

# Escher's Patterns

---

Peter Graf, Germany, (2024)

Date: 11.03.2024

*Everything has already been said, but not yet by everyone.*

Karl Valentin  
Bavarian philosopher

## Introduction

---

Years ago I was reading

Stan Wagon, *Mathematica in Action*, 2nd edition  
ISBN-978-0-387-98684-5

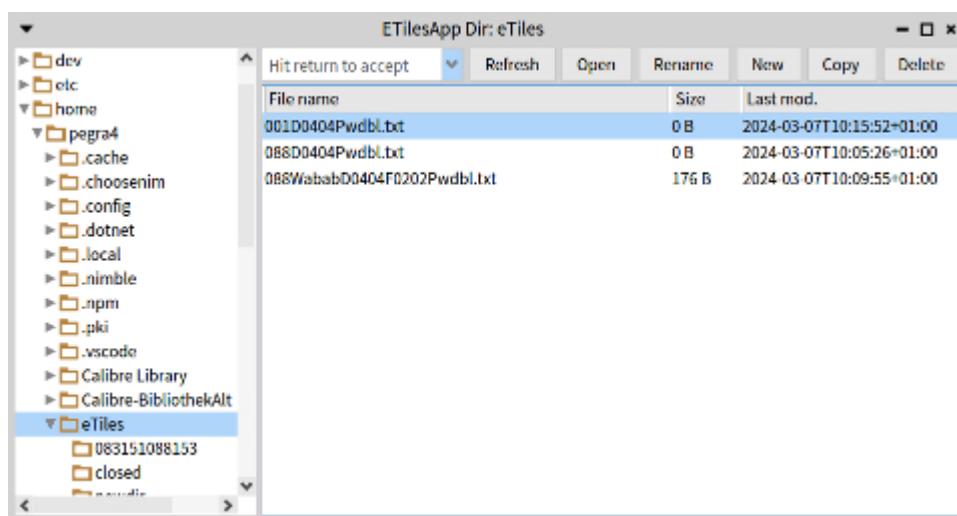
Now I found time for a closer look on the chapter about Escher's Patterns.  
And here's my result.

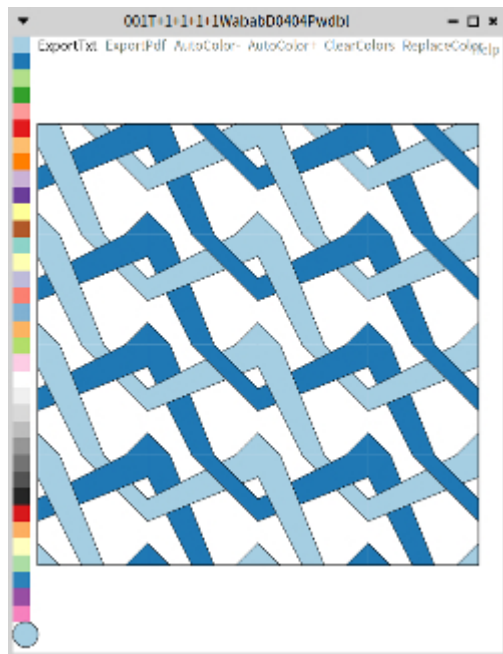
Pharo Smalltalk and the excellent Roassal, thank you for your help.

## ETileApp

---

The ETileApp consists of two windows, a file browser and a tile editor.





A selected file is opened in a new window by pressing the open button.  
The tile editor always takes its input from a filename and the file's content.  
The filename describes one of 154 possible *Escher Tiles*.  
The user can interactively assign colours to the strands and alter the weaving of a tile.  
The current state could be stored in a new file or exported as a PDF.

