# Integer factorization tessellations - Unpredictable endless variations generated by hard mathematical problems

*Artwork*

**Santo Leonardo**
*ICT Manager / Generative Artist, Italy, Milan*
https://www.instagram.com/leonardo5mart/
*e-mail:* santo.leonardo28@gmail.com

## *Abstract*

Some mathematical problems are simple to describe, however very difficult to solve. Integer factorization (e.g. 28=2x2x7) is one of them; apparently simple, it is exploited in the widely used RSA cryptography because there is no known method for a fast solution on a classical computer.

This project utilizes mathematical hard-to-solve properties related to integer factorization in order to add another level of unpredictability to the generative artwork.

Basically, it transforms any number (n) into an image made of exactly n tiles, attempting to capture the uncertainty and complexity embedded in the natural number sequence. Each integer is decomposed into its prime factors and the resulting structure is used for both an iterative fragmentation process and for coloring the resulting tessellation (see artwork example). By mathematical construction, the created image series has infinite unique variations, while preserving a common style.

The artwork also attempts to match shapes and colors in order to highlight the “numbers’ character”, by connecting color hue and intensity to specific integer properties (e.g. abundancy index). The result is that different types of integers, such as primes, stand out with their identity from the overall succession.

In addition to the variations produced by each integer within the sequence, several parameters can be tuned to generate many different series, which is typical of Generative Art.

**Artwork-example:**
**https://youtu.be/Ca5l0qRFW_U**

## 1. Algorithm description

The algorithm maps any integer (n>=2) in a tessellation of n tiles. As a first step, each n is decomposed into its prime factors ordered from the smallest to the largest ($n=p_1 \times p_2 \times \ldots \times p_k$).

Then the factors list is used in order to produce the geometry of the tiles and their coloring (see figure 1 for the mapping of n=4025=5x5x7x23).

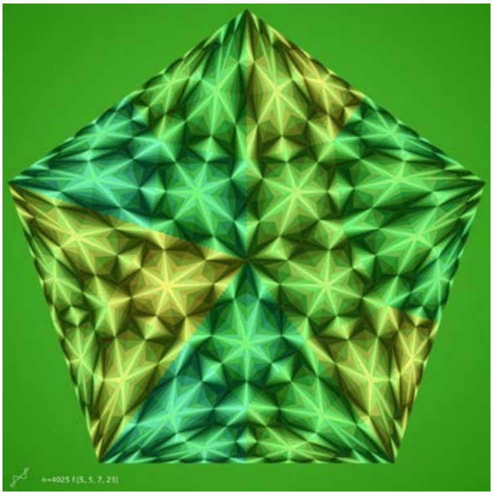

*Figure 1: image of integer n=4025*

## 1.1 Geometry

The tiling process starts with a polygon whose number of sides is linked to the lowest factor ($p_1$), and is iterative. The initial shape is fragmented into $p_1$ tiles (similarly to "splitting a pie"), each resulting tile is then furtherly fragmented into $p_2$ tiles, and the process is iterated until the factors of n are exhausted.

Various approaches can be used for the tile fragmentation into sub-tiles, in particular for the metric used to partition the tile boundary: assigning unit distance to consecutive tile vertex ("vertex metric") [2], Euclidean distance, etc. (see figure 2).

According to the "fundamental theorem of arithmetic" [3], the prime factorization of integers is unique, therefore each n has a one-to-one correspondence with its image.

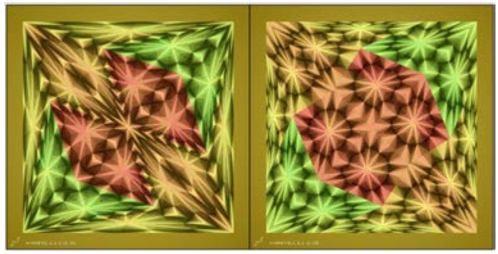

*Figure 2: integer n=4048 mapped with "vertex metric" (left) and Euclidean metric (right)*

## 1.2 Colors

The algorithm is built in the HSI color space (Hue, Saturation, Intensity), where each tile receives a color as a function of its position in the fragmentation procedure (referred hereafter as "tile index", $T_i$).

