



UNIVERSIDADE DE  
COIMBRA

Eduarda Mendes Duarte

## VARIABLE IMAGE PATTERNS

**Dissertation in the context of the Master in Design and  
Multimedia, advised by Professor Tiago Filipe dos Santos Martins  
and by Professor Artur Luís Gonçalves de Azevedo Rebelo Alves  
and presented to Faculty of Sciences and Technology /  
Department of Informatics Engineering.**

September 2021



Master in Design and Multimedia  
Faculty of Sciences and Technology  
University of Coimbra  
September 2021

Eduarda Duarte  
Advised by Tiago Martins  
and Artur Rebelo

Variable

Image

Patterns



## Acknowledgments

To my parents, Carla and Duarte and my sister Magda,  
for the immeasurable love and support every single day;

To my grandmothers,  
for always keeping my heart warm;

To my advisers, professor Tiago and professor Artur,  
for the guidance and for encouraging me to be a better designer;

To Luís,  
for the all the kindness that seems impossible for one to hold;

To my second family, António, Daniela, Caio, Ana, André, Luís,  
for every conversation, opinion and smile;

To everyone that contributed and whose names I forgot to mention;

*Thank you from the bottom of my heart.*



## **Abstract**

For designers and artists, one of the most challenging tasks is to keep discovering how they can use the same techniques and components to different uses. A possible solution is to use modular systems as a tool to produce more complex results by looking at those familiar components but from new angles.

With the introduction of new programming languages and the rise of the new media, the problem posed is how to design a modular program capable of interpreting varied inputs and adapting them to new formats.

By researching the state of the art on image synthesis and the use of modules on computational systems, we understood a variable grid and varying shapes could depict any visual information.

Therefore, to answer the problem, we propose a website that accepts images, videos or live capture and transforms its cells into variable monochromatic tiles that represent a tone of grey. When seen from a distance, these compositions of tiles return optical illusions in which the user can recognise the input image, therefore resulting in variable mosaics. By varying the cell grid, the combination of tiles and their parameters, the user gets a wide pool of patterns to choose from.

Additionally, several functionalities are implemented in order to achieve a more flexible tool, such as a function to add text that has the same visual language as the images and a function to draw that provides a way to fully control the value represented by each tile in the grid. By having a system like this the visual parameters can be easily manipulated to represent images, text, drawings or just abstract patterns.

This approach seeks for a platform with a level of flexibility that allows the generation of a wide variety of visual artefacts. Therefore, its results can be used in different areas of Design, for example for visual identities, and also New Media Art, for the creation of mirror installations.

### **Keywords**

Modular Design  
Generative Design  
Image Synthesis  
Pattern





## Resumo

Para artistas e designers, uma das tarefas mais desafiantes é continuar a descobrir como usar as mesmas técnicas e componentes para diferentes usos. Uma possível solução é usar sistemas modulares para produzir resultados previsíveis ao olhar para estes componentes familiares de um novo ângulo.

Com o desenvolvimento de linguagens de programação e com a ascensão dos novos media, o problema é como criar um sistema modular capaz de interpretar diversos inputs e adaptá-los para novos formatos.

Pesquisando o estado da arte de síntese de imagem e de módulos em sistemas computacionais, percebeu-se como uma grelha variável e formas variáveis conseguia representar qualquer informação visual.

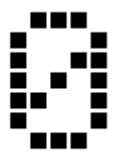
Então, para responder a esta questão, propomos o desenvolvimento de um website que aceita imagens, vídeos ou captura de imagem e transforma as suas células em módulos programáveis e monocromáticos capazes de representar um tom de cinza. Quando visto à distância, estas composições de módulos retornam uma ilusão óptica em que o utilizador consegue reconhecer a imagem de input às quais podemos categorizar como mosaicos variáveis. Ao variar a grelha, a combinação de módulos com diferentes módulos e parâmetros, o utilizador consegue uma amostra de padrões entre os quais escolher.

Adicionalmente, diversas funcionalidades são implementadas de forma a obter uma ferramenta mais flexível, como por exemplo uma função para escrever texto com a mesma linguagem visual que as imagens, e uma função para desenhar que fornece uma forma de controlar totalmente o valor representado por cada módulo da grelha. Ao ter um sistema assim, qualquer parâmetro visual pode ser manipulado para representar imagens, texto, desenhos ou padrões abstractos de diferentes maneiras.

Esta abordagem pretende obter uma plataforma com um nível de flexibilidade que permite a geração de diversos artefactos visuais. Desta forma, estes podem ser usados em diferentes áreas como Design, para criar diversos materiais para identidades visuais, e como New Media Art, para a criação de instalações de espelhos interativos.

### Palavras-chave

Design Modular  
Design Generativo  
Síntese de Imagem  
Padrão



# **Table of Contents**

<b>1. Introduction</b>	.....	<b>12</b>
Context		13
Objectives		14
Structure of the document		14
<b>2. Work Plan</b>		<b>16</b>
Methodology		17
Activities		18
<b>3. State of the Art:</b>	.....	<b>22</b>
<b>A Look into Images     and Modularity</b>		
History		23
Image Synthesis		36
New Media Art		41
Design		47
Typography		53
Final Thoughts		54
<b>4. Practical Work</b>	.....	<b>60</b>
Concept		61
Context		
Proposal		
Concepts and terms		
Development		64
Early prototype		
Generator		
Features		
Outputs		91
Evaluation		102
Applications and Dissemination		114
<b>5. Conclusions</b>	.....	<b>118</b>
<b>References</b>	.....	<b>122</b>
<b>Appendices</b>	.....	<b>128</b>
Appendice A		129
Appendice B		139
Appendice C		151

1 .

# Introduction

## 1.1 Context

Modular systems have always played an important role in graphic design. Designers have been using them to produce predictable results by looking at familiar systems from a new angle. Modularity presents itself as a special kind of constraint to the creative process, in which modules are fixed elements used within a larger system. They allow for an exponential number of possible combinations and results to be easily obtained. It is possible to find design processes that translate this concept to build typefaces, images and identities from blocks or modules (Lupton & Phillips 2008).

More recently, advances in new media and its technologies contributed to the development of programming languages for art and design. Following this, the demand for dynamic and generative design has increased as well as new opportunities for the designer to become a creator of tools. The rising popularity of creative coding enabled the development of these custom tools, opening a new way to use modular systems to solve and simplify complex problems (F. Pil & Cohen, 2006).

Some design projects create modular systems to generate visual compositions according to a concept. They can range from stylising images and linking them under a concept, like a graphic identity, to presenting a dichotomy between different perceptions of the same image. Other projects use these systems to develop modular typefaces that can be easily adapted to different environments. Additionally, new media art projects have also been building and applying similar modular digital tools from an artistic approach. These offer a new way to interpret and decompose the reflection or the interaction of the user to generate and show a new image.

Our proposal experiments with generative design to achieve novel visual languages. In particular, we explore the computational creation of images using a modular system of tiles capable of changing their shape to achieve different patterns. The variation in the shape and size of each tile helps to create the illusion of different shades of grey. By passing an image to the system, this last one translates it into a grid of tiles that are adapted to match the shades of the image, tricking the eye to recognise the input image in the generated digital mosaic. From here, we also explored the application of text and drawings in these compositions, which results in a tool that combines these different visual elements under one visual language.

The way this was applied was through creating an online website that allows users to interact with the system to obtain customised images or videos. The website can be accessed through the following URL:

[cdv.dei.uc.pt/2021/digitiles/](http://cdv.dei.uc.pt/2021/digitiles/)

## 1.2 Objectives

The presented project is placed in the areas of Modular Design and Generative Design and it consists of both a theoretical and practical approach. Abstractly, its purpose is to understand the limits and challenges that take place in producing a parameterizable and modular system with variable elements. Regarding the practical work, the goal is to develop a computational system capable of translating an image into a new version composed of tiles placed on a grid. The challenge in this resides in balancing the interdependence of the available parameters and components without eliminating opportunities. Although this could enhance the performance, it can also reduce the system's utility.

To better understand how to address these problems and to discover what relevant projects were already developed, one of the objectives was to research the historical evolution of methods to represent images from elements and collect existing projects in different areas. This research helped to make decisions regarding the final project and to refine its end goal. Therefore, it contributed to the constant development and evolution of the proposal. Lastly, a record of all of the processes, research and conclusions were assembled into this document.

## 1.3 Structure of the document

The remainder of this dissertation is structured in five chapters: Introduction, Work Plan, State of the Art, Practical Work and Conclusion.

The first chapter introduces the context, exposes the problem, and presents the project for this dissertation.

Following, the second chapter describes the methodology and model applied to organise the process and also the description of the activities that took place.

The third chapter focuses on the State of the Art, synthesising and analysing references and works related to the theme of this project. It begins by collecting historical context on techniques to represent images from small elements. Then, it analyses projects that apply modularity to image synthesis, namely in Design, New Media Art and finally Typography.

The fourth chapter, entitled Practical Work, presents the system developed in the context of this dissertation. It documents in detail its context, development, evaluation and applications. Throughout this chapter, we explain the decisions made and show the results.

Finally, the conclusion resumes the most relevant points of the document, reinforcing the results and answers obtained through the final version of the system.



2 .

# Work Plan



## 2.1 Methodology

In order to conduct the process most adequately, the project was conducted as a Second-order feedback loop model (Figure 1). This model encourages a constant refinement of the goals as a result of the constant feedback. Therefore, it prioritises continuous improvement and reevaluation. The goal of this model is to manage the work cyclically without assuming a constant and stable goal. In this case, a second loop is created in the process whose action is to evaluate and redefine the goal of the first loop. This first loop would focus on continuously developing and meeting the established requirements while the second loop would consist of periodical meetings, from 15 to 15 days to discuss if the project was heading in the desired direction, and if not, its goal and requirements would be redefined.

Besides these meetings, reflections were periodically done by evaluating the progress, comparing the actual state with the desired state and determining the difference. These involved meetings to discuss the state of the project, as well as answering a set of questions established in the beginning. This set of questions helped to track the changes in the project's concept and structure.

*What problem are we trying to solve?*

*What goal do we want to reach?*

*Why? What is the context that makes the objective relevant?*

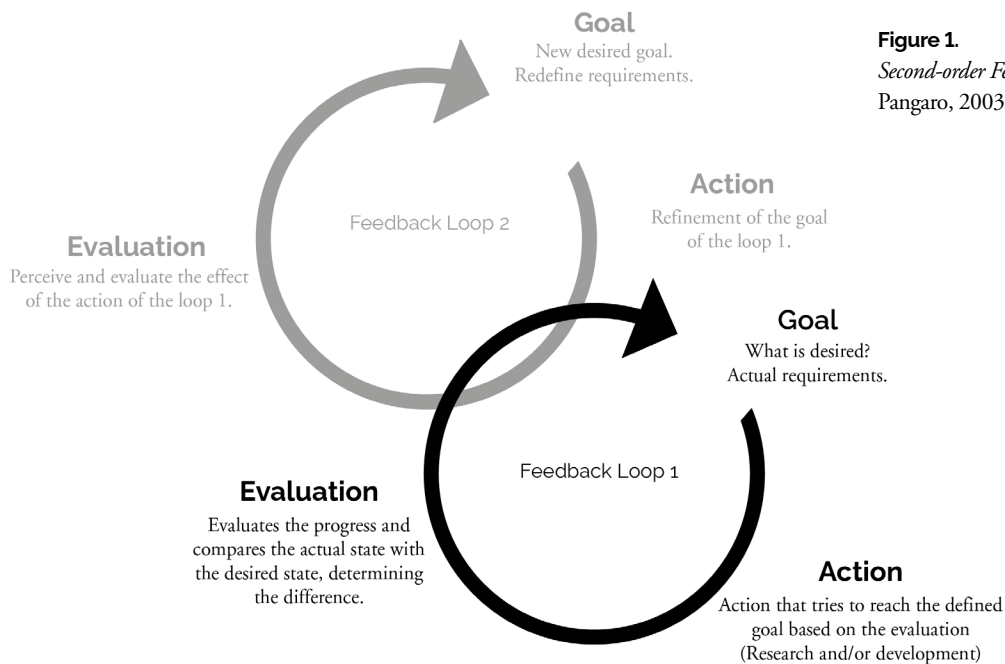
*How? What's the procedure to try to fix the problem?*

The answers progressively changed as more knowledge was introduced and our understanding of the system evolved: ideas would appear and influence the removal or addition of new requirements.

To manage the work, we used the method Kanban, which consists of having the tasks visible on a Kanban board separating them into three lists ("To do", "Doing", "Done"). The tasks are only pushed into the process when possible, rather than being pushed by request. In addition, we used the method *MoSCoW* to prioritize them by the categories "Must", "Should", "Could" and "Would" through colours. Through this, we managed to have a clear view of the backlog but also of what tasks should be done urgently. These methods would also give an idea of how far the project was from

being stable or meeting the goal: if too many tasks categorised as "Must" were still on the "To Do" board, we were not close.

The result of combining the *Second-order feedback loop* with the method of *Kanban* and the *MoSCoW* is an iterative process of adding, removing and changing tasks priorities, which ultimately lead to a stable version of the system that met the requirements.



**Figure 1.**  
*Second-order Feedback Loops,*  
Pangaro, 2003

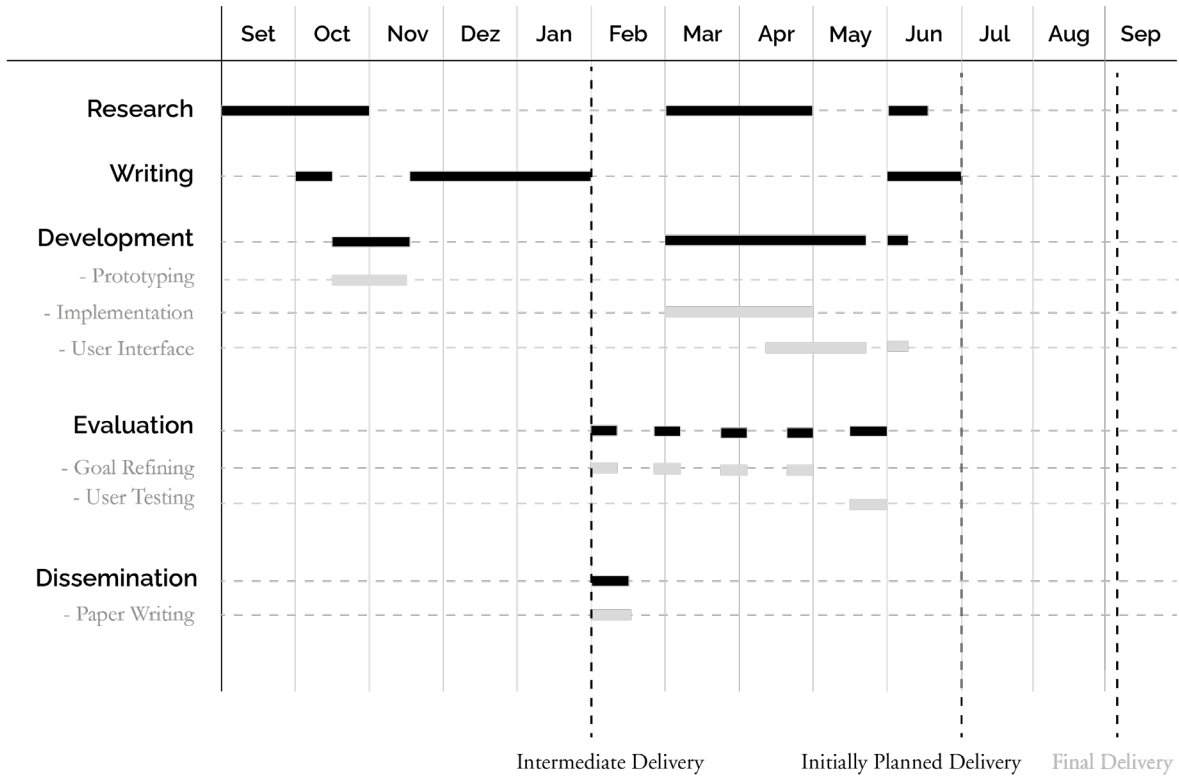
## 2.2 Activities

At the beginning of the project, a plan was created that would involve not only establishing activities but also estimating the needed time to finish them. These activities, as hinted in the previous chapter, did not follow a linear order, making it more important to properly trace a general plan in a Gantt Diagram. One diagram was mapped in the first phase (seen in Figure 2), while another one was mapped at the end of the project, reflecting how the events took place (Figure 3).

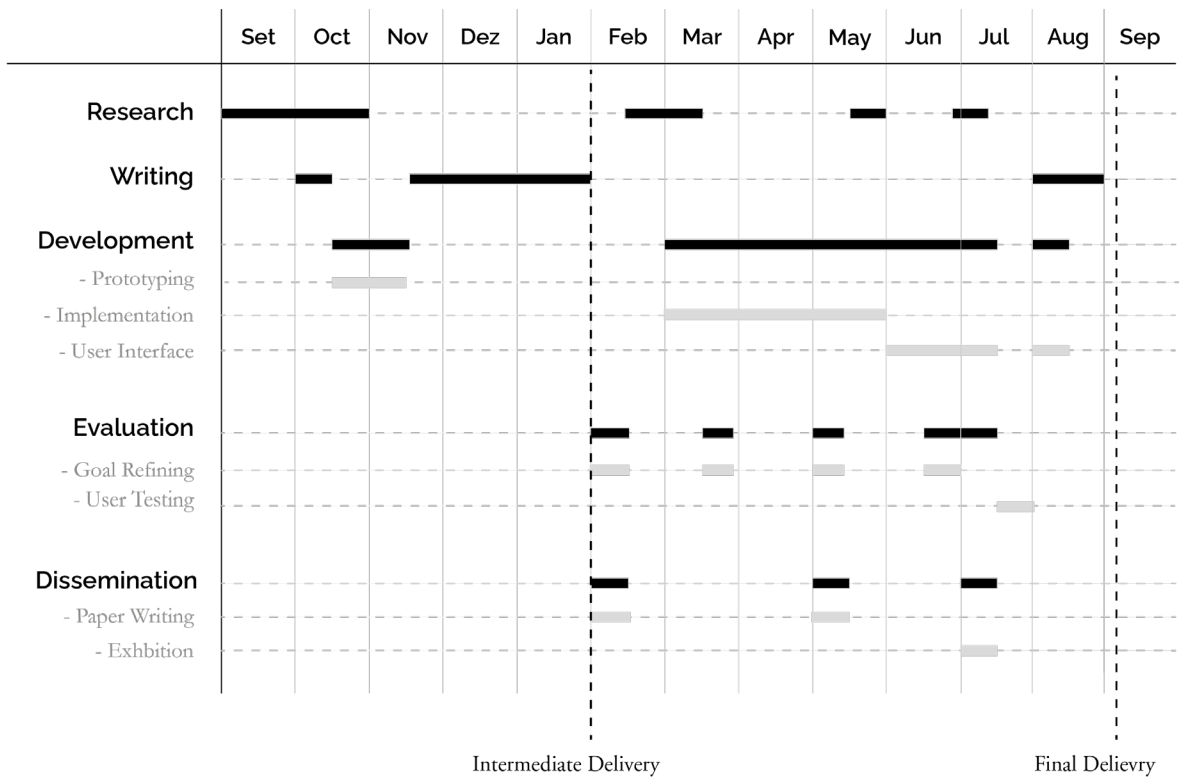
The differences between these diagrams reflect some errors in estimating the time for each task but also in estimating the complexity of the project. The project was delivered in September instead of July as initially planned, offering more time to invest in each task. This became a necessity

as the objectives for the proposal changed and more functionalities were introduced into the system. Another difference was in the dissemination, where new opportunities appeared. In order to better understand each activity pointed out in the diagrams, the following explanation is given.

- Research** This group of activities refers to the whole process that involved researching and collecting bibliographic references. It had a bigger incidence at the beginning of the project, however, it never stopped as new techniques and knowledge were constantly needed.
- Development** The development phase had the goal of experimenting and developing by putting into practice the acquired knowledge from the research phase. It is the phase that materialises the proposal and from which it is possible to obtain feedback.
- Evaluation** Groups all the activities involving reflection, testing and other forms of evaluation. In this case, a reflection was done periodically to check if the goal was redefined. Additionally, user tests were made at the end of the project.
- Writing** The written document encapsulates all of the work: it resumes all of the bibliographical research concentrating it in the "State of the Art" chapter and documents all of the developed work in the project.
- Dissemination** The dissemination of the project is done through the writing of two articles at two different times of the development of the project. The first one was written in February, related to the discoveries made until that time, and the last will be published when the project is completed. Additionally, an exhibition using the developed system took place in July.

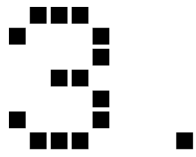


**Figure 2.**  
Gantt Diagram of tasks and events for the first phase and estimation for the second phase.



**Figure 3.**  
Gantt Diagram of tasks and events that took place in the entire project.





**State of Art:  
A Look into Image  
and Modularity**

The objective for this chapter was to document research done on the areas of Modular Design and Image Synthesis and also to analyse projects and artworks that are relevant to these themes.

It begins with a historical view of the evolution of techniques of image synthesis that can be seen as modular systems that use grid, monochromatism and shape variation to produce results. Not only this helped to uncover the origins and purpose, but also revealed how these evolved from the analogical to the digital world. It also showcases the evolution in the relations between technology and humans, and functionality and art.

Next, this chapter exposes the use of modular systems in specific areas namely Image synthesis, New Media Art, Design and Typography. Presenting different projects from each and analysing them, we understood what had already been done and gathered ideas for features and new techniques. Each one of these sections offers insight into the purposes, advantages and disadvantages of using modular systems through these widely different areas.