## What is a tile?

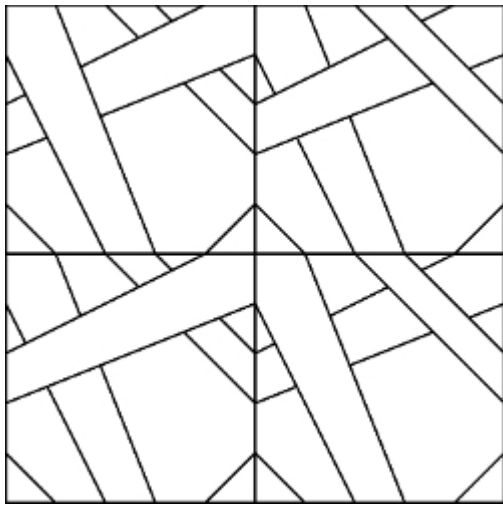
A tile is a rectangle where the colours at the edges match each other on top and on bottom, and on the left and on the right.  
The tile editor takes care of this rule.  
The tile editor additionally doesn't allow to colour crossing different strands with the same colour.

## What is a strand?

A strand is a sequence of polygons that are connected.  
Due to the tile property each polygon having an edge at the border of the big tile is also connected to the polygon on the opposite side of the rectangle.

## 1x1\_tile

A basic Escher Tile is a square with five polygons inside.  
The weaving of a 1x1\_tile describes the possible crossings of its polygons.



The image shows four times the same 1x1\_tile with the four possible weavings:

$\begin{matrix} a & b \\ c & d \end{matrix}$ 
 (= 'abcd' ).

Stan Wagon used different names for the weaving:

" for 'a', 'a' for 'b', 'rr' for 'c' and 'rl' for 'd'.

## What to do with a 1x1\_tile?

A 1x1\_tile can be rotated left and optionally finally flipped horizontally.

These eight operations on a 1x1\_tile are named:

- '+1' no rotation
- '+2' rotate left 90°
- '+3' rotate left 180°
- '+4' rotate left 270°
- '-1' no rotation, flip
- '-2' rotate left 90°, flip
- '-3' rotate left 180°, flip
- '-4' rotate left 270°, flip

Stan Wagon used '1', '2', '3', '4', '-1', '-2', '-3', '-4'.

These operations don't affect the weaving of a 1x1\_tile.

## 2x2\_tile

M. C. Escher always grouped four 1x1\_tiles together in a square.

I call such a group a 2x2\_tile.

There are  $8^4 = 4096$  possible 2x2\_tiles.

But it turned out that there are just 154 different 2x2\_tiles.

All other 2x2\_tiles are similar to one of these.

Dan Davis, On a tiling scheme from M. C. Escher,  
Electronic Journal of Combinatorics (1997)

So a sequence of four operators i.e. '+1+3+1-1' (index=88), describes a 2x2\_tile.

The 154 different 2x2\_tiles are labeled with '001'..'154'.

I use the order introduced by Stan Wagon.

That makes it easier to work with both sources.

## Count 2x2\_tiles

Class ETilesCount2x2 counts (do 'ETilesCount2x2 run') the possible patterns for 2x2\_tiles.

The runtime is 5 seconds.

Transcript shows the list of patterns with the indices "1" to "154" together with the indices of the survivors in the 4096.

If a *rotate left 90°* is done on a 2x2\_tile each 1x1\_tile gets a new position in the 2x2\_tile:

$$\begin{array}{cc} +1 & +2 \\ +3 & +4 \end{array} \quad (= +1+2+3+4) \quad \rightarrow \quad \begin{array}{cc} +2 & +4 \\ +1 & +3 \end{array} \quad (= +2+4+1+3).$$

But each of the four 1x1\_tiles is rotated too:

$$\begin{array}{cc} +2 & +4 \\ +1 & +3 \end{array} \quad (= +2+4+1+3) \quad \rightarrow \quad \begin{array}{cc} +3 & +1 \\ +2 & +4 \end{array} \quad (= +3+1+2+4).$$

So +1+2+3+4 and +3+1+2+4 are similar to each other.

The same idea is used for *rotate left 180°*, *rotate left 270°*, *flip*, *shift1 right* and *shift1 down*.

## Count 3x3\_tiles

Class ETileCount3x3 counts (do 'ETilesCount3x3 run') the possible patterns for 3x3\_tiles.

The runtime is 13 minutes.

There are 934144 survivors in the  $8^9 = 134,217,728$  candidates.

And of course there is no list shown in Transcript.

