmic spiral so that the stars don't run into each after several loops.



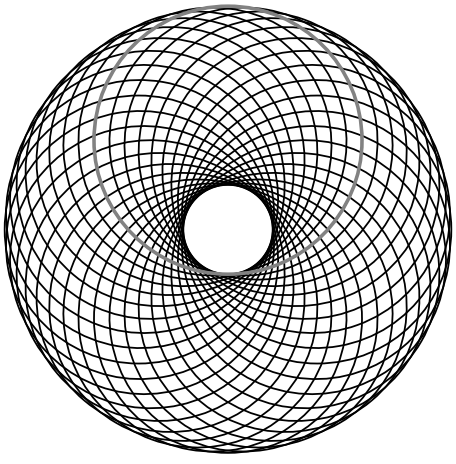
Stars on a logarithmic spiral. The tile size is increased by 2.5% with *Base* set to 2.7. Each tile is rotated 20°.



Stars on a logarithmic spiral. The tile size is increased by 2.5% with *Base* set to 2.7. Each tile is rotated 20°. The per column shift has been set to 60% (with the *Exclude tile* box checked).



A “P1 symmetry” tiling. 8 rows, 21 columns. Rotation of −11.5° per row and 20.6° per column, *Scale* of 39.3% per row and 24.2% per column with a *Base* of 2.7 for both *x* and *y*. The pattern matches that for a pine cone with 8 rows in one direction and 13 in the other. For the mathematicians: note that 13 times the per column scaling is equal to 8 times the per row scaling and that 13 times the per column rotation minus 8 times the per row rotation is equal to 360°. This is due to the constraint that the 14th star in the first row is the same as the 9th star in the first column.



A circle tiled on an arc. The red circle with the *Rotation center* moved off center was the source tile.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z