## 3.1 History

The capacity to produce and perceive different graphics presupposes an exclusive relation between language and the use of tools (Leroi-Gourhan). This new capability allows the human being to distinguish, register and communicate timelessly through graphics like images, videos symbols.

Nowadays, as we are constantly surrounded by all types of visual information, it is easy to forget how complex these systems are. The precious transition between analogical methods to the digital world was only achieved through a long path of creating and evolving new techniques of image synthesis.

In this section, some of these marks are highlighted by offering context about the history and functioning of these techniques. However, it is important to note that due to the extension of the information and details about each technique, the given context will be only the relevant and the necessary one in the context of this document.

## MOSAICS

The desire to create and represent a new visual artefact places questions about the process and the tools to choose since there were several options. The answers will condition the difficulty of the execution, the cost, the result and, consequently, the interpretation of the final image.

The first forms of representation, typically imagined, are drawing and painting. These techniques consist of adding pigment to a flat surface and require talent and time. However, it is of interest to analyse other options that bring economical advantages to the bidimensional representation: methods that allow the creation of visual artefacts, in an easily adaptable, economic and fast way.

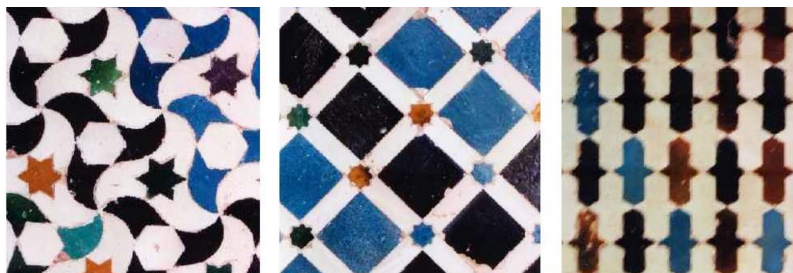
One of these options appeared in Ancient Mesopotamia, being later reproduced by the Ancient Romans and Greeks (Figure 4). This method would consist of, from smaller elements, making images, patterns or graphics called mosaics. These mosaics would be made from the incrustation of smaller pieces on a surface. In the past, these pieces would normally be stones (called tesserae) and they were used to decorate pavements, walls and other precious objects. Besides this, they were easy to maintain and economical (Cartwright, 2019).

The mosaics revealed not only a new artistic ability but also transported valuable representations and information about their cultures even before the appearance of written records (Dunbabin & Dunbabin, 1999).

This technique continued being used throughout many centuries and different cultures. For example, in Islamic architecture mosaics were used in religious buildings. Eventually, the technique suffered adaptations and similar effects were achieved through the use of painted tiles (Figure 5).



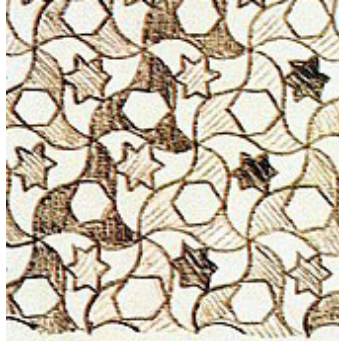
**Figure 4.**  
Roman Mosaic with geometric drawing, Tarentum, South of Italy. 2nd-century CE. (National Museum of Archeology of Taranto, Italy)



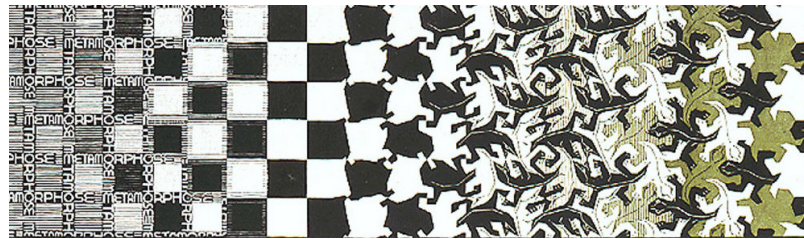
**Figure 5 .**  
Walls of mosaics of the Moorish palace Alhambra, built in 889 C.E. in Spain.



**Figure 6.**  
Escher's sketches  
of a tessellation in the  
walls of Alhambra in 1936.



**Figure 7.**  
Metamorphosis II, M.C. Escher,  
1940.



**Figure 8.**  
Portuguese pavement in Praça dos  
Restauradores, in Lisbon.



These historical remains influenced artists such as M.C. Escher, who was inspired by the Moorish tessellations he saw on the walls of Alhambra (a palace in southern Spain). Following this, the artist focused a part of his work on sketching studies (Figure 6) and later created works based on these tessellations such as Metamorphosis II (Figure 7) (Piller, n.d.).

Today, mosaics developed as a popular craft and its artisans use varied types of materials. Physical mosaics can be found in street art and house decoration. For example, we can find a kind of two-toned mosaic in Portuguese pavements made from irregular or regular stones which showcases the reliability of the technique (Figure 8). From this old method that still survives, the concept of tessellation and modularity in image synthesis was born.

## PRINT

Eventually, with the introduction of written records, the needs would change and the technology would accompany these changes. From the need to register and propagate information, either written or pictorial, the press evolved more rapidly from which several techniques were born.

Many of these techniques initially consisted of engravings, in which the block is sculpted by hand with the drawing to print. There were also Intaglio techniques, in which prints would result from having ink retained in the indentations of the sculpted block and then transferred to the paper through the application of pressure (Lechêne, n.d.).

The majority of images in journals were produced from engravings and, although, the attempt was to copy photographs, the sculpted images to print, would still look drawn by hand and unrealistic. Besides this, the techniques like these would only allow one colour at a time to be applied at a time (one colour per block), what would require the use of techniques such as stippling, hatching and cross-hatching that would produce the illusion of different tones and texture through the different distribution of black and white in the different zones of the depiction (Dickinson, 1972).

The commercial press needed a more practical and realistic way to reproduce photographs to the printed page. However, the most common processes of mechanical printing only allowed them to print uniform areas of ink or leave empty white areas. This meant that it was not possible to produce a range of continuous tones: only black ink (or another colour), or white space. Luckily, the process of halftone showed up and surpassed these limitations, becoming the norm for books, journals and other artefacts (Meggs & Purvis, 2016).

## HALFTONE

It was William Fox Talbot who, in 1842, for the first time, patented the idea of using photographic screens combined with the Intaglio technique. From there, this idea that a photographic print plaque could print the nuances found in a photograph if a screen would translate its tones to dots of varying size started to be explored. This way, the tones could be achieved even though it was a uniform application of the ink from the relief printer. Many researchers kept working on this problem and contributed to its evolution (Meggs & Purvis, 2016).

Later on, a big advancement occurred on March 4th of 1880, when the New York Daily Graphic published, in its journal, the first reproduction in large scale of a photograph with a complete tonal range. It was entitled "A Scene in Shantytown" (Figure 9) and it was printed from a halftone screen invented by Stephen H. Horgan.

This screen would break an image into a series of small dots whose sizes would vary, creating an illusion of tones. The intention was to trick the eye of the spectator to perceive different tones, due to the variation of the quantity of ink printed in each zone of the image. The range of tonalities

**Figure 9.**

*A Scene in Shantytown*, published on March 4th of 1880 in the journal *Daily Graphic*. The first reproduction of a printed photograph using a halftone instead of an engraving.



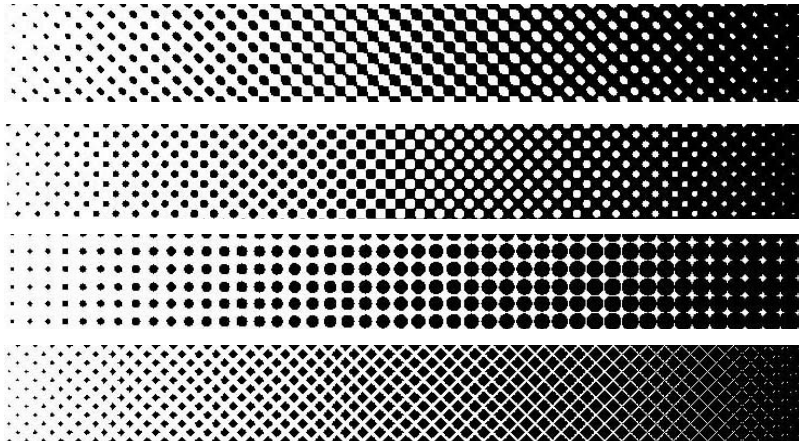
between the pure white of the paper and the solid black ink was, in this way, simulated through the quantity of ink printed in each zone (Meggs & Purvis, 2016).

Halftone is, therefore, defined as the process in which an image with a wide range of tones (continuous tones) is converted to dots purely white or purely black, meaning it is a binary depiction that is perceived by the human eye as being identical to the original (Ulichney, 1999).

This method evolved and introduced new adaptations, using different versions of dots as exemplified in Figure 10 (Petrulis, 2006). Later, eventually, this method evolved to what today is known as Digital Halftone.

**Figure 10.**

Demonstrations by Alan Petrulis of different versions of halftone dots (from bottom to top): Euclidean Dot, Elliptical Dot (Rhomboid), Round Dot and Square Dot.

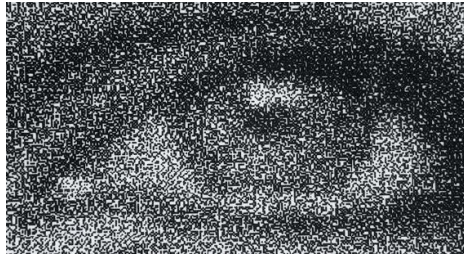


## DIGITAL HALFTONE

Halftone was a technique originally analogic; this meant that when translated to digital methods, new problems were raised on how to render the dots with the new digital tools (Benson, n.d.).

The solution was to use a device called imagesetter with a much higher resolution than the dots in the desired halftone. Nowadays the technique is used in different printers as well as in digital image devices that are based on pixels. Inclusive, around the '70s, laser printers began to be able to generate halftones, although initially, they had a very low resolution that would limit their use. However, their performance improved with the introduction of higher resolutions and the introduction of dithering techniques (Lau & Arce, 2018).

Digital Halftone is obtained from raster or bitmap images, in which each monochromatic unit of the image, or pixel, can be on or off, in other words, painted or empty. To emulate the original halftone dots, the resulting digital cell is composed of groups of monochromatic pixels. The proportion of the area of painted cells to the area of empty cells corresponds to the lightness, or the tonal value, of that cell (Lau & Arce, 2018).



**Figure 11.**

Richard and Tanya Donnelly, stochastic printing by Richard Benson in 1994, Museum of Modern Art, New York (full image on the left, close-up on the right).

Besides the traditional black and white, some printers are capable of printing in CMYK. However, they would still not be capable of reproducing continuous tones. For this, it would be necessary to do a quantisation of the values by translating an image with a continuous system to a system whose amplitude is limited to a fixed number of values (Lipshitz et al., 1992).

Dithering methods are frequently used to perfect the processing of halftone images and solve quantisation problems. These methods can be applied to a colour matrix with more than 2 values. There are different ways and variations of halftone patterns, whose goal is to help achieve photorealism or even create different stylisations.

One of these variations is the introduction of stochastic printing in which the dots are disposed on a grid with a random pattern, still covering the correct areas of the image to create the illusion (Figure 11). In other words, there is an introduction of noise that, as it will be seen next, enhances the illusion of these continuous tones, bringing humanity a step closer to photorealism (Benson, n.d.).

## DITHERING

When it comes to the image synthesis in digital format, it is hard to mention halftone without referencing dithering processes in the quantisation of tones and colours. While the halftone process involves transforming an image of the great amplitude of resolution into another one with a smaller amplitude, to obtain more faithful results, this process requires minimising mean squared errors that result from the quantisation of the values (Ulichney, 1999). The quantisation of the images in greyscale represents a scalar problem, especially when dealing with monochromatic transformations (Christou, 2008).

In the context of image representation, it happens that many times the display does not have available the range of tones necessary. For example, when it is pretended to simulate a photograph with millions of tonal values through video hardware that is only capable of representing two tonal values (Christou, 2008).

The problem is dealt with by repositioning the halftone dots by introducing noise to randomise the quantisation error that may occur (Furht, 2008). This process avoids the formation of patterns that would interfere with the similarity between the new artefact and its original. The result ends up, therefore, being more realistic and having more depth of luminosity (Ostromoukhov, 1999) (Figure 12 and 13).

It is important to understand that there exist several methods of dithering applications that vary in the probability of noise occurrence and distribution. This allows readapting the algorithm for different needs (Ulichney, 1999).

Lastly, regarding the process of rendering an image, from which the halftone and dithering processes are a part of it, the goal is to take the data of an image and adapt it to a certain display. Its phases are filtering and redimension, adjustments of colour (quantisation), application of dither and conversion of space and colour (Ulichney, 1999).

**Figure 12.** Comparison of images in halftone format, without application of dithering (left) and with (right). Example by Robert Ulichney in his paper.



**Figure 13.** Comparison of an image without application of dithering (left), in a web-safe colour palette, and with Floyd-Steinberg dithering in the same colour palette (right).



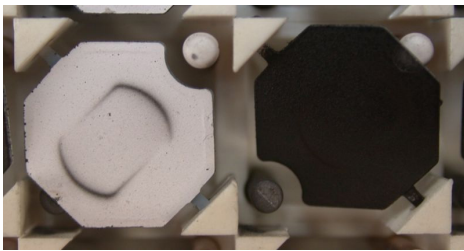
## FLIP-DISC DISPLAY

In parallel to the advances of digital techniques, other analogical formats continued to be developed from similar concepts applied to substantially different formats. Among them, there was the flip-disc.

The flip-disc display was developed by Kenyon Taylor at request from Trans-Canada Airlines while working for the Ferranti-Packard company. By the time that the product was patented, in 1961, the company would already have lost interest and it didn't consider the project interesting enough to further investment.

This display consisted of a grid of small metal disks that were black on one side and from a different colour on the other (typically white or yellow) (Figure 14). Its functioning is as follows: each disk is annexed to an axis with a small magnet attached. Close to the magnet, there is a solenoid that, when pulsed with the correct electrical polarity, the magnet in the axis aligns with the magnetic field rotating the disks. A computational system reads the information, typically characters and turns the disks that are necessary to produce the intended result (Flip-Disc Display Inventors, n.d.). Other versions were created, later on, introducing more devices like the Dot-LED.

Initially, these displays were expensive since the manufacturers had to sew each disk by hand. While the prices of their production decreased, due to a reorganization of the production, the technology was finding its place in highway signals and information systems of public transportations. Throughout Europe and America, some similar displays became popular in gasoline stations and, in 1974, Ferranti built other versions for buses and trains, making a huge profit (Liberatore, 2016). Although finding some flaws is pretty common (Figure 15), these displays require low effort and



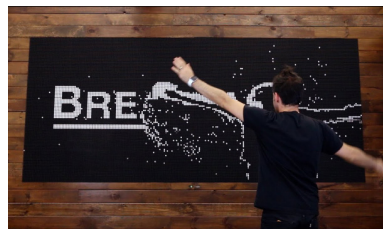
**Figure 14.**  
Close up of Flip-Disc elements.



**Figure 15.**  
Display with faulty dots.



**Figure 16.**  
BREAKFAST's display, sponsored on the studio's website.



**Figure 17.**  
Echo installation, by BREAKFAST in 2018.

cost for their maintenance (ALFAZETA, n.d.). Nowadays it is still possible to find newer versions of these devices in public transformation.

Around 2013, the technology was approached from a different light: the digital. Studios such as Non-grid and BREAKFAST took into account the popularity of kinetic devices and gave new life to the old flip-disc concept, making available their new systems of programmable Flip-Disc displays (Figure 16). These systems can be customised in many different ways and allow users to display videos, images or even have basic interactive experiences (Liberatore, 2016). BREAKFAST got orders from different companies and the studio itself created installations from the system as seen in Figure 17, inviting people to reflect on the relationship between the physical and the digital world (BREAKFAST, n.d.).

The truth is that all the technologies mentioned until this point do not allow the common individual to create and express himself without needing advanced knowledge of some sort about these tools. Having this into account, the following sections are about more accessible approaches using unexpected means. Like in many other situations in History, it is in this dichotomy between restriction and freedom that the human being can innovate and find its way of expressing through what he can access.

#### **TYPEWRITER ART**

Around 1868, with the growth of the popularity of the typewriter, a new form of a mechanical way of writing began to be available to a wider number of people. Its initial goal was to print characters on the paper by clicking individual keys. Its use was simple: a sheet of paper was introduced in the back of a cylinder. Next, it would be moved in a way that the sheet of paper would be around a ribbon with ink. When a letter was pressed on the keyboard, a bar of type would rise and hit the ribbon to then mark the glyph onto the paper (Tullett, 2014).

When the typewriter machine is used as a print device the situation becomes more curious. The same paper can be inserted several times, under different angles to print more than one time, layering different prints and composing a different image. The goal is to combine different printed letters into a tonal value, in an illusory representation of the shade range (Ridell, 1975).

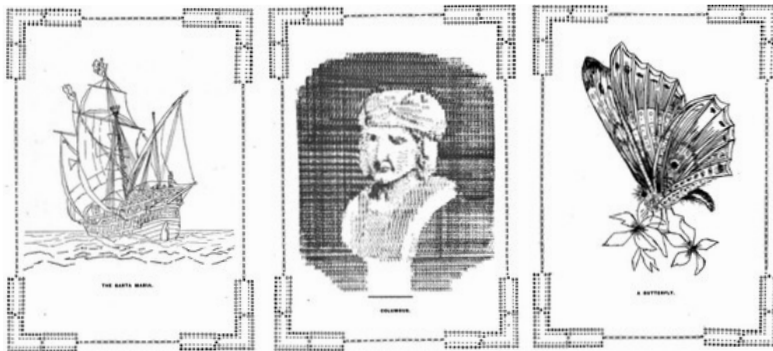
This form of representation brings special restrictions attached to the device. For example, the fact that the characters are monospaced, which allowed an individual to layer and align more easily different prints using always the same grid.

While overlays of punctuation and characters created the illusion of the presence of different tonal values, the use of the spacing would also work as a tool, as well as the variation of pressure applied to the keys (Tullett, 2014).

Typewriter Art was not initially well accepted and was even marginalised. In response to some journals that would publish about the "foolishness of attempting to make sketches by means of typewriters.", a

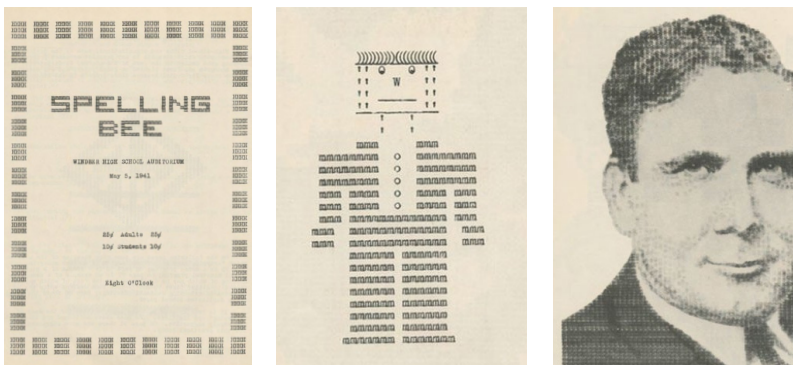
magazine called *Illustrated Photographic World* wrote: "We believe that any endeavour which will cultivate painstaking and accuracy on the part of operators should be encouraged...The endeavour to excel in artistic typewriting unquestionably does this. The pen maketh the exact man; so will the typewriter, which is only the modern pen.". Along with this message, they published three works of typewriter art and offered five dollars for other works of the same genre that were sent to them (Madrigal, 2014). Some examples of these works can be seen in Figure 18.

As the usage of the typewriter as the norm for writing dissipated, more artists began using this technology, now considered old-fashioned, to generate more pictorial representations. In 1939, an enthusiast named Julius Nelson, who was a dactylography instructor, published the book *Artyping*, an enormous collection of the best works of typewriter art at the time (Figure 19) where he also registered instructions and techniques (Nelson, 1939).



**Figure 18.**

Compositions of typewriter art created and sent by readers of the *Illustrated Photographic World* journal. They depict a ship (left), Christopher Columbus (centre) and a butterfly (right).



**Figure 19.**

Works from Julius Nelson's book *Artyping*. We can see ornamental frames (left), a man to whom Nelson called Type (acentre) and Wendel Wilkie (right).

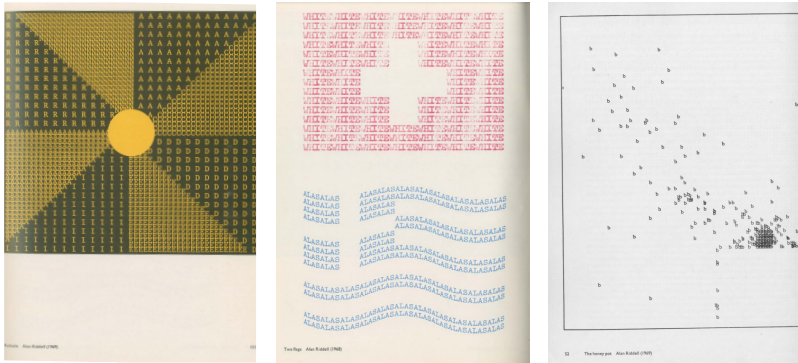
Around 1960, this form of art came back with concrete poetry which would be represented by an abstract and expressive print of typewriter characters (Ridell, 1975). By this time, the machines were relatively cheap, which helped this new form of art to grow more complex and sophisticated (Madrigal, 2014). In 1972, Alan Ridell published his book *Eclipse*, a collection of his concrete poetry (Figure 20). Later he also published *Typewriter Art*, choosing 119 works by 65 practitioners from 18 countries. He paid tribute to the machine and its particular qualities. Amongst these works, we can find some of his concrete poetry (Figure 21).