## Rubber stamps

You can think of a 1x1\_tile as a real rubber stamp.

Because of the flip operator you will need each rubber stamp twice.

And concerning the four possible weavings you will end up with eight rubber stamps.

Equipped like that you would be able to manufacture any of the 154 patterns.

## Proper sizes for rectangles with 2x2\_tiles

If you take just one 2x2\_tile you will find out that they look rather dull when it comes to colouring.

It's because most of them allow just one colour.

Stan Wagon offers a list of minimal rectangles with 2x2\_tiles inside to build a reasonable colourable bigger tile.

I use that list as default values.

But you are free to start with any rectangle  $\geq 2 \times 2$ .

Try '151WababD0202Pwbl.txt'.

It's a pure up and down pattern and allows four colours.

## ETileApp Tile Editor

---

### Filename

The tile editor opens on a given filename that contains parameters:

nnn ['Ttttt'] ['Wwww'] ['Drrcc'] ['Frfcf'] ['P...'] '.txt'

- nnn  
'001'..'154' Index of a 2x2\_tile  
mandatory  
order by Stan Wagon
- Ttttt  
t = '+1'|'+2'|'+3'|'+4'|'-1'|'-2'|'-3'|'-4'  
optional  
is info only  
default: from a table
- Wwww  
w = 'a'|'b'|'c'|'d'  
weavings for a 2x2\_tile  
optional  
default: from a table
- Drrcc  
rr = '02'..'99', number of rows  
cc = '02'..'99', number of columns  
optional  
default: rr, cc from a table (Stan Wagon)
- Frfcf  
rf = '01'..'99', rows factor  
cf = '01'..'99', columns factor

optional

default: rf = '01', cf = '01'

- P...

binary options, up to ' (' or '.' in the filename

optional

default: all of them false

Show in the exported filename

't' shows the pattern for the 2x2\_tile

'w' shows the weaving of the 2x2\_tile

'd' shows the dimension of the big tile

Borders in the tile editor

'b' paints a big frame

'i' paints a frame for each 1x1\_tile

'l' paints the polygons with borders

Actions

'f' colours all uncoloured polygons using few colours

'm' colours all uncoloured polygons using many colours

'p' exports the tile as a PDF and closes the tile editor

The filename specifies what you want to edit and some actions.

The 'nnn' part and the extension '.txt' are mandatory.

All other parts are optional as described above.

The new file '001D0202Pwdbl.txt' will show you an uncoloured big tile with the default weaving.

'Frfcf' will be explained later.

## Working with the tile editor

---

### Choose a colour

Select a colour in the palette.

The selected colour is shown in a circle.

The colours in the displayed palette are the joined palettes paired12, set38, greys8, spectral5 and two colours from set19.

### Colour a strand

Assign the selected colour to a strand with a left mouseclick.

If nothing happens then surely there is a different crossing strand with the same colour you are intending to assign.

Assigning the same colour again to a strand would uncolour the strand.

## Alter the weaving

Alter the weaving in any 1x1\_tile by pushing a polygon under the other polygons in this 1x1\_tile with a right mouseclick.

This will also change the weavings of all corresponding 1x1\_tiles of the big tile.

## More about colours

Where are the colours of a big tile?

It's the content of the file.

Please don't change the content.

The content of the file represents the colours of the polygons.

The specific content is also related to the dimension of the exported big tile. Starting the tile editor with the exported file will show you the same big tile again.

But in case you change the dimension ('D') in the filename, the content would normally not suit to the new dimension.

Opening the tile editor will show you a mess of colours in the big file.

Simply clear all colours and choose 'Autocolour-' or 'Autocolour+'.

But there is a way to repeat the colours of the smaller big tile in a bigger one.

Suppose you have exported a well coloured big tile having the dimension 'D0203'.

To get a bigger shaped big tile with dimension 'D0809' you need to insert the factor part 'F0403' into the filename.

Now open the tile editor on this renamed file.

The entry 'D0203F0403' in the filename will result in a big tile with the dimension 'D0809'.