The largest factors of n are used together with the tile index in order to build the **Intensity** function (e.g. based on $T_i$ modulo $p_k$) [4]. As those factors are the last ones to be employed during the fragmentation process, color intensity highlights the geometry details (see figure 3).

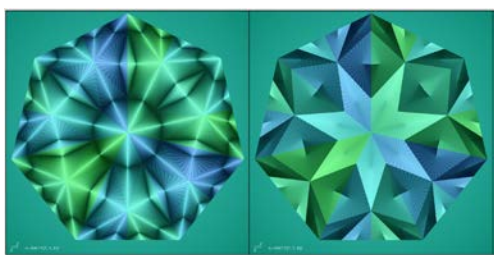

*Figure 3: integer n=4067=7x7x63 mapped with Intensity function of factor 63 (left) and 7x63 (right)*

The average **Hue** is calculated through an additional mathematical hard-problem: it is a function of the "abundancy index" (ratio of the sum of divisors of n over n). Since Euclid's time, it has remained unknown how many integers are equal to the sum of their proper divisors ("perfect numbers", e.g. 28=1+2+7+14). The proposed coloring criterion adds unpredictability to the sequence and gives a characteristic tone to each n: integers smaller than their sum of proper divisors ("deficient numbers") have an image tending to blue and, conversely, "abundant numbers" tend to red [5].

Within each image, every tile has a different Hue according to its tile index. The Hue variation is a function mainly of the smaller factors ($p_1$, $p_2$...), therefore linking Hue distribution to a high-level view of the image [6].

Finally, **Saturation** is determined as a simple inverse function of Intensity.

## 2. Algorithm results

By construction, the process applied to the integers sequence generates endless different images with increasing complexity, in terms of number of tiles, while preserving a common style.

In the resulting images succession, "deficient numbers" are bluish and "abundant numbers" are reddish; in particular prime numbers, made of one factor, have a flatter shape and a blue tone (see figure 4 for a sequence of 9 integers).

The artwork aims at capturing the unpredictability of the underlying integers' behavior: for example, the position of prime numbers is a very complex problem, as reflected in their sudden appearance as "flatter bluish" images.

*Figure 4: images of consecutive integers from 4065 to 4073*

As mentioned in the algorithm description, in addition to the variations produced by each integer, several parameters can be tuned to generate many different series, both in terms of geometry and of color distribution.

## 3. The artwork

The artwork consists of a video displaying a subrange of the infinite succession of integers. For this project, the set 4001-4100 has been selected in order to have a good balance between the complexity of displayed numbers (increasing with n) and the display resolution requirements (decreasing with n). Each integer image is presented for a few seconds.

It has been implemented through my own transformation algorithms, and built in Python within the Processing.org integrated development environment (IDE) [7].

## *Notes and references*

[1] for Integer factorization and RSA (Rivest–Shamir–Adleman) cryptography see for example https://en.wikipedia.org/wiki/Integer_factorization. An efficient factorization method is known for Quantum Computers: P.Shor, *arXiv:quant-ph/9508027.*

[2] the proposed metric assigns unit distance to consecutive vertex, and the fraction of the Euclidean distance of the 2 adjacent vertexes to the points in between. It is used, together with normalization of 3-side tiles to 4-side tiles, for obtaining an artistic effect with higher symmetry.

[3] G.H.Hardy, E.M.Wright, *Theory of Numbers*, 6$^{th}$ edition.

[4] in the artwork, Intensity and Hue are periodic functions of the "tile index" mod "product of selected factors".

[5] refer to [3] page 311 for "perfect numbers". The average Hue function is partially normalized in order to account that there are more deficient integers than abundant ones, see M.Deleglise, *Bounds for the Density of Abundant Integers.*

[6] an enhanced tessellation effect is achieved by applying also a "hue-shift" according to $T_i$ mod 2: differentiating tiles with odd/even tile index.

[7] "Processing IDE" is a graphical library for the electronic arts, new media art, and visual design communities; see https://processing.org/

## Keywords

Generativeart, Mathematics, Number theory, Computer Graphics, Digital Media