**Figure 20.**  
Concrete poems by Alan Ridell's  
from his book Eclipse, 1972.



**Figure 21.**  
Typewriter Art by Alan Ridell's from  
his book Typewriter Art, 1975



*"Seen from a distance, the hundreds of dots, in virtue of the visual phenomenon known as persistence of vision, coalesced into larger figures," applied mathematician Philip Davis wrote of Georges Seurat's pointillism... When in the 1880s typewriters became commonplace, this kind of image was done on the typewriter with letters or blank spaces, was known as typewriter art. In the first generation of computers, typewriter art was automated, and pictures of Washington, Lincoln, Harry Truman etc., were produced in this way. When computer output moved from the typed page to the television or video screen, the whole screen was subdivided into a certain large number, say  $1,024 \times 1,024 = 1,048,576$  areas or so-called 'pixels', each of which could be addressed, shaded, coloured or otherwise transformed or manipulated." (Ernest, 2003)*

In other words, the decomposition of images into dots or other types of marks is one more conceptual step to the creation of the pixel. In this timeline, the typewriter art is not only an ancestry of the ASCII art but also contributed to the concepts used today in displays and monitors. By welcoming a technological commonplace like the typewriter, we see how technology and art can continue to reinforce each other (Ridell, 1975).

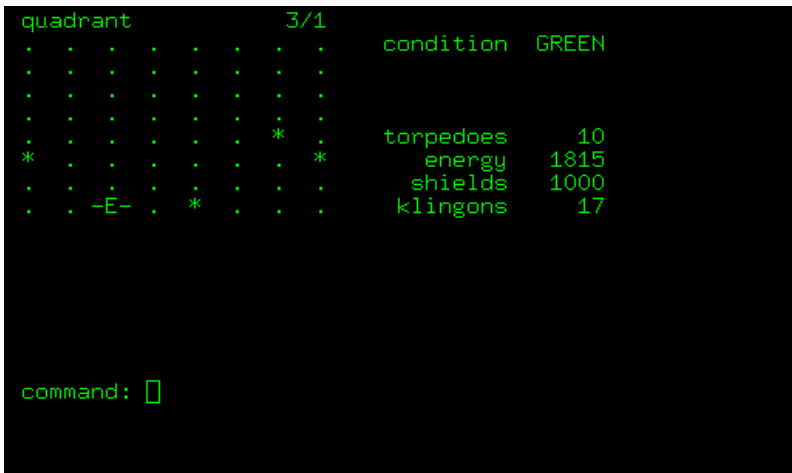
## ASCII ART

From the beginning of the communication using a computer, people used the available characters provided by the system to express themselves. From the traditional smiley face to digital portraits, it seems unavoidable for the human being to challenge himself through expression.

ASCII art represents the technique in which images or patterns are created from coded characters. ASCII is an abbreviation for the American Standard Code for Information Interchange that consists of a system of codification of characters used for electronic communication. The system was born from the need for a standard system for the communication industry during the '50s. Their development was taken under the care of the American Standards Association (Mackenzie, 1980).

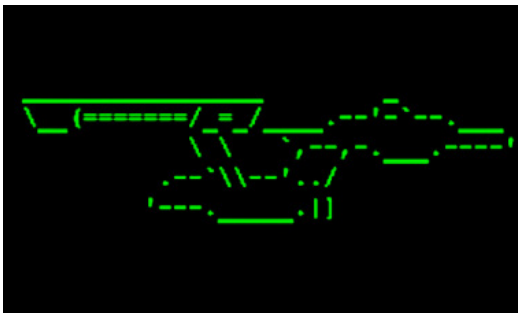
It was possible to find images composed of these coded characters in several software or even servers like *Bulletin Board System*, which allowed users to connect to a system through a terminal. *BBSs* were eventually abandoned after the emergence of internet connection, however, it would take a few years before the complete development of the web browser that would offer ways of combining text and images that would not consist in using just characters (Cassel, 2018).

The intent of this form of art was to create ways of communicating and expressing pictorially or to even create visuals for games. In some old computer games, for example, in the 1971 game *Star Trek*, the maps were composed of ASCII art, although very simple (Figure 22) (Smith, 2013).

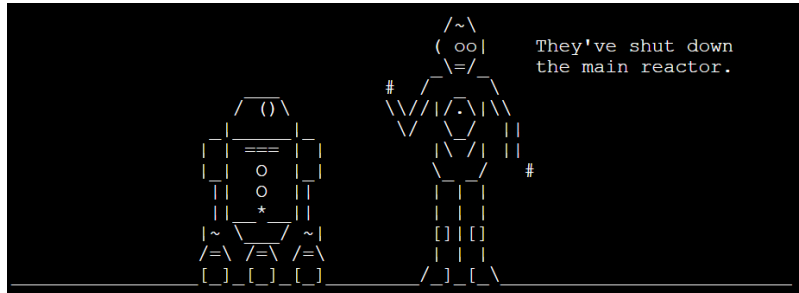


**Figure 22.**

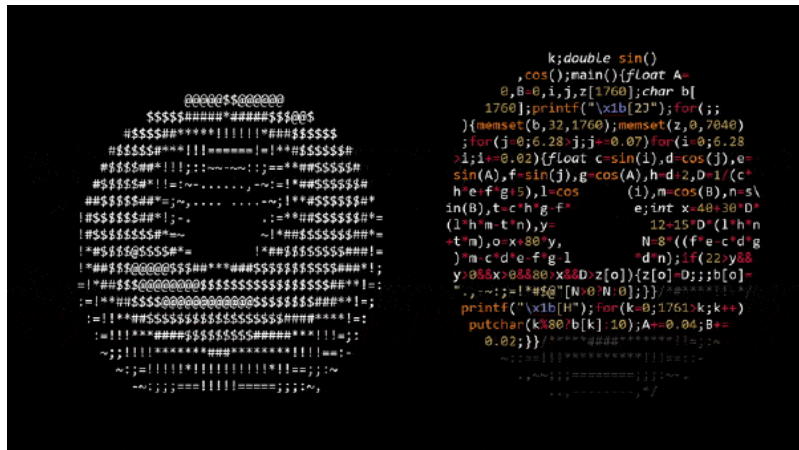
Computer game *Star Trek* from 1971. The graphics were created with ASCII coded characters.



**Figure 23.**  
*ASCIImation*: animation of the  
 movie *Star Wars: Episode IV* by  
 Simon Jansen in 1997.



**Figure 24**  
 Animated rotating Donuts with  
 ASCII characters created by Andy  
 Sloane in 2011



Other older projects using the same imagery revealed clues for potential future uses of graphics. For example, *cowsay*, a program written in Perl, in 1999, by Tony Monroe would generate figures with ASCII of a cow with a speech balloon with a customisable message. Another example is the *Star Wars ASCIIMATION*, started in 1997 by Simon Jansen, that consists of stop-motion animation using ASCII art from the movie *Star Wars: Episode IV - A New Hope* (Jansen, 1997). It is possible to see one of the hundreds of frames in Figure 23.

Although achieving the peak of popularity before the introduction of the web, ASCII art is still being explored under different artistic contexts (Figure 24). Inclusive, some programs receive an input image and transform it into ASCII patterns, providing results with just a few clicks. From this wave, there were also other similar trends like *Shift\_JIS* art which was the Japanese version, which created an immense success and started a cultural phenomenon in Japan.

Besides some of these examples being peculiar, they demonstrate the amazement of using available modules or pieces to transform and create new graphics.

## 3.2 Image Synthesis

The rapid evolution and success of photorealistic computer graphics at the beginning of the 90's opened new artistic possibilities. New techniques of non-photorealistic rendering started being developed. Two decades later, the field of non-photorealistic rendering expanded significantly, covering several expressive styles for visual communication (Kyprianidis et al., 2013).

### 3.2.1 Non-Photorealistic Rendering

Although the term Non-Photorealistic Rendering is a term normally used in the Graphic Computation subject, it is hard to describe exactly what it is, causing disagreement between investigators (Schofield, 1994). It becomes easier to describe NPR for what it doesn't represent: a way of generating something that aspires to reality. In other words, it is any type of graphic computation of images that tries to get away from the paradigm of photorealism (Gooch & Gooch, 2001).

More controversial definitions define NPR as being the subject of graphical computation that focuses on creating expressively stylised digital art. For historical reasons, the tendency of these systems is, in fact, to generate images that appear to be made by hand, like drawing with pen, pencil, aquarelle or even charcoal. Several of these computational algorithms, today, are capable of simulating a great variety of artistic styles and techniques (Schofield, 1994).

The most common classifications of NPR techniques are stroke-based rendering, example-based, region-based techniques, image processing and filtering, region tiling and tonal depiction (Kyprianidis et al., 2013). By taking into account the context of this dissertation, it will explore projects that fit the category of region tiling and mosaicking (Hausner, 2002).

Although a lot of times NPR has the intent of originating images with an artistic style, the area focuses on a scientific and mathematical approach. However, as seen in typewriter art and ASCII art, there were also pioneers in the world of Computer Art, using these techniques to approach a new form of art. The examples that follow sit between these concepts, showcasing mathematical approaches but also appropriating and recreating art.

#### IMAGE MOSAICS

In 1998, Adam Finkelstein and Marisa Range, created a process for creating image mosaics. These consist of a groups of small images arranged in a way that when they are seen from a distance they suggest a larger image. To achieve this, the small ones are arranged to match the large picture, and then their colours are adjusted to better suggest the overall form.

This ability to combine groups of pictures affords opportunities for the aesthetic and associational juxtaposition of images. Furthermore, there are other potential applications for this system such as encoding extra information in an image for transmission or security, new forms of halftoning screens for printing, and association of images for advertising.

Their process is similar to an artistic screening technique developed by Ostromoukhov and Hersch, where variations in brightness across the larger image are produced by varying the sizes and shapes of tiny subjects or characters. These small figures are described by closed contours separating black from white. In contrast, Finkelstein and Range's method varies the brightness of tile images composed of shades of grey, so that the resulting mosaic is a grayscale image.

In the documentation of the project, they remark how some images work better and remark how recognition plays an important role. For example, iconic figures such as actors, famous works of art, and well-known scenes work better (Figure 25). They also add that colour distribution in an image can also influence the quality of the result. Tile images with relatively uniform distributions of brightness tend to be easier to identify when their colours are colour corrected (Finkelstein & Range, 1998).

**Figure 25.**  
Result of Image Mosaics with  
the famous Mona Lisa  
(using three layers).



### JIGSAW IMAGE MOSAICS

Inspired by Arcimboldo art, Kim and Pellacini introduced a technique to form mosaics where images and random shapes were used to compose the final figure. The idea was similar to a photomosaic, where the mosaic is composed of images as its individual modules.

The algorithm had three phases: first, the process would choose and position the modules, ignoring deformations, secondly, it would refine and reposition and warp the module if necessary and finally, it would assemble the final mosaic, by positioning each module in its position and fitting each warped image in the corresponding module. The method that was used produced nice results and marked advancement in tiling algorithms (Figure 26) (Pellacini & Kim, 2002).



**Figure 26.**

Results of Jigsaw Image Mosaics demonstrated by the authors of the project.



### HALFTONE QR CODES

This next project applies the halftone technique in a new context. In 2013, Hung-Ku Chu, Chia-Sheng Chang, Ruen-Rone Lee and Niloy J. Mitra presented an automatic algorithm to create a new type of visual QR code to which they called halftone QR code. The algorithm combines halftone images with QR codes with the intent of producing high-quality visual QR codes that are still machine-readable (Figure 27).

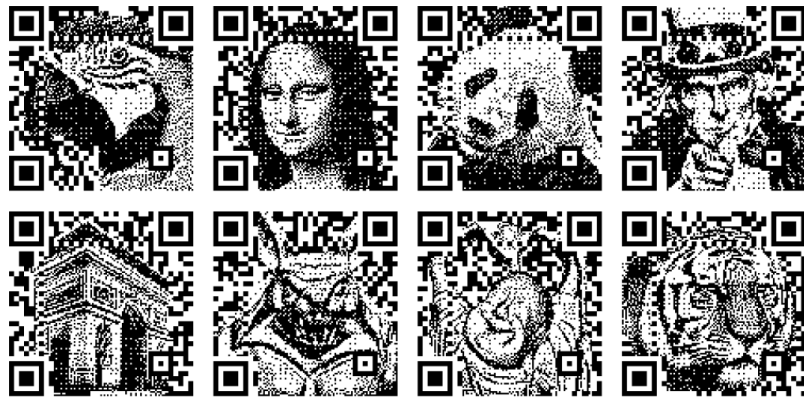
In their approach, they subdivide each module into  $3 \times 3$  submodules and bind the module's colour to the centre submodule, which leaves the remaining eight submodules free to independently change their appearance. They use this flexibility to characterise the appearance of modules using a set of binary patterns.

The main challenge lies in how to incorporate high-level visual features (e.g. colours, letters, illustrations, logos, etc.) with ordinary QR codes

without compromising their readability. There are two major components in their approach. The first one is a representation model that minimally binds to the original QR codes and is flexible to adapt to target halftone images, and the other is an introduction of pattern reliability which enables the control over the level of readability.

They also point out that the quality of the results is restricted by the halftoning technique which might produce low-quality images due to the low image resolution and contrast (Figure 28). Although they could have used a subdivision model to increase the resolution, it would exponentially increase the number of binary patterns. Besides, the readability of the results is inherently sensitive to blurred images caused by image downsampling. The centre pixel of the pattern which carries the original data module would be polluted and damaged by its neighbouring pixels (Chu et al., 2013).

**Figure 27.**  
Results of Halftone QR Code  
with different images.



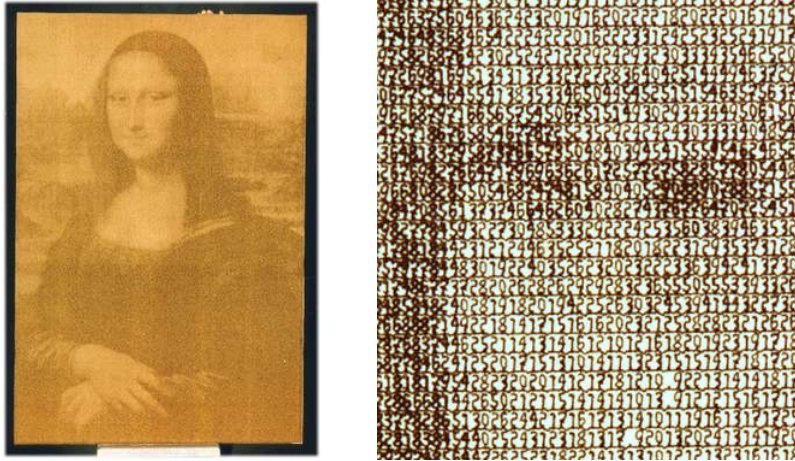
**Figure 28.**  
Results of the Halftone QR  
Code for the same image using  
different levels.



### DIGITAL MONA LISA

In 1965, H. Philip Peterson used a computer CDC 3200 and a scanner to create a digital transposition of the famous painting *Mona Lisa* by Leonardo da Vinci. This work is known as the *Digital Mona Lisa* or *Mona By the Numbers*, due to the technique used for its creation (Digital Mona Lisa | Database of Digital Art, n.d.).

Peterson created a translation of the originally analogic representation to a digital one. The idea was to consider an original painting as a grid of tiny colourful dots (now known as pixels). However, the colours would be reduced only to grayscale values. These were obtained from a Calcomp



**Figure 29.**  
*Digital Mona Lisa or Mona By The Numbers*, by H. Philip Peterson, complete image (left) and detail (right).

plotter using numerical characters, some overlaid so that they would create an optical illusion caused by the variation of density in each zone of the image (Figure 29). The production's process took 14 hours to complete the image of 100.000 pixels.

The fascination for this piece comes from the contrast between the technology available at the time and the result itself. It represents, without a doubt, a pinnacle landmark of the digital transposition of a scanned photograph to this computational output. (Patros, n.d.).

### **KEN KNOWLTON'S MOSAICS**

Among the most relevant artists of computational art, we can also find Kenneth C. Knowlton: a pioneer of graphical computation, mosaicist and portraitist (Knowlton, n.d.-a).

In 1966, he started to experiment with photographic mosaics along with Leon Harmon, creating prints from collections of small symbols and images. One year later, the two created the *Studies in Perception I*, also known as the *Computer Nude*: an image, composed of small elements, portraying a nude of a woman (Figure 30). This image was obtained through scanning a photograph originally taken with a camera. The image was mapped to binary numbers to which were then attributed a typographic symbol based on halftone densities, reproducing a similar effect to halftone (Albright-Knox, 2018).

The result was printed by The New York Times on October 11th of 1967 and was displayed in one of the first exhibitions of computational art entitled *The Machine as Seen at the End of the Mechanical Age*, in the Museum of Modern Art in New York, in 1968.

Besides this work, Knowlton created various images out of others, calling them, inclusive, mosaics. The old tesserae that constituted these mosaics were not rocks, but images that portrayed different objects, like seashells, domino pieces, dices, letters, cars, symbols (Figure 31) or other imaginable options (Knowlton, n.d.-b).

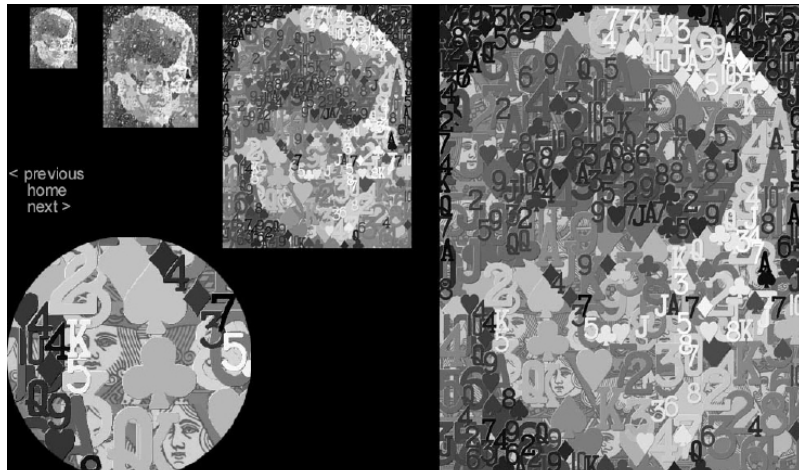


**Figure 30.**

*Studies in Perception I: Computer Nude*, by Ken Knowlton and Leon Harmon in 1966. Entire image (left) and detail (right).

**Figure 31.**

Lennard Green portrait, card magician. Collage made computationally using cards' symbols, by Ken Knowlton in 2002.



### 3.3 New Media Art

There are also varied ways to explore the concept of modularity in the area of digital art. This area is designed entirely by the experience of its viewing or use. It can test the possibilities and constraints of digital design by failing or succeeding on the strength of its interface. (Bolter & Gromala, 2003).

This form of art shows that the computer is not becoming invisible in our cultures, as the electric motor did. Until today, computers still fascinate us and digital art explores this fascination. In our culture, the computer and related digital technologies are media technologies and support a range of new media forms. We use them to read, draw, watch movies, and listen to music. Their look and feel is a part of the experiences they provide, therefore, they are not meant to disappear from our consciousness. Instead, one way to see this is that rather than operating computers, we interact with them, and similarly, digital artefacts are designed to be experienced instead of being simply used.

*“We continue to be exhilarated by and sometimes frightened of what digital technology is doing, and we are tantalized by the prospect of what it might do in the future. It is the task of digital art to fascinate, exhilarate, and sometimes provoke us. Appliances, on the other hand, don't fascinate us; they brown our toast.”* (Bolter & Gromala, 2003)

The best digital art consists of experiments that can help us figure out how to design for multiplicity since this art can adapt itself to the user rather than forcing him to follow a rigid script. They can dive into the relationship between themselves and its user, therefore they are both mirrors and windows. Digital interfaces can be like mirrors in the sense that they reflect the user in the context (their physical surroundings, immediate working or home and culture). The most interesting interfaces will make the user aware of their contexts and in the process, redefine them. Mirror installations take a direct approach to this: they both reflect and redefine the reflection. In this sense, their designers create an interface that is transparent and reflexive. Transparent as they try to create this illusion and spectacle of different technologies and strategies for the user, and reflexive in the sense that they offer a different way to approach an image that makes the user aware (Bolter & Gromala, 2003). To reach multiplicity and adaptability in a mirror, some artists looked for answers in modular systems.

### 3.3.1 Mirrors

The projects that follow are a collection of interactive mirrors that reflect the spectator or that interpret his actions using only modules. The use of modularity in this context helps to simplify the programs and their understandings of things, preparing them to deal with cells instead of a whole image. These installations change the usual role of the spectator of a work of art and raise questions about authorship since the spectator has control over the content that shows.

#### DANIEL ROZIN'S MIRRORS

Daniel Rozin is an artist, educator and developer working in the field of digital interactive art. As an interactive artist, Rozin has created installations and sculptures with the ability to adapt and respond to the presence and perspective of the spectator. In some of their works, the individual becomes the content while in others, he becomes the guest taking an active role in the creation of the final work.

His work splits into mechanical mirrors, software, kinetic sculptures, proxi prints and painting with video (Rozin, n.d.). Although computers are typically used in his work, it is rare that they are actually visible.

As many artists and designers referred to in this document, Rozin has an enormous portfolio of works that would be interesting to analyse, each one with particularities that would be relevant to mention. However, two stand out and these will be the ones analysed: one entitled *Wooden Mirror* and the other, *Circles Mirror*, both consisting of mechanical mirrors.