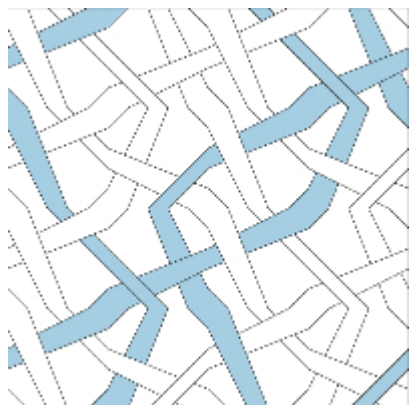
And of course the colours are properly copied into the other tiles.

Remark:

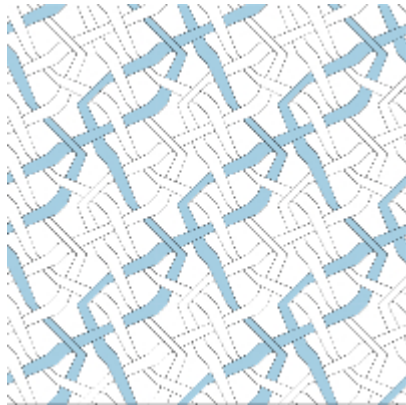
The copying of the colours is done directly from polygon to polygon.

This bypasses the rule that crossing different strands cannot be coloured with the same colour.

The example shows '088D0404'.

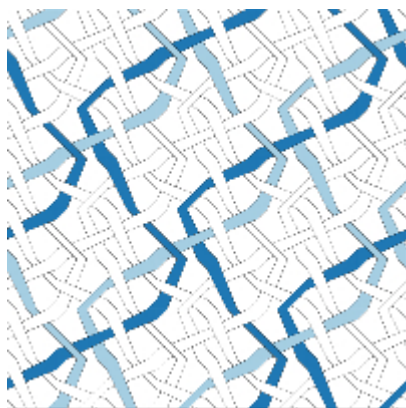


Just one strand is coloured. The strand is self crossing.  
Export it and rename the file to '088D0404F0202Pwdbl.txt'.  
Open '088D0404F0202Pwdbl.txt'.



The tile editor shows already D0808 in the title.  
The colours from 088D0404 were copied to the other tiles.

But now the former self-crossing strand are different strands.  
Colour one of the strands with a different colour.



And now the tile editor would deny to recolour the strand with the colour it had before.

## Tile editor menu

---

### ExportTxt

Writes the current big tile to a new file.  
Please mind the refresh button in the file browser.

### ExportPDF

The pdf contains the canvas with the big tile without the colour palette and the menu.

Therefore the big tile looks like being framed with a broad white margin.



I used an online service to delete the white margin in a pdf file.

Keywords for internet: pdf white margin crop

## Autocolor-

Colours all uncoloured strands using few colours.

All *diamonds* get the same colour.

## Autocolor+

Colours all uncoloured strands using many colours.

All *diamonds* get the same colour.

Short strands like *squares* and *kites* get just few colours.

## ClearColors

Deletes the colours in all polygons.

If the polygons are presented without borders all uncoloured polygons are *cyan*.

## ReplaceColor

Select in the colour palette the colour you would like to replace.

Then select a new colour in the colour palette and press *ReplaceColor*.

All *strands* will get the new colour as if you would have done it manually.

## The tile editor never overwrites a file.

To cope with equal filenames I use the behavior that I know from my chromebook.

In case of a conflict the new file's basename is extended by adding ' (n)'.

Where 'n' is the next possible number making the filename unique.

When renaming a file I always try to get rid of the ' (n)' or use a lower 'n'. So you need not delete ' (n)' manually.

## ETileApp Filebrowser

---

- **Filter:** A text the filename has to start with to be shown in the list of files.  
After a file has been exported from the tile editor, the new file isn't visible in the list of files. To refresh the view on the filenames, press 'Enter' in the filter or just press the *Refresh* button.
- **Refresh** the file list
- **Open** the editor on the selected file
- **New** file
- **Rename** file
- **Copy** file into the same directory and give it a new name.  
Especially before doing *Rename* I myself do a *Copy* and rename the copied file.
- **Delete** file

ETileApp is offering only minimal file handling.  
Please use the filebrowser of your operating system too.

Have fun!