The mirrors contain video cameras, motors and computers that share their behaviour: any person that stands in front of the mirror is immediately

reflected in its surface, producing, additionally, a soothing sound according to the interaction (Rozin, n.d.).

His project *Wooden Mirror* dates from 1999 and represents its first mechanical mirror (Figure 32): a work that explores the line between the digital and the physical, through the use of natural materials, like wood, with the intent of abstractly portraying the notion of digital pixels.

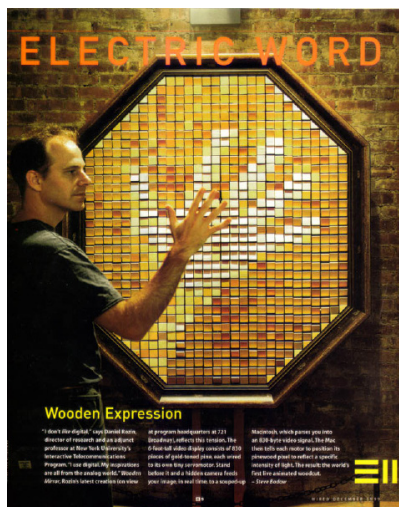
Its display consists of 830 pieces of gold pine wood, each one connected to a small servo motor. When in the presence of someone, the hidden camera feeds the image to the software which, in real-time, processes it and indicates to each motor the rotation that each piece of wood must perform in order to reflect a specific luminosity of the image. At the same time, it plays with the shadows to achieve this luminosity, resulting in the first work made of animated wood that, inclusive, raised the attention of the media (Figure 33) (Rozin, 1999).

As its name indicates, *Wooden Mirror* is a paradox. It is a mirror made of an opaque material, and yet it reflects. It is a digital artefact, and yet it forms its image through a delicate mechanism. The project is also a reminder that a mirror is a surface and not a window onto a different world (Bolter & Gromala, 2003).

**Figure 32.**  
*Wooden Mirror* Installation in the  
Museum of Israel.



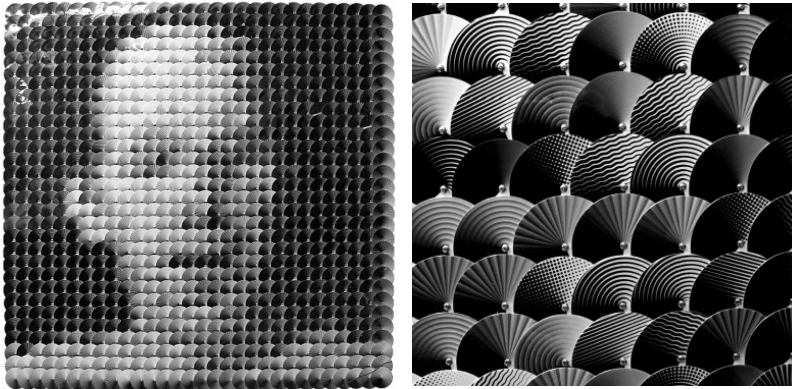
**Figure 33.**  
Article about the *Wooden Mirror*, in  
Wired magazine.



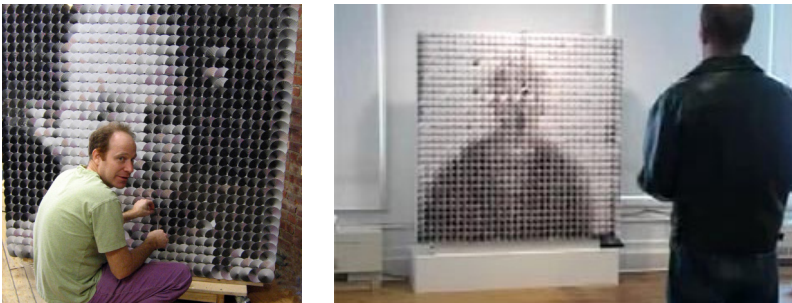
With a similar concept, his project *Circles Mirror*, from 2005, has a different format: each piece rotates to change the luminosity, however, this one is obtained from the quantity of white and black in the pattern that is printed in the part of the piece that is visible at the moment, and not from the shadow of the physical object, as seen before.

Using 900 overlaid circles, that are laminated and digitally printed, and a video camera, a computer and customised software, Rozin created the *Circles Mirror*: a mechanical structure that, again, reflects the person who stands in front of the piece (Figure 34).

Printed in these circles exist twelve different patterns that represent gradations between black and white. The circles are randomly placed on the board and rotate to expose a whiter or darker pattern, according to the zone of the image they are trying to assimilate (Figure 35). The patterns are also monochromatic, making only use of pure application of black and white to generate an illusion that there are different tones (Rozin, 2005). This piece is an interesting combination of the use of graphics and computational and mechanical work.



**Figure 34.**  
*Circles Mirror* reflecting the image of the spectator (left) and close-up of the patterns (right).

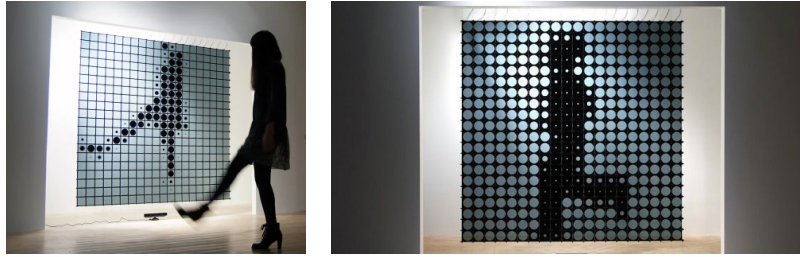


**Figure 35.**  
Interaction with the *Circles Mirror* reflecting Rozin's image.

## IRIS

The Korean studio Hybe is responsible for the next presented project. Entitled *Iris*, the project consists of a captivating display defined by his minimalism. Still, within the same format of a modular mirror, this project creates images from the darkness, instead of light, using LCD's to create these complex works of what it resembles pixel art (Figure 36).

**Figure 36.**  
*Iris* installation made by the Korean studio Hybe, in 2013 for the Da Vinci Idea exhibition.



Using a grid of LCD's, the design of this installation brought practical advantages besides esthetics. “Many people believe that LED is very green technology with great brightness and low power consumption,” Han explains. “It is true for a single LED module but is absolutely false when it becomes a media canvas with thousands of them” (Wilson et al., 2012).

Internally, the design of this LCD counts with the ambient light to offer contrast, instead of light-emitting diodes. The squares with 3,5 inches use about 90 microWatts. As a reference point, a display like this that measures 27x3 meters in total can operate with a laptop battery. The cycles of animation are smooth, the mirror is inviting and, although the LCD technology is not a novelty, the studio innovates the concept (Hybe, n.d.).

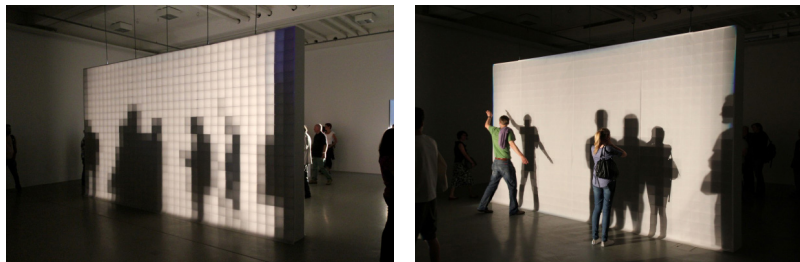
Upon analysis, it is visible how the work takes advantage of the old halftone concept, where monochromatic modules (in this case it is ellipses) of varying sizes are disposed of on a grid. In this case, the intent is not to represent an image accurately and with quality, but to react and respond to an interaction with the system from which the spectator can recognise the information as to his image and create an understanding of the work's pattern of behaviour.

#### 0,16

The project *0,16*, by Aram Bartholl, dates from 2009 and it is a light installation in which the shadows of the individuals that pass through it are transformed into pixels. The name of the project is due to the resolution of the screen of the installation, which is 0,16 ppi (pixels per inch).

The installation is formed by little squared frames covered by transparent paper, in the front and in the back, and the third layer of paper is placed in the centre of the frames (Figure 37).

**Figure 37.**  
*0,16* Installation exhibiting in Raum Schroth im Museum Wilhelm Morgner, in Germany in 2019. It is possible to see the generated image (left) and the input and interaction by the spectators (right).



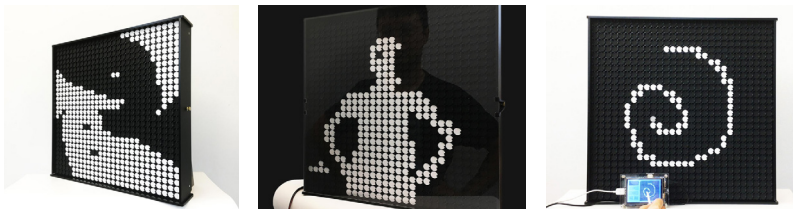
When it comes to this work, the concept of processing and quantising an analogical input with continuous tones to another fixed range of values becomes more obvious. The result resembles binary pixels, a low resolution and digital way of seeing reality. The majority of Bartholl's installations have this relationship between the digital space and day-to-day life as a theme (highlike.org, n.d.).

*"In which form does the network data world manifest itself in our everyday life? What returns from cyberspace into physical space? How do digital innovations influence our daily actions?"-Aram Bartholl*

### INTERACTIVE FLIP-DOT DISPLAY

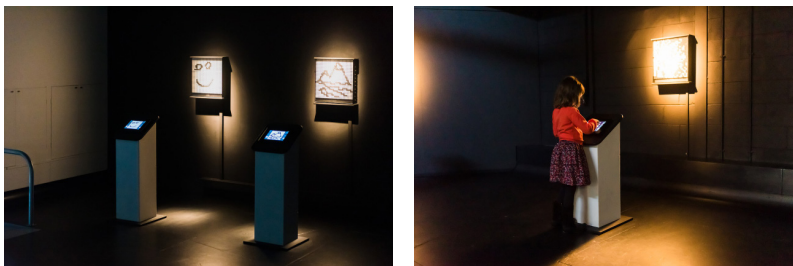
The *Interactive Flip-Dot Display* is an artistic and kinetic installation created by the studio Think Create, in 2018. The installation has three modes: the first one exhibits videos, static images and text that are fed as input, the second one is interactive and it tracks in real-time the body that is standing in front of the device, with the help of a 3D depth camera. Lastly, the third mode allows the user to draw, also in real-time, through the touch screen included in the installation (Figure 38).

The size of the installation is also customisable (Figure 39). This is possible due to the fact that the device is modular and allows it to vary between displaying in one unit or hundreds of units, consequently altering the detail of the images (Thinks Create, 2018).



**Figure 38.**

The three modes of the installation *Interactive Flip-Dot Display*: image representation (left), a reflection of the spectator (centre) and drawing (right).



**Figure 39.**

*Interactive Flip-Dot Display* installation, exhibiting in the event Shapeshift in the Eureka! The National Children's Museum in Halifax, United Kingdom, in September of 2018.

## 3.4 Design

Typewriter art, ASCII art and computer art reveal a logic that manifests with the introduction of new technologies that is their use as an individual artistic expression and their use under different contexts that were not initially planned.

The intent of the section is to analyse how these systems can contribute to graphic design and art by changing and adapting images to any different contexts by changing its pattern.

### 3.4.1 Dynamic Compositions

Several contemporary designers explore this dynamic of using the same pieces in order to obtain a wide range of results. Different to some of the examples seen before, there is a focus on the graphic value of the modules, distancing even further from a realistic and simple image to new artefacts with layers of expression.

#### THE WORLD OF KAREL MARTENS

Karel Martens is a Dutch designer whose work expands over 50 years. Looking at his collection of works, which vary between stamps, signals, murals and books, it is perceptible his mastery of print, colours, mathematics and language. His role oftentimes changes to that of an artist, experimenting with what he has in his reach (Ong, 2020). Balancing freedom and order, he finds inspiration in the limitations, turning obstacles into challenges (Bil'ak, 2004).

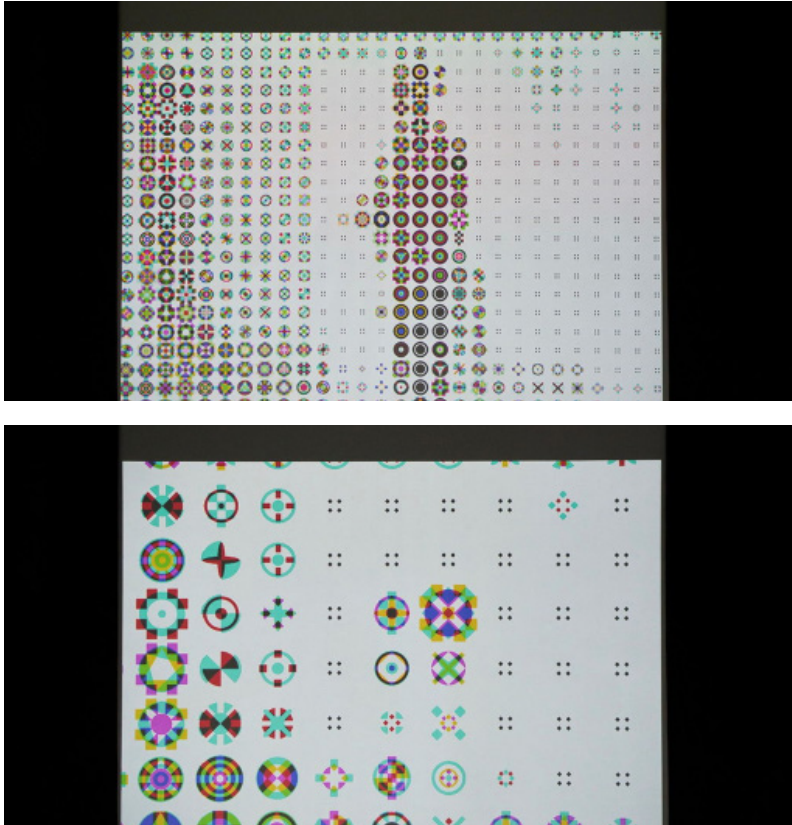
*“I discovered everything through reflecting. There can be a lot of rules in the game (of design), but you have to order them and kind of sort them out. I think limitation is an important thing.”*

- Karel Martens in an interview for the magazine *It is Nice that*, 2020

Regarding his approach to the profession of designer, he highlights the importance of having restrictions in his creative process and also of trying to reduce the number of elements in the projects, to only the necessary ones.

Among the many works of the designer, there is a selection where the use of modularity and grid are strongly highlighted in order to represent images under different contexts. His works reveal graphical explorations on how to include two readings in one image only by using small symbols to portray bigger pictures.

More recently, he applied technology in his work obtaining different geometric and kinetic creations, which is the case of his installation *Icon*



**Figure 40.**  
*Icon Viewer*, an installation by Karel Martens in 2015.

Viewer (Figure 40), whose resulting images were later transposed to the paper (Ong, 2020). In this way, through the play of materials and methods, Martens became one of the most influential contemporary designers.

Due to this design approach, he becomes a very important reference to this work: from his way of thinking to the relations he creates between grid, shape, colour and identity. To exemplify these characteristics of the designer, we analysed his work *Dutch Clouds* (Martens and Martens, n.d.).

The project is a mural, dated from 2011 and has a different approach from the traditional mural. When seen from afar, one sees a normal image of clouds, when from up close, it is clear that the image is composed of modules with their own expressivity and that gain stronger attention. This duality maintains the design dynamic. The modules were developed by the designer as a way to combine and obtain different tones from different combinations of coloured shapes. They reflect the mastery that the designer has with monoprint and overlays that he uses to create this illusionary perception experience (Figure 41).

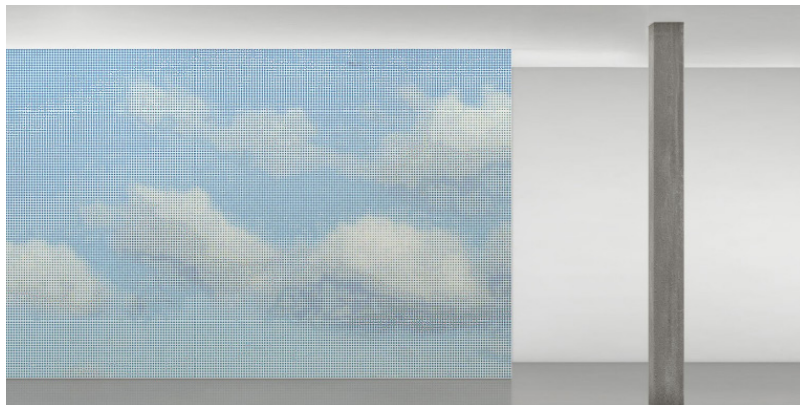
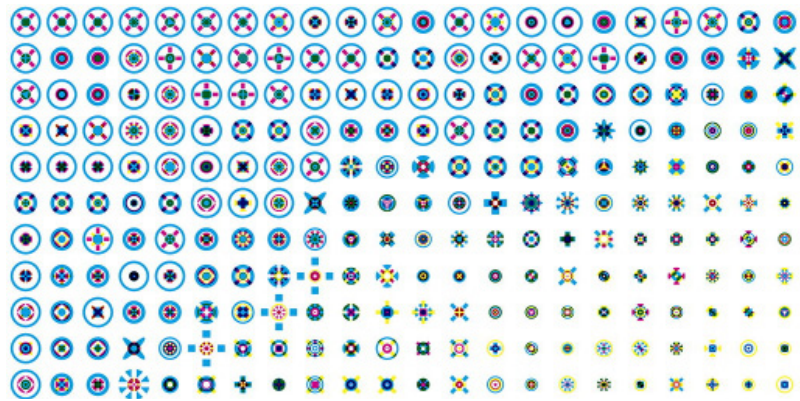
The ruling and stylisation of the modules in *Dutch Clouds* are different: their expressivity is so strong and distinct. Besides this, the final results gain a new relevance when remembered that from the same system it is now possible to simulate any image.

*“How many different elements do you need to tell the story?”*

-Karel Martens in interview with Peter Bil’ak, in 2004



**Figure 41.**  
*Dutch Clouds* by Karel Martens.  
 An image generated with symbols,  
 Maharam, Nova Iorque, 2011.



### GSYIN \$ VANNETI

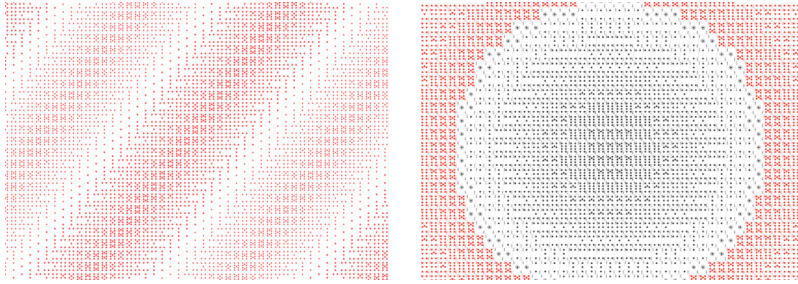
Nowadays, the studio *ertdgvb*, also known as Gsyin and Vanetti explores a lot of the themes that were seen so far, becoming a very rich and relevant referring thanks to their variety of projects that make use of dithering, pixel modulation and other concepts to create image, patterns, interactive mirrors and even ASCII art.

The studio holds a fascination for the use of mechanical technology, low resolutions and binary palettes. They try to understand how to construct complex images from small grids, exhibiting them in displays of, usually, high contrast and of industrial quality.

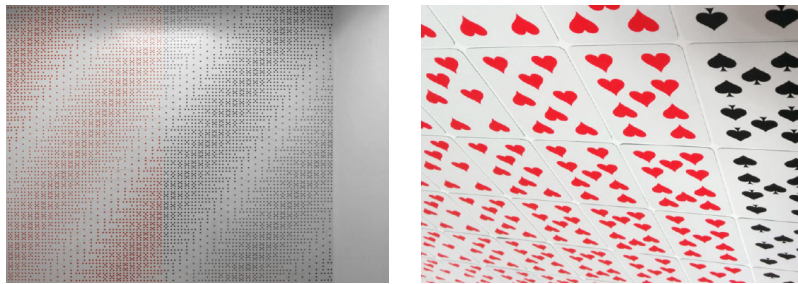
When they solve the question of which display to use, they move on to drawing the programmes, normally using Processing, in which the project's mechanical impulse is simulated on the digital screen. These programs allow them to work with different compositions and to simulate the use of the physical engine. Therefore, these software tools would work as sketching tools which made the project more agile. Oftentimes, these programs also evolved to become edition software or even software to communicate with the physical display (Visnjic, 2017).

One of their more distinct projects uses deck cards as modules to generate bigger patterns. This work from 2015, entitled *Carte*, distances itself from the norm of the studio since it was not kinetic. Nevertheless, this

project still applies a computational approach, besides the analogical one, to draw the final images. To create and plan the compositions the studio built an application that would generate patterns with the algorithm (Figure 42). The same application would also allow the user to manually edit the result, serving as a mockup program. The cards would later be glued to a wall (Figure 43) (Visnjic, 2017).



**Figure 42.**  
Digital application of the *Carte* system showing one of the generated patterns.

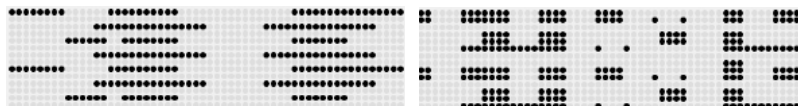


**Figure 43.**  
*Carte* Installation, physical installation. Photo by Luca Meneghini.

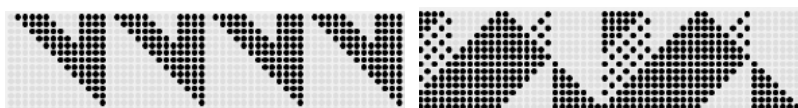
As previously mentioned, the technology Flip Dot Discs kept its presence in the world of design and art and it did not escape this duo of designers who transformed them into projects like *Dots 2* and *Dots 3*.

These two similar projects used two customisable panels shaped horizontally, for which they developed compositions on top of the original dot grid. These panels were inspired by their previous work (*Dots 1*, de 2010) that also consisted of a horizontal display of a bus.

In one of the panels, the result is obtained from an orthogonal grid (Figure 44) while the other is from an oblique grid (Figure 45). A series of patterns is generated, limited by the low resolution of the grid, which contributes to the overall style, and it is exhibited in sequence. This kinetic aspect is used to generate transitions and rhythms on top of the shapes, adding another layer to the project (Visnjic, 2017).



**Figure 44.**  
Static Results of *Dots 2*, using a panel with an orthogonal grid.



**Figure 45.**  
Static results of *Dots 3*, using a panel with an oblique grid.

### 3.4.2 Visual Identities

After analysing cases of study from a mechanical and aesthetic point of view it is also important to reference the role of these systems in the area of design, where they can be adequate to solve a function. A suitable example would be the development of a graphical identity or its artefacts. This problem needs to fit substantially different elements under the same visual language in order to properly communicate to whom the message belongs.

Nowadays, the development of systems that allow us to create flexible identities is becoming more common, the reasons and advantages are clear. By creating this system with success, the number of adaptable options becomes immense. Instead of having the designer follow the identity's rules and manual to build the artefacts, the program is designed to do it for him, turning the designer into a creator of tools, by selecting their needs, instead of having to produce and explore every alternative. Such a system would be, in fact, an investment that would compensate in the long run, or with a wide variety of work to deliver.

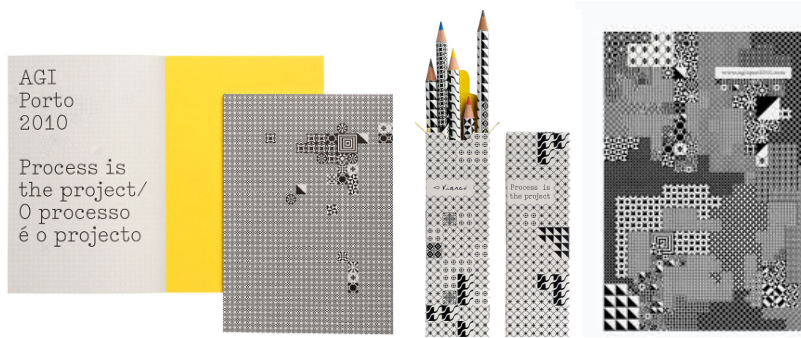
#### **AGI OPEN**

This project by the studio R2, a generative graphic identity based on modules, was created for the AGI Congress in Porto, in October of 2010.

The design of each module was representative of the characteristic and traditional use of the decorative tiles of the city of Porto. The versatile use of the grid allowed for them to be organised in different ways, creating an infinity of results that consisted of mixes of patterns and distortions. The intent was to reveal the beauty of these irregularities. Thanks to the versatility of this solution, it was possible to generate different posters and portraits as it is possible to see in Figure 46.

The success and uniqueness of the results are based on the use of the tile as a complex system of elements. These are explored and manipulated at different scales, which contributes to the final visual effect (R2, 2010).

Upon analysis, this project turns clearer to what the possibilities are of the proposed system of this dissertation. The project takes advantage of a specially parameterised system to a specific context: the pieces consist of the rules to use the grid, to define the modules and to adapt the typography. With these rules, it is then possible to generate various artefacts from a single system, artefacts that can constitute a graphic identity.



**Figure 46.**

Posters, portraits and other applications of the graphic identity generated for the AGI Open Congress in Porto, in 2010, by the studio R2.

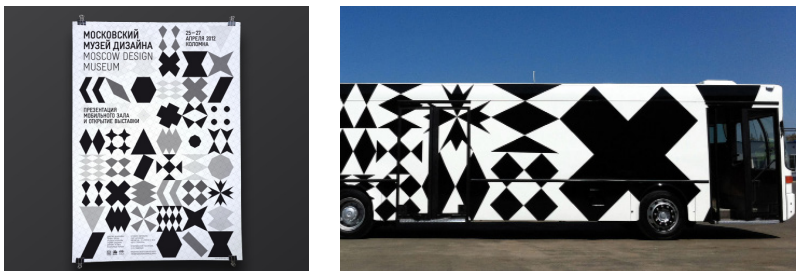


### MOSCOW DESIGN MUSEUM

The studio Lava, in 2012, developed the identity for the Museum of Design of Moscow. Similarly to the previous presented project, the identity is based on the patterns of Russian crystals that represent an inheritance of the history of Russian Design.

From what can be described as a digital stencil, it is possible to generate several geometric shapes. These shapes form the base of a graphic identity that was applied in exhibitions, communication, the bus, catalogues and other artefacts (Figure 47).

The goal of the studio was to develop a method of work that involved strategic thinking that worked on a unique design to solve the problem (Lava, 2012).



**Figure 47.**

Poster and bus decorated with the identity of the Museum of Design of Moscow.

## 3.5 Typography

Similarly to the development of flexible identities, there are flexible typefaces or modular typefaces that have recently gained more presence and opened a path for more cohesive and distinguishable types but also for adaptable and dynamic ones that shape into different sizes and weights.

### 3.4.1 Modular Typefaces

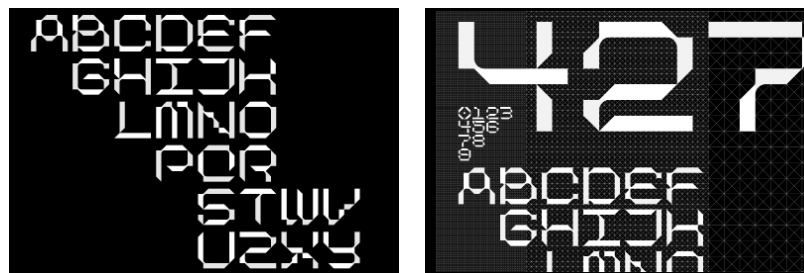
In this section we will be analysing typography projects that use basic shapes or tiles of varying shape to form font families. By using this system, each glyph becomes easily variable and they can even be programmed to react to, for example, music.

#### GRACE-FO

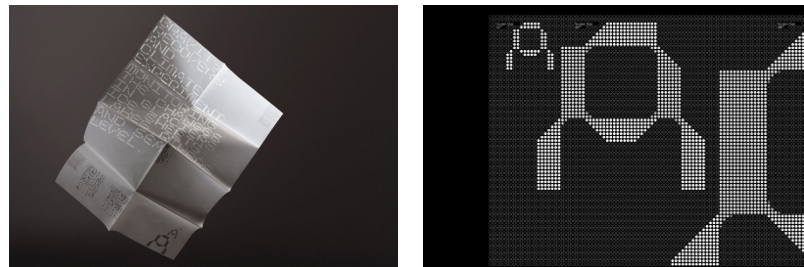
In 2002, NASA and the German Aerospace Center worked together on a project that intended to study the climate, geology and the oceans on Earth by observing anomalies in the planetary gravity. The program Gravity Recovery and Climate Experiment (GRAVE), in 2017, envisioned two satellites orbiting the globe. These satellites were called Grace-1 and Grace-2 and would be launched in May of 2018, working as a pair.

After the launch, studio Accept&Proceed received a request made by NASA/JPL, for an installation of a screen of 4.5 meters height where they would exhibit live information from the satellites to the JPL Headquarters visitors. Additionally, they also requested that they develop a typeface that

**Figure 48.**  
*Grace-Fo*. Typeface created by studio  
Accept&Proceed for NASA.



**Figure 49.**  
Examples of applications of the  
typeface *Grace-FO*. A flyer (left)  
and a display (right).



would display the satellite data. The studio, inspired by the relationship of these satellites, created and drew two elements that remembered the shape of the satellites. These, when combined, form the typeface ('Accept & Proceed', 2020). This last one is drawn over a triangular grid (Figure 48), creating a system that could be easily adapted to create other graphics and artefacts using the same visual language.

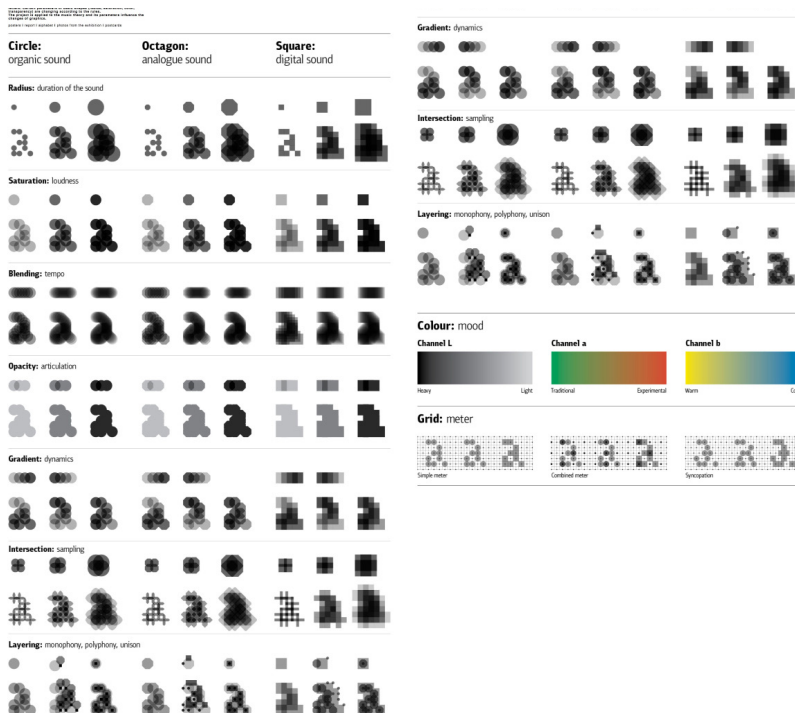
The goal of the studio was to represent the mission's voice through a typeface and a characteristic design that told an emotional story. The results delivered this through the use of modularity and a glimpse of the design of the future that can adapt to different formats (Figure 49).

## TYPOGRAPHIC MUSIC

Developed in 2011 as a graphic identity for a music festival, the project *Typographic Music*, by Dina Silanteva, exploits music theory to influence and change the values of the parameters of the graphics (Sullivan, 2011).

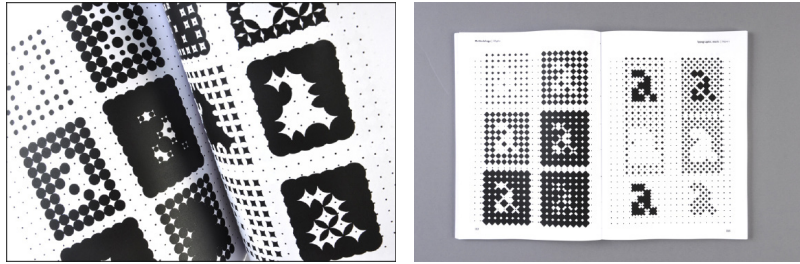
This work is a research project in generative typography using a basic grid and three simple geometrical shapes to build the letters. The parameters of these shapes, like the radius, colour, transparency, change in accordance with the rules of the system as seen in Figure 50 (Silanteva, n.d.).

The results can be used as a negative or monochromatically (Figure 51) and can be even overlaid over other results with different attributes, like colours as exemplified in Figure 52.

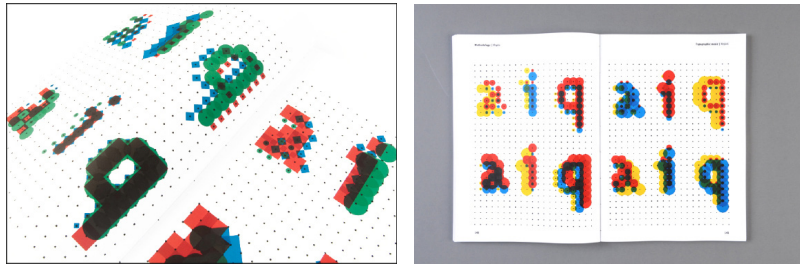


**Figure 50.**  
*Typographic Music*. Grid of attributes and reactions by the system to affect the letter, by Dina Silanteva.

**Figure 51.**  
*Typographic Music.* Printed  
monochromatic results.



**Figure 52.**  
*Typographic Music.* Printed  
chromatic results.



#### MUIR MCNEIL

The studio Muir McNeil, founded by the duo Paul McNeil and Hamish Muir in 2009 focuses on exploring algorithmic and systematic methods in the design of typefaces, graphic design and moving images. Their work is a valuable reference due to the regular presence of modularity and use of the grid in the advantage of the creation of systems to create typefaces. Besides this, the duo is known for taking advantage of strategies of optic illusion to create a balance between the adaptation of the letter and its legibility and readability (Gosling, 2015).

Following this, it will be analysed two of their works whose characteristics describe essential knowledge for the considered project of the dissertation: *TwoPoint* and *Intersect*.

*“As graphic designers who often employ a typographic approach, information hierarchy mediated in carefully structured ways has always been central in our design work. There are no mysteries in the systems we use – no traditional skills, trade secrets or black arts are employed. We are not type-designers in this sense but graphic designers interested in developing processes, generating forms and systems to work with.”*

-Muir McNeil in an interview for *It's Nice That*, 27 in April of 2015

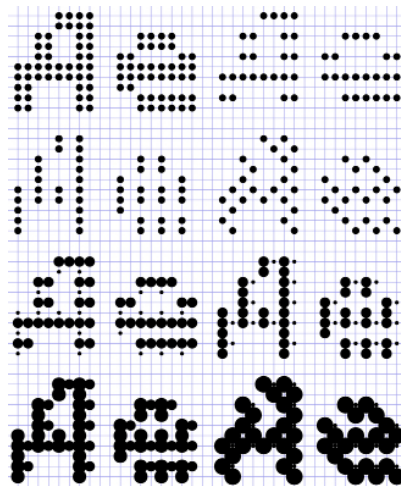
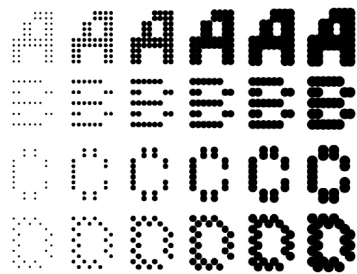
Inspired by the use of dot matrices and typography exhibitions in LED displays, *TwoPoint* is a geometric and monospaced type that explores the use of permutations in the parametrization of the system to create different typography shapes. This type of system makes it easier to produce a range of different outputs.

The individual parts of each letter work as components that vary in

order to adapt to different structures. A grid decides the position of the strokes and spaces, aligning the dots in an adequate way.

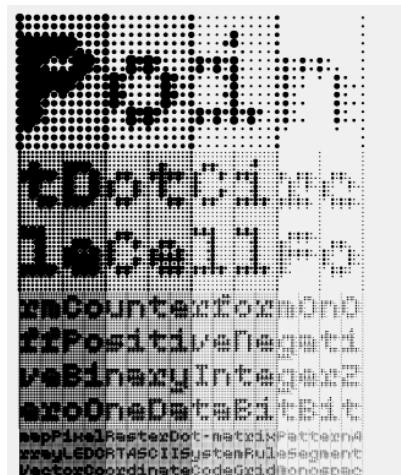
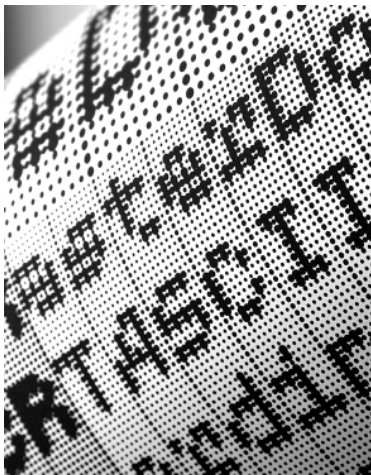
The system also allowed to choose between four different stylisations: a stroke made out of two points that is consistent in two dimensions, another one that removes alternately rows of dots to emphasise the horizontal shape of the letter, one that alternates in removing columns to give emphasis to the vertical shape and lastly, a mode where the removal is alternate diagonally (Figure 53). The four stylisations are designed to interact with each other, therefore increasing the available options. These options shine when it comes to working with print and other static media with specific limitations. It is effective due to the strategy applied to the option to redimension and vary the weight.

Each style has seven weights between Extra Light and Black. The weight is incremented without changing the positioning. Instead, the diameters of the dots that constitute each letter change progressively, working into an illusion of a stroke getting thicker or thinner. Additionally, the relation between size and weight can be calculated and controlled when doing typographic compositions thanks to the modularity (Figure 54) (Muir McNeil, n.d.).



**Figure 53.**

Some variations of the variable typeface *TwoPoint*.



**Figure 54.**

*TwoPoint* printed in varied sizes.

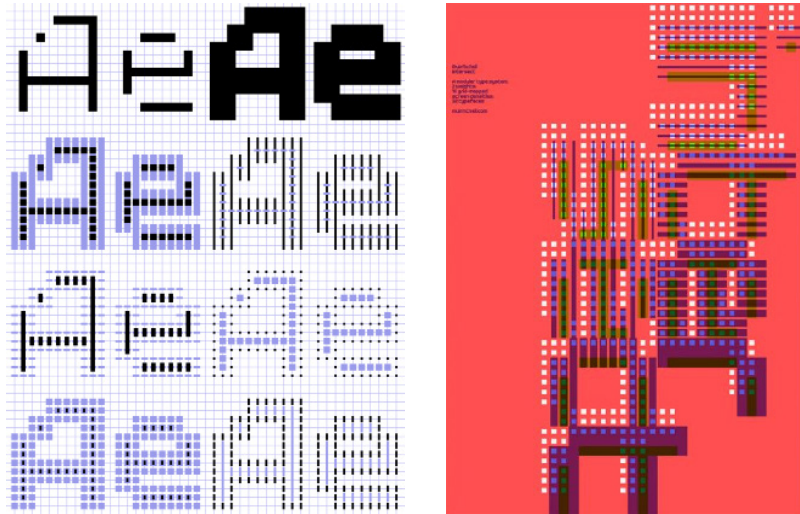


The typeface *Intersect* explores the notion of typographic weight, meaning the relation between painted areas and empty areas that form the characters. This typeface goes further than other traditional typefaces, where it is only possible to play with the binary contrast of black and white and shape and its negative. To surpass this limitation, the system simulates a range of weights that create different illusions of densities in the body of the letters.

This system makes available variants A and B. These share sixteen screen patterns distributed in four groups with four weights each. The system was set so the two hundred and fifty-six possible configurations would be combinable (Figure 55). The possibilities are not infinite, but they are more than enough.

**Figure 55.**

*Intersect* typeface: variations and grid (left) and poster (right).



## 3.6 Final Thoughts

So far we have analysed a collection of projects that contributed to the development of the project. Either by providing us important knowledge in their documentation, inspiring new functionalities that fit our goal or by showing us the advantages and disadvantages of building modular tools, this selection of examples influenced our decisions and the way we perceived our system.

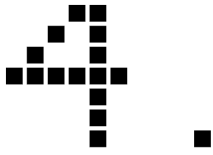
Although more projects could be included, each presented project brought a different topic that had a big influence on the decisions for the final project.

Throughout all of these there was a consistent use of a grid and modules (that can be tiles, shapes or images) to help portray a bigger image. However, each project suits a different context and purpose.

Examples such as *Image Mosaics* and *Halftone QR Codes* taught us how the qualities of the input images would affect the results and we could improve their quality. Projects from the area of New Media Art revealed how the platform could be experienced by exploring different user interfaces and adding more controls such as a drawing functionality inspired by the project *Interactive Flip Dot Display*. From the Design section the importance of having different export methods, resolutions and sizes became clearer. Lastly, the projects seen in the section of Typography inspired us to explore fonts specifically drawn for the project and to add different settings such as size and stroke.

Following this, the next chapter will explain the developed project in detail, where it will be noticeable how each one of these projects influenced each decision and functionality.





# Practical Work

## 4.1 Concept

The concept of modularity is evolving in the area of Design. To correctly define a problem or a goal and to break it into several tasks, facilitates the process of answering correctly to its needs (Lupton & Phillips, 2008). The result is a list of available solutions from which we can select the most adequate (Gerstner, 1964).

The question is if these ideas can be applied to a visual tool. This would give the user a set of limited pieces that can be combined to form different combinations and answer different requests. In this case, we answered this through the generation of digital mosaics. Mosaics carry the bases for a modular system, where the modules are the tiles and the bigger structure is the final image. Since we are working within the digital world, the tile can be dynamic and highly parameterizable which means it can be used to obtain different images, sizes and formats.

*“A modular system is composed of units (or modules) that are designed independently but still function as an integrated whole. Designers achieve modularity by partitioning information into visible design rules and hidden design parameters. Modularity is beneficial only if the partition is precise, unambiguous, and complete.”*

(F. K. Pil & Cohen, 2006)

### 4.1.1 Context

The objective was to develop a framework that deals with a variety of problems instead of focusing on understanding one. Therefore, the described tool is not limited to the area of Design. We can imagine this framework applied in the area of New Media Art, generating live results from different inputs such as movement, live capture or other scenarios.

The results that we produced are also not limited to the one format. We can generate these variable patterns for social media posts, digital installations and websites. However, besides the digital representation, the results can be printed for book covers, posters, packaging, used in laser cuts or even clothing.

### 4.1.2 Proposal

Our proposal is to create a computational system that can interpret an input image and translate its areas into adaptable monochromatic tiles that represent different tonal values. The output is an optical illusion, where the human eye perceives the original input, however, it is communicated through a transformed visual language: a mosaic.

The developed system interprets several types of inputs and quickly transforms them into mosaics with customisable patterns. These inputs can be images, videos or even live capture. The system then interprets the received inputs with the rules of the chosen monochromatic modular system, retrieving a mosaic. When it comes to parameterization, the user can change the grid, tiles of the pattern, colours of the tiles and the background and other features. To further extend the options, the user can draw and write over the result. In the end, they can choose different formats to export.

To evaluate the concept, we proposed adapting the system into a website and testing it with users. The website required developing a visual interface to bridge the engine and the user and allow for an easier way to manipulate it. By doing so, we ought to test its usability and additionally gather feedback on the overall concept.

Lastly, throughout the first part of the project, the target audience included just designers, however, this proved to be a narrow perspective. Even by analysing the state of art, it is noticeable the number of similar projects used in New Media Art. Moreover, the presented system was also a part of an art exhibition which served as a way to determine the interest and value within this area. Due to this, the target audience for the website includes designers and new media artists with experience in image editing and graphic tools.

### 4.1.3 Concepts and Terms

Following this, it is imperative to justify any terms and notions used throughout this project. The goal is to keep a cohesive visual language and avoid different connotations.

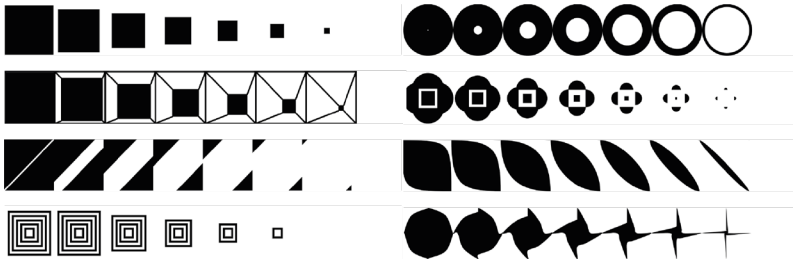
Throughout the document, we mentioned that the system transforms an input image. The system groups its pixels by cells depending on a chosen grid and calculates the average lightness. After this, it passes this average to modules. Then, these modules draw the shapes in their respective proportion and place them in the grid.

Regarding the modules, these are programmable and scalable. We will refer to these as tiles since it is a closer definition of the units we chose. Their modular character is represented by how they are interchangeable despite each having its strategy to interpret inputs. By this, we mean that each tile (Figure 56) has its drawing rules which indicate how to draw the tile depending on different parameters (Figure 57).

**Figure 56.**  
Varied tile designs.



**Figure 57.**  
Varied tiles with weight variation.

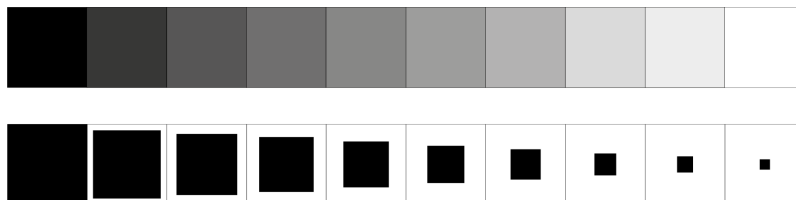


Keeping in mind the monochromatic constraint of the tiles, these needed to represent a tonal value of a given zone of an image (or a cell) by only playing with the shape and counter shape or, in other words, black and white space. The average amount of black and white in the drawing of each tile represents its weight. This weight represents the tonal value of the tile that the human eye perceives as an optical illusion.

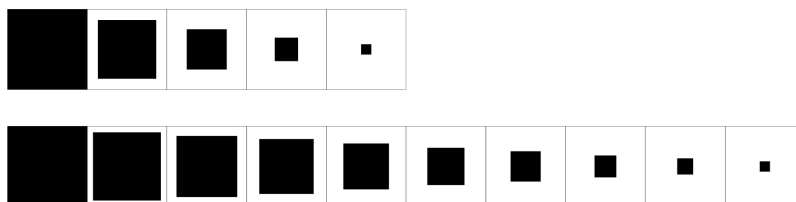
The number of available weights is also a parameter. The bigger this number is, the bigger the number of tile variations and, consequently, the more tonal values it can depict. Therefore, there will be more detail in an image since more tonal values will be perceived (Figure 58). The range of weights can be defined as continuous, in which it is possible to represent any tonal value of an input image (Figure 59). As an example, if an image cell has an average of 0.2 (between 0 and 1) in tonal value, and there are five weights available in a tile ordered from darker to brighter, the tile, which in this case will be the most similar, would be the second darkest weight.

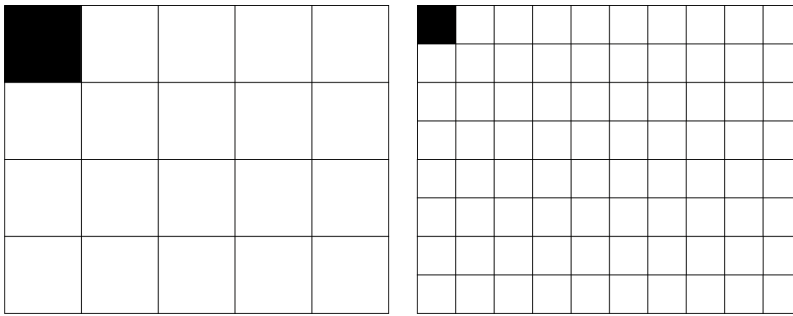
Another component that is important to explain is the grid. The dimensions of the image's cells that the system will interpret will depend on the set size of the grid (Figure 60). The more columns and rows, the smaller the size of the cell to evaluate and therefore. This will also increase the number of cells there will be. This scenario results in a higher level of readability of the output and less for the module and vice versa.

**Figure 58.**  
Translation of grayscale to a monochromatic representation.



**Figure 59.**  
Different ranges of weights.  
The range number may be continuous instead of static.





**Figure 60.**  
Different sizes of the grid influences the size of the module.

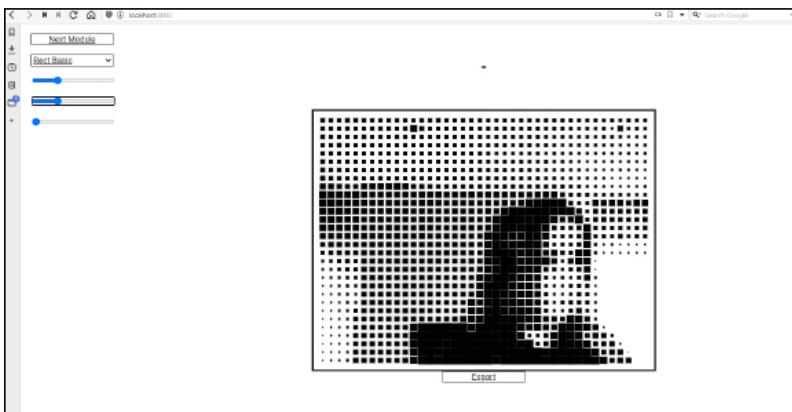
Lastly, we set a monochromatic constraint so the focus is on the relationship between the base image and the generated patterns. This allowed the system to not focus on how to deal with a colour matrix with more than two options.

## 4.2 Development

### 4.2.1 Early Prototype

An early prototype of the system was developed during the first phase of the project. The goal was to visualise the idea and help to predict possible problems and realise needed features. This prototype began as a simple web platform: the users could import an image or use the input of their webcam, and the system would translate these into a grid of tiles. From this point, the user could choose the tile they would like to apply, colours, and the size of the grid.

Several ideas and problems appeared with this early prototype. The first problem regarded the visual inputs loaded into the system. If they were overexposed, underexposed or lacked contrast, the results would have low quality. Therefore, the system should provide a way to make visual adjustments to the image. The problem was solved by adding filters that



**Figure 61.**  
Screenshot of the early prototype, capturing image.

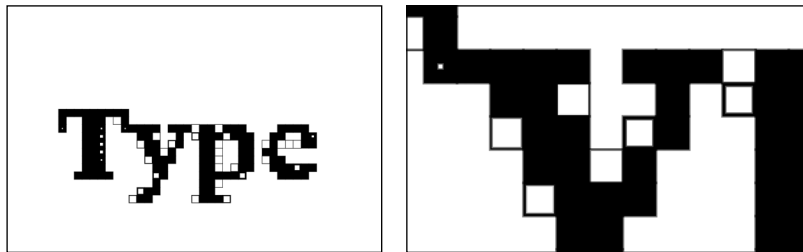


let the user control brightness, contrast and sharpness (Figure 61). Later, it will be shown how this would not be the only necessary change to improve the image's readability.

Secondly, some experiments were conducted using several images of text composed using different fonts, such as Roboto (Figure 62). These were used as the system's input, and the results showed the potential for integration of typography. However, we decided that instead of treating the glyphs the same way as the input images, a custom algorithm was going to be developed to draw fonts designed specifically for this system and its small grid sizes.

Regarding the user interface of this prototype, even though it was a more primal version, the structure could be seen: settings on the left, and canvas on the centre, which ended up being the most effective way of organising the different available functionalities.

**Figure 62.**  
Early experiments of the system with typography, using images of letters as input.



## 4.2.2 Generator

In this section we will explain how the main functionality works. This means we will be going through the technology we used, experiments with different algorithms, problems that were found and solutions.

### TECHNOLOGY

As a way to make the system easily accessible and exploratory, we developed a website built with JavaScript and with the help of the libraries p5.js, Raphael.js and OpenCV.js. The first one was chosen due to the need for an easy and reliable way of drawing the tiles onto the canvas. This method would prove to be faster than using vectors.

Although initially, the system processed the inputs with the help of p5.js, when it came the time to process them with different filters, p5.js did not offer a stable way to do it. After that, the inputs and filtering responsibilities were dealt with OpenCV.js, which simplified the process.

Finally, Raphael.js was used to generate some animations that can be

---

[1] More about p5.js can be found in [www.p5js.org](http://www.p5js.org)

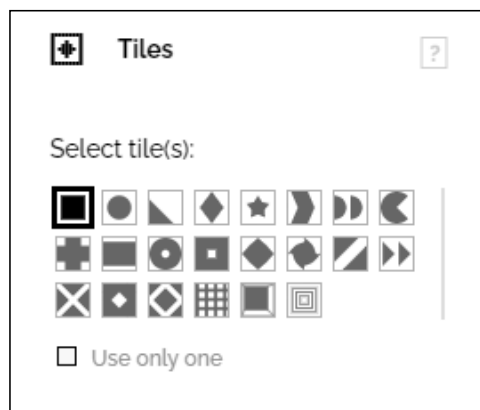
[2] More about Raphael.js can be found in [www.dmitrybaranovskiy.github.io/raphael/](http://www.dmitrybaranovskiy.github.io/raphael/)

[3] More about OpenCV.js can be found in [www.docs.opencv.org/3.4/d5/d10/tutorial\\_js\\_root.html](http://www.docs.opencv.org/3.4/d5/d10/tutorial_js_root.html)

seen in the interface (Figure 63). However, there was no time to program all the tiles with this method. We are planning to use Raphael.js to replace all the responsibilities of p5.svg.js, using it to visualise in vector format.

Additionally, the library jsPDF was used to support some PDF export and the library Spectrum to allow for a more customisable colour input. For MPEG and GIF export, the system uses the library CCapture to process the results.

Lastly, it is important to note that the current version of the project was built and tested for the Google Chrome browser (including other Chromium-based browsers). Although the project works on Firefox and Microsoft Edge, these are not advisable as they were not fully tested.



**Figure 63.**  
SVG's drawn using Raphael.js  
and p5.svg.js.

## SVG VS. CANVAS

When choosing how the user would visualise the outcomes before exporting, two options were available. The first one would be to draw the mosaic in vector format (SVG), which would offer interactivity with each tile, higher quality (since it would resize whenever needed) and a more dynamic export format. However, a disadvantage found with using SVG's was that they become increasingly heavier and slower when dealing with a higher number of objects.



**Figure 64.**  
Close-up of images exported with  
different formats: as bitmap (left),  
and as SVG (right).

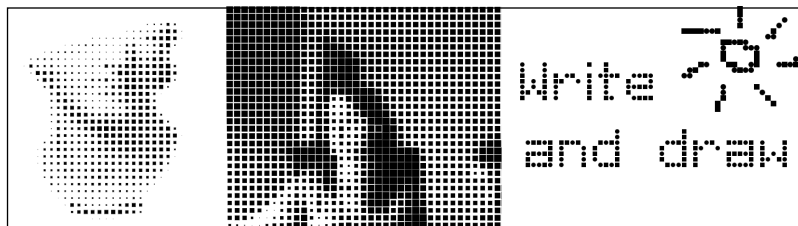
On the other hand, the second available option was to use HTML canvas which uses Bitmap. This option allowed the system to draw more objects in real-time. However, it would be at the cost of the quality of the image in terms of resizing (Figure 64). It would also make the process of interacting with the tiles harder to implement. The advantage of using canvas was that the number of frames per second, when drawing the shapes in a loop, did not get increasingly smaller with a larger quantity of objects. The opposite happens when using SVG, resulting in a very low framerate when drawing multiple objects.

The idea of having both options for displaying the result was discussed. If the changes were taking too long to process or the frame rate was low when drawing with vectors, the user could get a notification to change to bitmap mode. However, this option was not a priority since we could still export as SVG despite the fact it was being displayed as a bitmap. It would not serve the user more than other features in the backlog would do. Therefore, we saved this feature for future work.

## INPUT

As initially pointed out, the engine can receive different inputs, namely Images, Videos, and Video Capture (Figure 65). However, the user can also choose to create a blank canvas in order to freely draw and write. These different inputs also define different strategies to generate the results, as we will see moving forward in the Grid section. After uploading, the system can apply a set of filters to these inputs as it will be seen later on.

**Figure 65.**  
Different types of inputs: image file (left), live video capture (centre) and white canvas (right).



## GRID

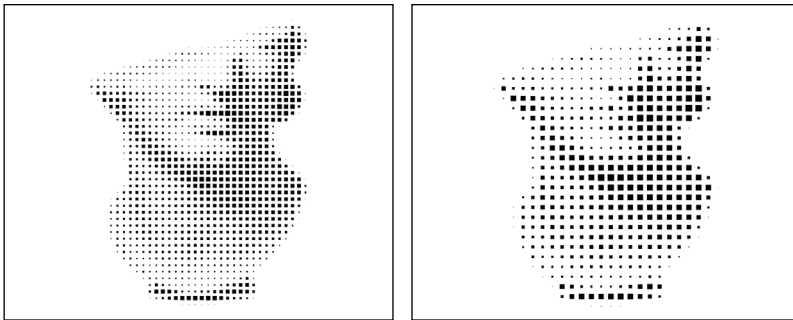
After loading the input and processing it, two different types of grids are available. The first one adapts itself to the original aspect ratio of the input (Figure 66). This means that despite the changes that the user makes, the grid always goes back to the correct ratio. This type of grid is used in Image and Video uploads because we do not want to change their aspect ratio since it would distort the images. The way it functions is as follows. When an image is uploaded, the system creates a resized copy of it to match

the grid size (which will always be proportional to the aspect ratio of the original image). For example, if the grid has 40 columns and 60 rows, the image will have a width of 40 pixels and 60 columns if its original ratio is 2:3. This is done to obtain the lightness of each pixel. This way, we do not have to iterate over the group pixels in the desired zone of the image to get their average lightness. It is also important to note that this resizing is obtained through the OpenCV function `resize()` using linear interpolation.

The second type of grid, called Extendible Grid, works the opposite way. This one sets the grid to any columns and rows the user decides, and it is used in the Drawing Canvas mode where not following an aspect ratio does not impact the texts and drawings made by the user (Figure 67).

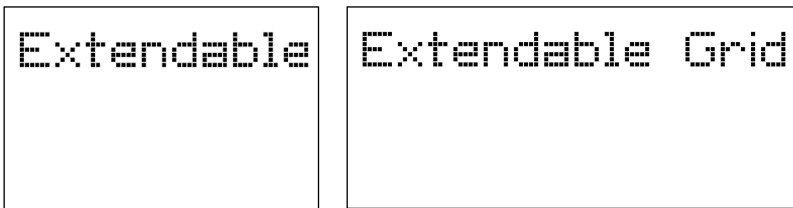
Both options are rectangular grids in which the cells consist of fixed squares. More grid options can be added in future work to obtain an even greater range of results. These will present a higher level of complexity as they imply rethinking how the matrix of lightness values is calculated and how the system positions the tile in the canvas.

Initially, it was possible to modify the size of the cells in each axis, which would result in the rectangular tile. However, this feature was withdrawn since it would distort the image.



**Figure 66.**

Image using proportional grid: set with 60 columns (left) and 40 columns (right). The number of rows adapts itself to match the original proportion of the input image.



**Figure 67.**

Image using extendable grid: set with 60 columns (left) and 90 columns (right). The number of rows does not change.

## HALFTONING

To organise all tiles and their parameters, the system generates a list of Cells whenever the grid changes size. This list has one cell for each pixel of the resized image. The cells store the value of the lightness of the pixel, its position and a tile.

Initially, the main goal of this list was to facilitate the process of saving the lightness values to create smooth transitions between frames. This feature is currently unavailable due to some adaptations made to increase the speed of the platform; however, it is interesting to mention the attempt

of making the tiles transition between values smoothly. The function used to achieve this effect calculates the transitioning value of the current frame. For this, it needed a current value, a target value and a speed, resulting in the following formula:

$$\text{currValue} = (\text{currValue} * \text{speed}) + (\text{targetValue} * (1 - \text{speed}));$$

In the prototype, the weight of each tile was equal to the value of red of the pixel. However, this proved to be not the best formula to obtain the lightness of each zone. As seen in Figure 68, the images whose grayscale values were obtained purely from the red values of their pixels lose the visibility and contrast of any previous yellow tones.

Several formulas were tested, but we ended up picking the most commonly used. The chosen formula reflects the lightness (also known as luma or perceived brightness) function: where green light contributes the most to the intensity perceived by the human eye and blue light the least. This lightness value can be calculated from the RGB components with the following formula:

$$\text{output}[\text{row}][\text{col}] = (0.2126 * r) + (0.7152 * g) + (0.0722 * b);$$

In the end, by drawing the tile of each cell with the attributed weight in the correct position, we get an optical illusion. In the result, we can recognise the input image due to the different tile weights that, from a distance, look like variations in lightness.

**Figure 68.**  
Result of the application of the first  
formula (left). Result of the second  
formula (right).



## TILES

Regarding the tiles, we developed a group of these to represent the weights with different designs. This weight represents the respective tonal value of a given zone in an image, which in this case, is a cell (since we resize the image to match the number of columns and rows of the grid we set). The higher this value, the bigger the area that the tile occupies (Figure 69). For example, if the colour of a tile is set to black and the value of lightness of a cell is 0 (in a range of 0 to 1), it would mean that the tile would be drawn with the maximum black pixels in the cell according to its design rules.



**Figure 69.**  
Tile changing its design to match different tonal values.

The parameters that the tile draw needs to receive are the following: position in the X-axis, position in the Y-axis, width and height of the cell where the system is going to draw the tile, the weight of the tile (representative of the luminosity value it intends to portray), tile colour and background colour. Here's an example of a line of code that draws a tile:

```
rectTile.draw(posX, posY, cellW, cellH, weight);
```

Following this, the procedure goes through mapping the value of the weight to the area of the tiles' shapes. Then, we apply the resulting dimensions to the design. For example, if the design consists of a square that scales in both X and Y-axes, it will have a width of the tonal value times the width of the cell.

```
sideW = map(weight, 0, 1, 0, cellW);  
sideH = map(weight, 0, 1, 0, cellH);  
rect(posX, posY, sideW, sideH);
```

Several tiles can be applied in one composition of mosaics. When more than one is selected, the tiles are distributed through the cells using a pseudorandom order. Since the designs vary from tile to tile, the maximum areas of black vary as well. Unfortunately, this means that mixing tiles in the same pattern can be tricky and can create noise in the final result (Figure 70). This created a problem that revolved around how different designs of each tile would disturb the proportion between tonal values and the amount of black pixels.

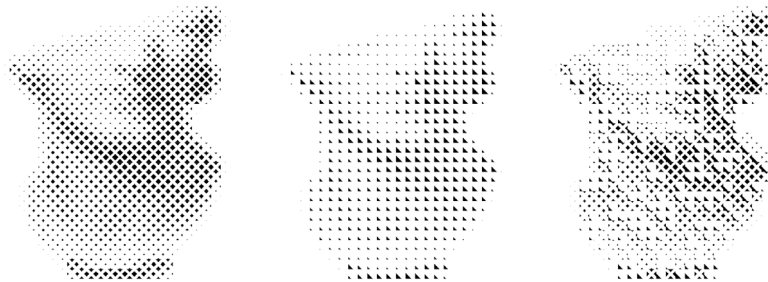
To solve this, two options were available. The first option was to compare them and calibrate them. We could achieve this by having a JSON

file with the maximum values that each tile could take when combined with others. The second option was to entrust that the user would choose tiles that had synergy. This last option would not interfere with the initial vision for each tile. For instance, the simple and visually light triangle tile seen in Figure 71 was initially designed to be used in clean text or with other light and simple shapes such as the circle. Meanwhile, some complex tiles were envisioned to build an interesting pattern on their own.

One of the initial ideas was to allow users to make their own tiles. This could be achieved by uploading a bitmap or an SVG or drawing it in a separate interface specifically designed for this purpose. However, given the main goal of the project, which was to show how a modular system could quickly transform different inputs into different artefacts, this task was set to a lower priority. It was possible to anticipate technical issues that the development of this feature would bring. It was unrealistic to try to solve them and maintain the quality in the available time frame. Nonetheless, it is interesting to imagine this possibility in future work. For now, the available tiles allow us to obtain a wide range of results as well as implement and test the value of the concept.

As previously hinted in this section, some tiles have different intentions. Some of them focus on basic shapes to create minimalistic patterns, others envision more complex combinations of tiles. Different factors account for the design of these tiles. One was the tile's symmetry which would influence the final pattern and the interaction between them (Figure 72). Lastly, another factor that accounts for different effects is the minimum and maximum shape of the tile, which influences the range of tonal values. Tiles with a wider area work better with images as they are capable of representing a wider range of tones.

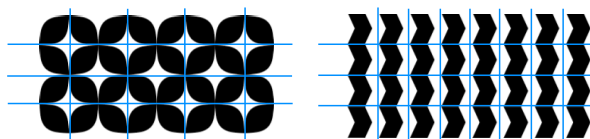
**Figure 70.**  
Image processed with different tiles:  
with a complex tile (left), a light tile  
(centre) and different tiles (right).



**Figure 71.**  
Triangle tile used with  
*Truchet Sans Serif*.



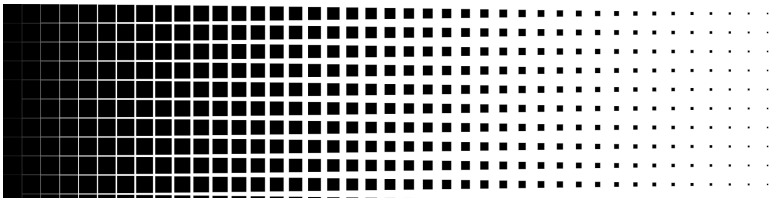
**Figure 72.**  
Different axis of symmetry in two tiles



## GENERATE

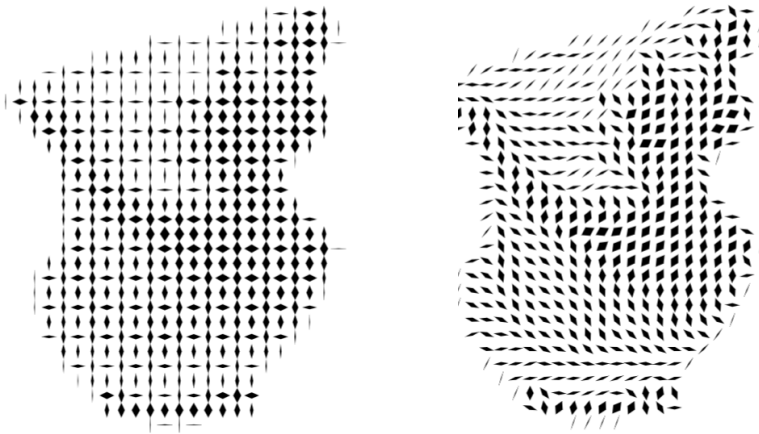
After having the matrix of cells set with the corresponding tonal values and type of tile, the generation of the output consists of piecing these steps.

By iterating through each zone in the grid and calling its respective cell, the system draws the design of the tile assigned to it (Figure 73). Different modifications can be added at this point to alter the result, as we will see later. We modify specific cells in this matrix to add text, drawings, or introduce geometric transformations such as rotations (Figure 74). After this, the system can finally draw all the tiles with the given tonal values (or transformed ones) at the designated cell position.



**Figure 73.**

Result of tiles portraying different weights to form a gradient.



**Figure 74.**

Image processed using the same tile but with different geometric transformations applied: rotation in sequence (left) and rotation according to the tonal value (right).

### 4.2.3 Features

Additionally to having the main algorithm of the system that executes the idea of the project, we decided to add other features. In the following subsections, each feature will be explained by elaborating on the vision behind them, how we developed them and the problems that we encountered throughout the process.



## FILTERS

The first feature that we identified as necessary, after the development of the prototype, was image adjustment (filters). We noticed how some images with less contrast or badly balanced became less readable after being processed by our system (Figure 75).

**Figure 75.**  
Image with low contrast and overexposure processed by the system: with no filters applied (left) and with 50 contrast, -80 blurriness and -70 brightness (right).

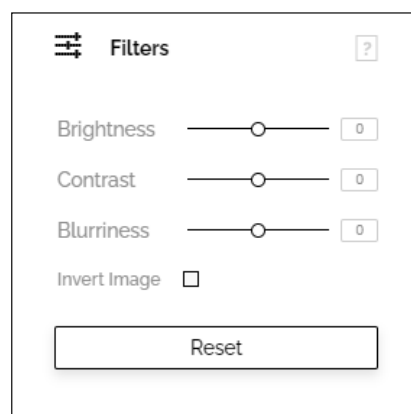


To improve this, a set of filters was added for the user to quickly change and visualise the result instead of having to resort to an extra editing tool. To better visualise the interaction and inputs, the interface for this set of features can be seen in Figure 76. This gave the user more instant control over the results by adding the correct filter settings to balance the tonal values. Another alternative solution that could help with problematic images would be to make the grid bigger. This would result in a higher image resolution and more tonal values represented.

It was due to the need for filters that the OpenCV.js library was introduced to the project. Although there were attempts to apply kernels to the image with the help of the library p5.js, we encountered a problem in which the p5.js library seemed to apply anti-aliasing to the resized image. We found that it was connected with calling the loadPixels function twice on the same image. However, this was necessary to do before applying a kernel and after the resize. Later, it was found that the library p5.js was not an ideal choice for image processing and pixel manipulation.

The order in which the filtering and resizing was done is as follows. Firstly, the image is processed with filters in its original size only if the parameters of each filter are set to a value that is different from their default

**Figure 76.**  
User interface for the Filter panel.





**Figure 77.**

Image processed with different exposure and contrast: exposure slider set to 50 (left), original result with slider set to 0 (centre) and with the contrast slider set to 50 (right).

value. This condition exists to prevent processing the image several times with no need. The filtering is purposely done before resizing the image to get a more trustworthy lightness value. If the image was processed after resizing, information would be lost since we would be processing values that represented an average. By manipulating the pixels that already represent an average of a group of cells, the results would further distance themselves from the original, producing a less defined and readable artefact.

Each time a filter is changed or the grid values are changed, the image that was originally uploaded goes through these two steps again.

The user can, therefore, adjust the Brightness, Contrast, and Blurriness/Sharpness. Other filters were pondered such as the introduction of median and bilateral filters and were saved for future work explorations. For Brightness and Contrast, the system applied the following OpenCV function where  $\alpha$  value and  $\beta$  value control the contrast and the brightness respectively. The limits chosen for the alpha and beta parameters were (0,2) and (-127,127), respectively. The results can be seen in Figure 77.

**convertTo(img, src,  $\alpha$ ,  $\beta$ )**

As for the Blurriness and Sharpness, also known as low-pass and high-pass filters, they were achieved by applying a kernel to the image. Moving the input slider of this filter to the right would increase the size of the kernel, meaning the image would get more blurry. This was achieved by using the calling the OpenCV function, where `img` and `ksize` would refer to the size of the kernel that was going to be applied:

**cv.GaussianBlur(i, i, ksize, 0, 0, cv.BORDER\_DEFAULT);**



**Figure 78.**

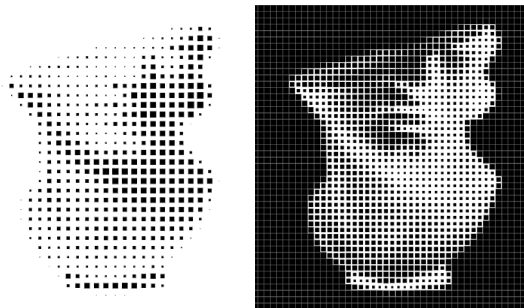
Image processed with different blur values: Blur slider set to -100 resulting in a sharp image (to the left), original with slider set to 0 (centre) and with the slider set to 100 (to the right).

However, moving the slider to the right (to a negative value) changes the strategy. The size of the kernel also increases but, instead of applying it to the image, the operation would be to subtract to the image the result of applying the kernel to that same image. The procedure would look like this, where *aux* represents a copy of the original image. While the function `GaussianBlur` applies a blur, the function `addWeighted` allows us to subtract the *aux* image to the original image (Figure 78).

```
cv.GaussianBlur(aux, aux, ksize, 0, 0, cv.BORDER_DEFAULT);  
cv.addWeighted(i, 1 + 1, aux, -1, 0, i);
```

Additionally, an option to invert the image is available in this section. However, instead of processing all the values of the image, the values of the final matrix are mapped invertedly (Figure 79).

**Figure 79.**  
Image processed with different mapped values: original result (left) and result after applying invert filter (right).

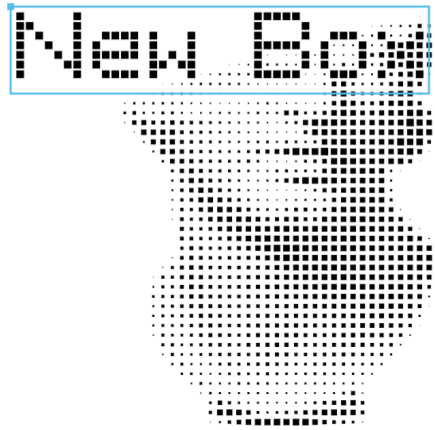
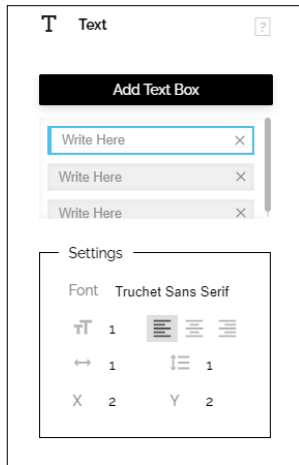


## TEXT

Another explored feature was a text editor that allowed the user to write and manipulate parameters such as font, kerning and position. For this, we designed fonts to work in harmony with the grid and the tiles. The characters and glyphs are defined in *.txt* files. Then, the system interprets them and lists the drawing for each glyph by saving their matrix of values. This matrix exists to overlap the original matrix of tonal values of the input.

Proceeding to explain how this algorithm works, the system fetches the *.txt* file for each font. It then analyses every existing character registered in each font and saves it into an object that works as a dictionary: when accessing a character, the object retrieves its interpretation of it, meaning it retrieves the matrix of values to draw it.

Alongside the font choice, parameters such as kerning and line height are available to the user (Figure 80). The user can also change the position and height of the text box, either by indicating the desired coordinates in the grid or by dragging the text box in the canvas. Some features were also planned for future work. These involve double-clicking in the textbox in order to edit its text and having the option to break the line, which will open the possibility to edit the line-height.



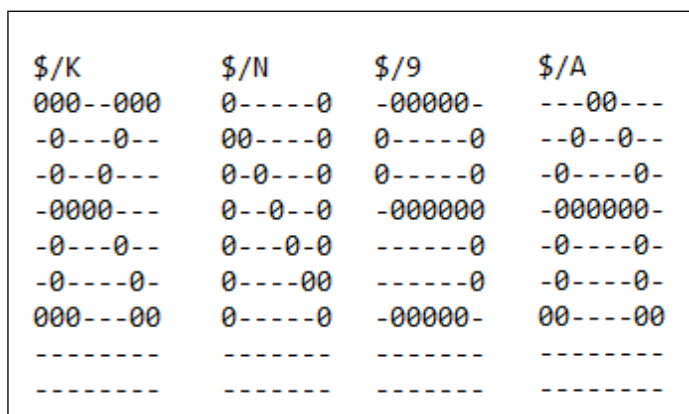
**Figure 80.**

Interaction with text panel: text panel with different available parameters (left) and its visualisation on the artboard (right).

The user can also scale up the font size. To achieve this, we applied an old technique used in bitmap transformations. Using this technique, for example, one pixel for a scale factor of two is used four times, twice on each of the two successive scan lines.

Having each font designed and modified in a .txt file made the process of creating more quick and easy (Figure 81). Throughout the project, we developed four fonts (Figure 82):

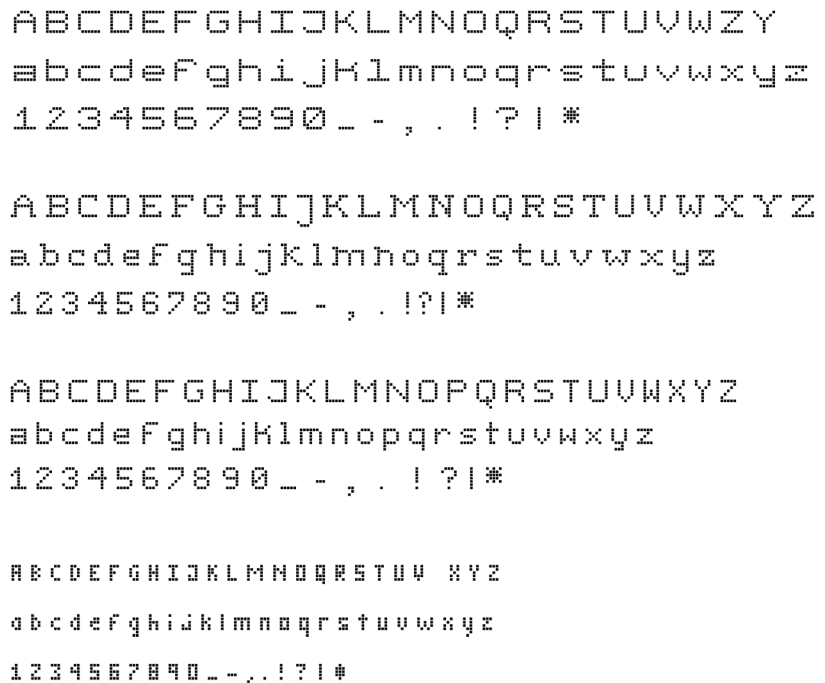
- *Truchet Mono*: a monospaced and non-serif font. It was the first one to be developed. Each character has a Cap Height of 9 points, and all the characters have a width of 7 points. It includes numerals and punctuation.
- *Truchet Serif*: serif version of the first font with slightly bigger descent height to fit the visibility of the serifs.
- *Truchet Sans-Serif*: sans-serif version of Truchet mono; the characters are narrower than both previous versions.
- *Eightbit*: this font was created for smaller grids, however, its use comes at the cost of readability. Each glyph occupies less space. Glyphs have a Cap Height of 5 points and X-Height of 4 points.



**Figure 81.**

Characters encoded in a text file.

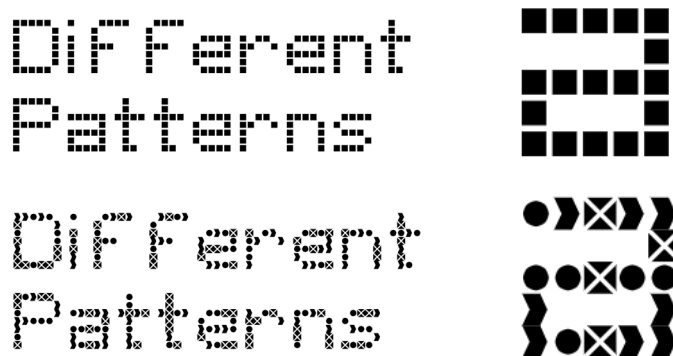
**Figure 82.**  
Typefaces made for the system: *Mono* (first), *Serif* (second), *Sans Serif* (third) and *EightBit* (fourth).



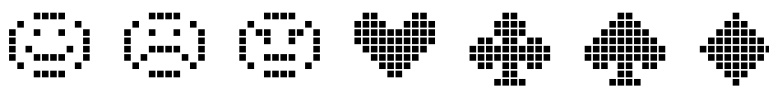
Contrary to images, text compositions deal nicely with patterns of different characteristics as it does not have a hard implication on the recognition of the shapes, especially if the background is clean (Figure 83).

We also attempted to create a pack of emojis (Figure 84). However, due to a lack of time, these were not included in the platform in a way that was easy to understand how they could be applied. For instance, while the user knew that a “smiley face” was obtained by inserting the characters “: )”, they would need further guidance to obtain an emoji of a “like”. Future work could include solving this by adding an emoji visual pack that the user can select and drag to the artboard.

**Figure 83.**  
Different tiles used in the same text configuration.



**Figure 84.**  
Some of the emojis created for the platform.



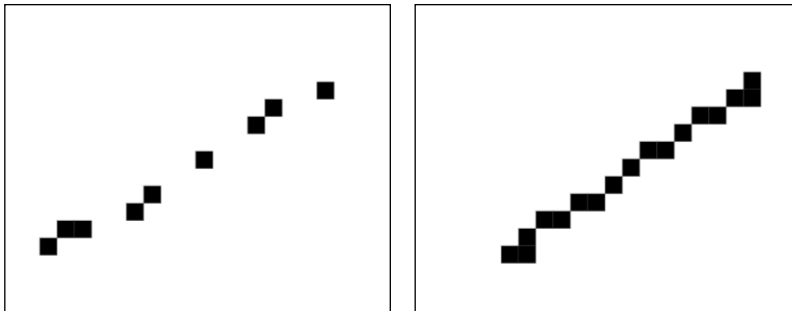
## DRAW

After adding the text feature, there was a desire to have the text over a white background. Therefore, we decided that we were going to add the mode of creating a drawing artboard. This was inspired by the case study *Interactive Flip Dot Display* (see Subsection 3.1).

The implementation of this feature required interacting with each tile and modifying its weight by clicking on it. As previously seen with the text feature, the system creates a new matrix for modifications such as text and drawing, which later is added to the main matrix.

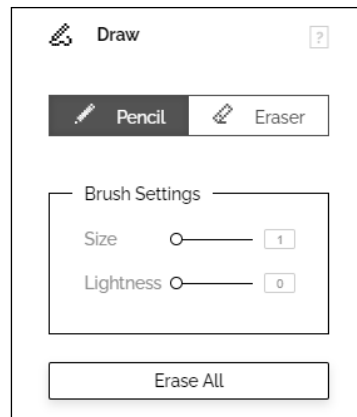
Another challenge in this implementation was to draw pixelated lines that connected two points. After the user clicks on a tile for the first time, the event mouseover is used to keep track of the tiles the mouse has passed. However, this was not enough to draw the entire path as the mouseover event was not always triggered in time (Figure 85). An algorithm was used to solve this problem, where a pixelated “line” was calculated and drawn from the last tile detected to the current tile.

The user can choose between two types of brushes: pencil and eraser (Figure 86). Both have the same parameters, but the pencil sets the weight of the clicked tile to 1 while the eraser sets it to 0. In addition to choosing the type of brush, the user can adjust its size and lightness. The size relates to the number of tiles that are modified while the lightness modifies the weight of the tile. Although this could be confused with the size parameter when using tiles that vary their design by scaling in two axes, this concept becomes clearer with more complex shapes (Figure 87).



**Figure 85.**

User drawing when quickly moving the mouse cursor before (left) and after implementation of a path predictor (right).

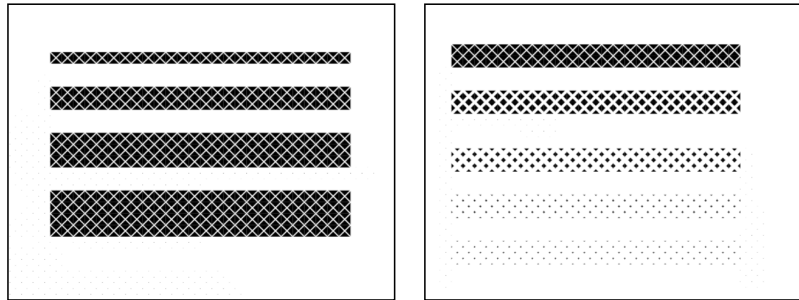


**Figure 86.**

*Draw* panel with different brush options.

**Figure 87.**

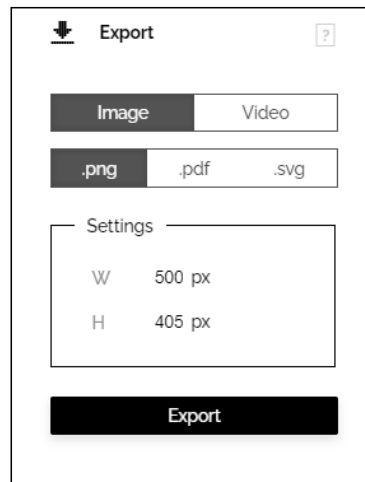
Lines drawn with different values for size (left) and different lines drawn with different values for lightness (right): the value increases from top to bottom.

**EXPORT**

Four modes of input were mentioned in the beginning: image, video, live capture and an empty canvas. Similarly, different types of export are also available. The user can export the result as a PNG, SVG, PDF, GIF and MPEG (Figure 88).

**Figure 88.**

*Export* panel with different file formats.



Each one offers different parameters to customise the output file. For the PNG and the SVG export, the user can change the size, however, only proportionally to the original input. As for the PDF, the user can configure the page format and orientation. Similarly to the PNG, the result will proportionally fit the chosen format, leaving white space on the page if the selected format is not proportional to the obtained result. The GIF lets the user pick the size as well and the number of frames. However, due to the restrictions of the file format, we limited it to a maximum of 300 frames. Finally, the MPEG allows the user to start a recording with a maximum of 3000 frames.

While the PNG was easy to implement thanks to the save function provided by p5.js, the other ones required more resources. The SVG, which we initially thought of solving with Raphael.js, implied that each tile would have to be redrawn with this framework. Additionally, another unknown library would have to be added to export the result. To solve this in the available time, we used p5.svg.js. Its learning curve was smaller and the

code used to export as PNG could be reused to export the desired SVG.

Moving to more challenging formats, we're going to proceed to explain the PDF export. This export had to be vector-based but also had to be executed on the client-side and with JavaScript. Fortunately, we found the jsPDF library to have a plugin that solved this exact problem.

Concerning GIF and MPEG, the system exports these using the library CCapture.js. This library contains features such as frame rate, offering more control over the settings. By clicking the Recording button, the user begins to save the frames in real-time of the results shown on the canvas. By pressing Stop and then Export, the export process begins. The system limits the extension of these files as a recommendation from the library developers and to prevent errors that could include forgetting about pressing Stop. For future work, it would be useful to have a feature where we could record specific time frame intervals of a live capture or video (or even simply recording an interaction with the artboard such as drawings). The user could loop, select the desired timestamps, and more importantly, it would allow the user to edit it as much as they wanted before exporting.

## CONTROL AND VISUALIZATION

In addition to these categories of features, other groups of settings are available to the user on the bottom bar of the platform's interface. These include settings that allow for visualisation manipulation such as *Zoom In* and *Out*, *Fullscreen* and *Fit Screen* (see Figure 89). The last one fits the artboard to the available space.

When it came to figuring out the zooming of the artboard, a slider was considered before opting for buttons and scrolling. We can find these options in most editing programs. The user can also scroll to the hidden areas of the artboard both in the X and Y-axis.

Regarding the buttons Undo and Redo, these were not implemented in time, however, they will be in future work.

In the next section, we will present the architecture of the developed system, where we explain the connections between all of the pieces overviewed so far. We will also show class diagrams to better visualise the system's architecture.



**Figure 89.**  
*Visualisation panel.*

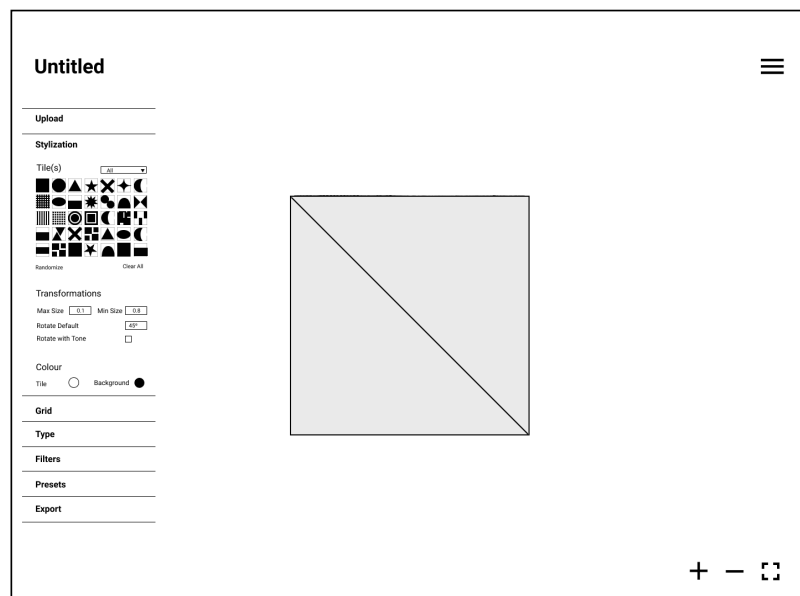


## 4.2.4 Interface

Following the implementation of almost all of the features, we began to develop the interface. This process took several iterations and tests to reach a stable version that was accessible and conducive for the user. This section will begin by explaining the wireframe, then it will proceed to explain the thought process that went behind the mockups and finally, the established guidelines for the design of the platform.

### WIREFRAMES

To decide how the platform should be organised, we sketched different wireframes. Their rough look allowed us to focus on the organisation of the content. It also instigated more debate since it was still a very low-fidelity version. For instance, we can see one of the developed wireframes in Figure 90 (more versions are available in Appendix A). The earliest version had the Export panel on the right side of the screen, so the user could more quickly export. It also had the import function on the bottom of the screen. However, this could lead to confusion on where to find some features, so we decided that all the features would be grouped into one menu. After more experimentation and researching standard interfaces for image editing tools, we decided to take the most common approach and place the menu on the left side. Additionally, settings for the visualisation, such as the *Fullscreen* and *Fit Screen*, can be found on the centre bottom of the page in a floating menu.



**Figure 90.**  
Initial wireframes.

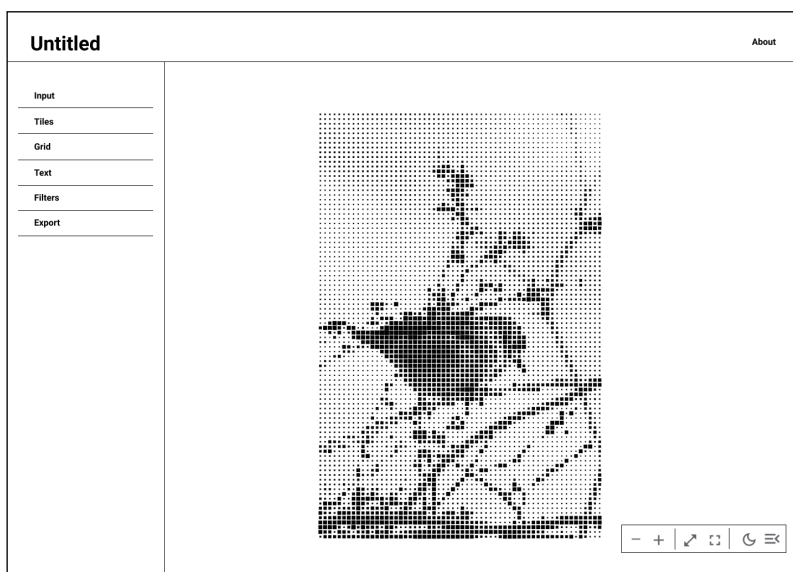
## FIRST MOCKUPS

The first versions of the mockups had the objective of experimenting with different formats and envisioned the platform's usage as being infrequent. They began by looking extremely basic as seen in Figure 91.

The second update had the intention of matching the platform to the theme of the concept. The goal was to have all of the pieces being modular. The inspiration behind this update was the design of the old software *MacPaint*, which had a very evident pixelated look (Figure 101). However, the mockups lacked hierarchy between the elements due to using monochromatic palettes and lack of dimension (Figure 93).

Several experiments followed this version, but they still had the same problem. We made attempts at adding colour, however, these had too much contrast, and the highlighted colour could battle with the colours the user chose for the tiles. These experiments can be checked in Appendix A.

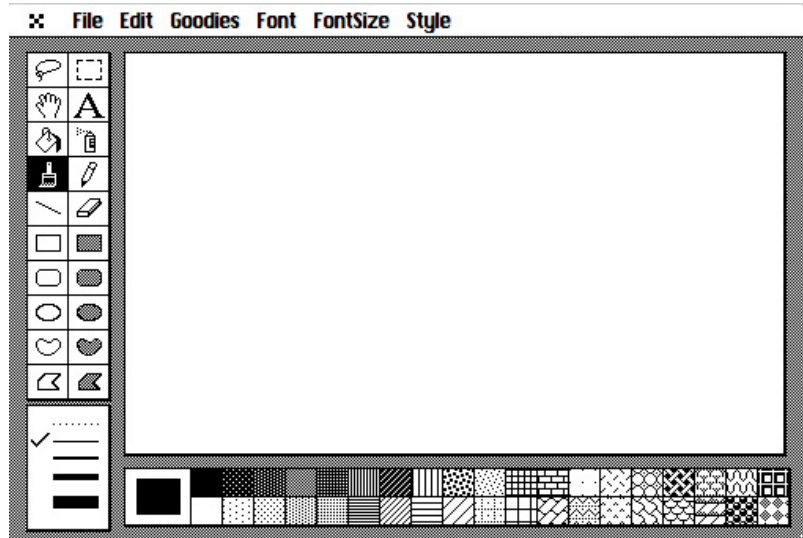
After some discussion and refining the concept, it was concluded that the interface had to distance itself from the generated mosaic, as it would battle for the user's attention. We also concluded that a more standard interface would conduct the user's experience to a smaller learning curve. This is important, as it would allow them to focus on creating rather than finding how to do the tasks or where to find certain buttons. Buttons became more distinguishable, there was a clearer hierarchy, and more importantly: the focus was the mosaic. The colour palette became neutral but had more levels, icons were added, and a system was created (Figure 94). In the next section, we will further explain the guidelines that we set for the design of this interface.



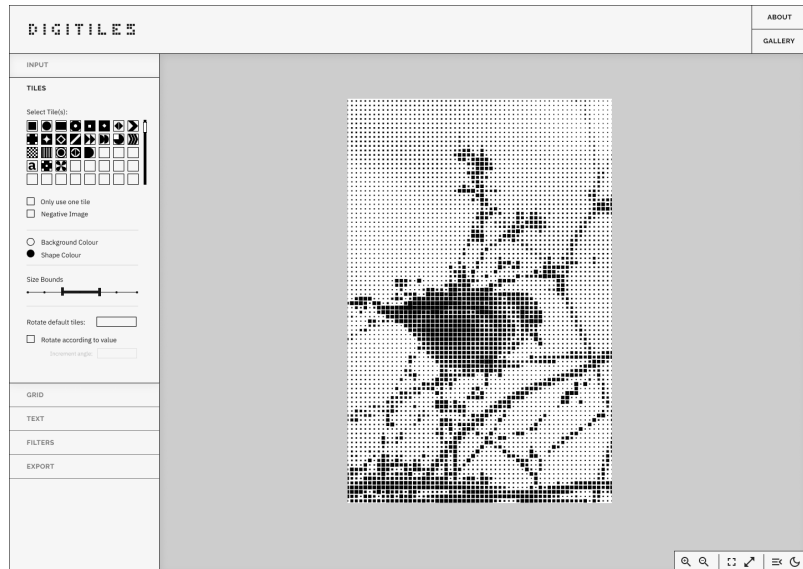
**Figure 91.**

First mockup for the platform's design.

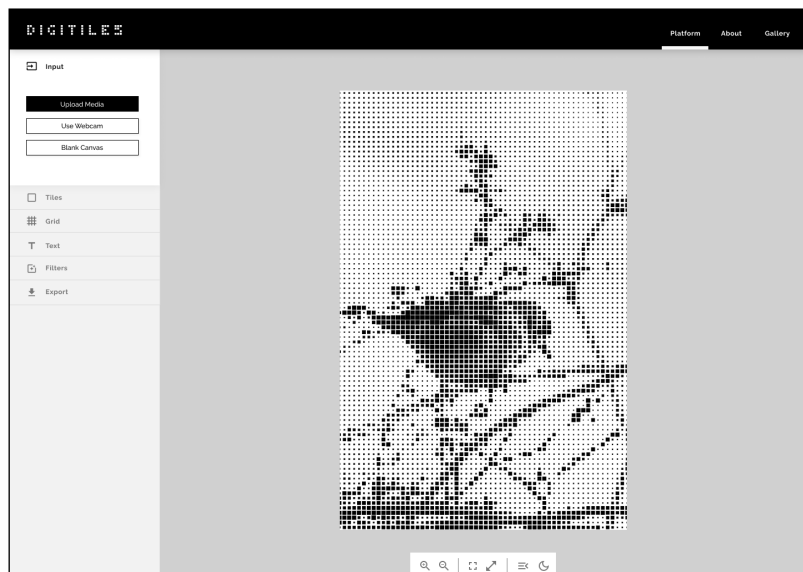
**Figure 92.**  
Screenshot of MacPaint tool.



**Figure 93.**  
Mockup inspired by MacPaint.



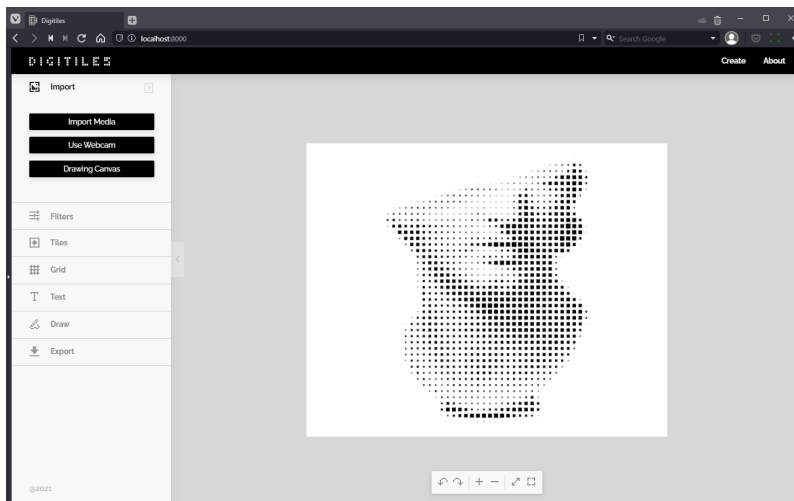
**Figure 94.**  
Final mockup.



## GUIDELINES

The objective of this subsection is to showcase the visual guidelines for the platform and justify the decisions. The design was inspired by other image and graphic editing tools such as Pixlr, Canva, Fotor, Figma, Illustrator and Lightroom. Throughout the development of this interface, Nielsen's 10 Usability Heuristics were always kept in mind and checked.

Before beginning, it is important to understand the panels of the page. On the left side of the platform (Figure 95), there is a menu of grouped features. The first group, called *Imports*, is related to the inputs, such as images, videos, live capture or white canvas. The second group contains functions that process the input with filters. Thirdly, we have the *Tile* panel that groups all parameters related to the design and controls of the tile. Next, there's the *Grid* panel where the user can customise the grid, followed by the *Text* panel where we can find features related to the input of text. Lastly, we have the *Draw* panel that lets the user pick and edit brushes with which to draw, and the *Export* that allows the user to toggle between different formats to export. For more detailed information about the features, these are available in the previous section "Features".



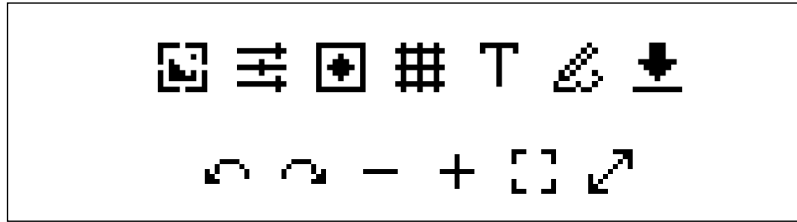
**Figure 95.**  
The final version of the website.

At the centre of the screen, we have the artboard that displays the generated mosaic. This artboard needed to be the focus of the page, meaning it drew the most attention.

Below the mosaic, there is a menu with a set of parameters that control the platform and the visualisation such as *Undo*, *Redo*, *Zoom in* and *Out*, and *Fit Screen* and *Fullscreen*. Finally, at the top, we can navigate to the *About* page.

Several icons were designed to help the user localise the right interaction. These were all designed on the same squared grid as seen in Figure 96. This not only matched the concept but also hinted at the possibilities that can be created from a system such as this. The icons can be found in the menu, and at the bottom centre of the page.

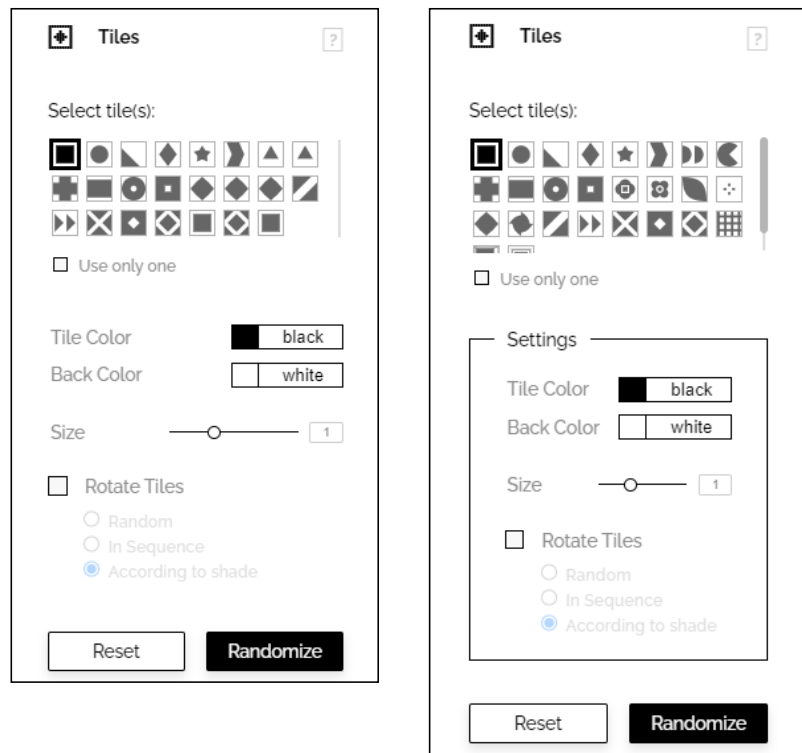
**Figure 96.**  
Collection of icons created for the  
website.

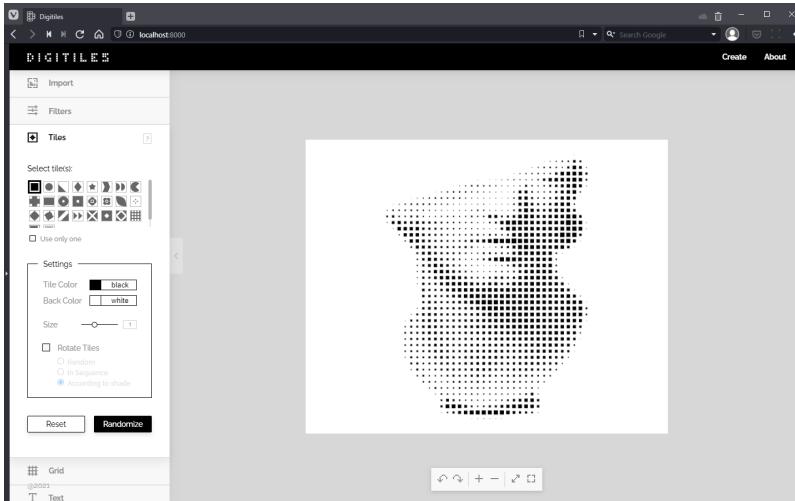


The colour palette of the platform uses neutral tones to not distract from the monochromatic mosaic. Nonetheless, it presents enough contrast to create a hierarchy so the user can distinguish between focused and non-focused elements, buttons and secondary buttons. All the buttons and input fields follow the same visual rules. However, this was not enough to organise busier panels such as the *Tile* and the *Text* panels. Therefore, we will expand on the choices made for these.

As seen in Figure 97, the *Tile* panel allows users to pick tiles and edit their characteristics. However, in the first version, users got confused trying to locate the inputs that were related to the tiles. Following this, we introduced settings boxes. These add another level in the hierarchy by indicating that their parameters relate to items outside of them. In the *Tile* panel, settings change the selected tiles (Figure 98).

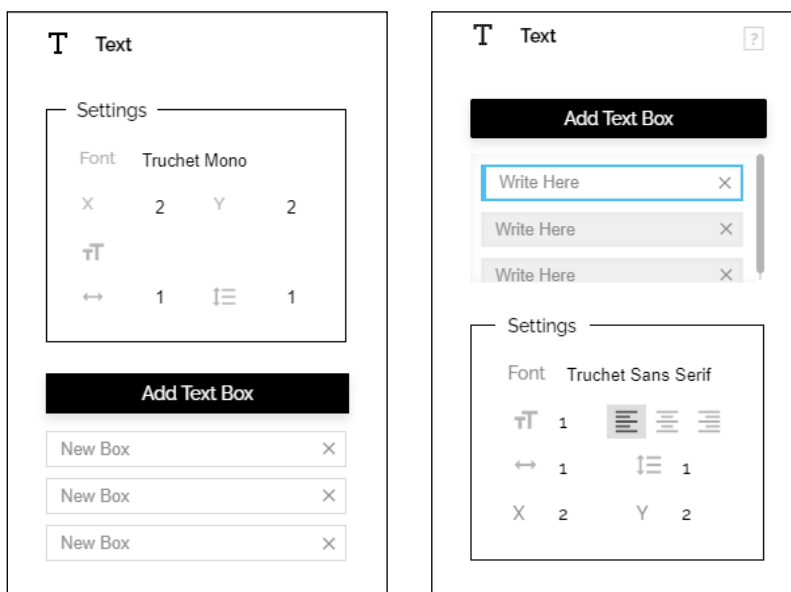
**Figure 97.**  
Close up of the *Tile* panel interface.  
Old version (left) and updated  
version (right).





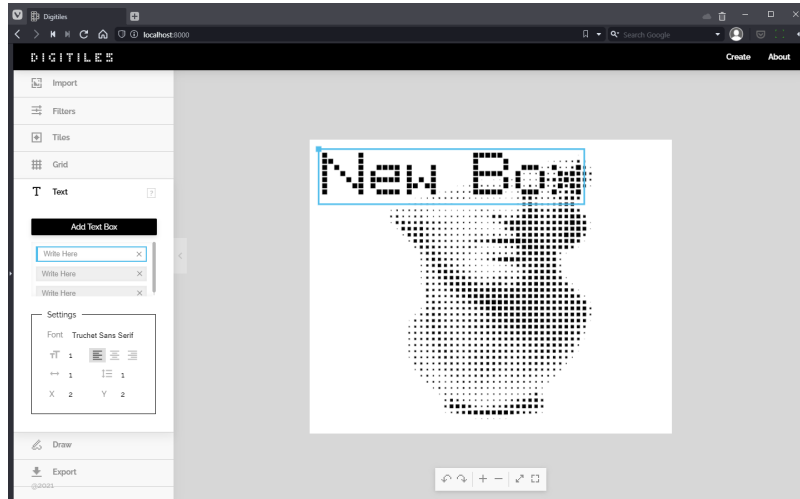
**Figure 98.**  
The final result of the Tile panel on the website.

In the *Text* panel, a similar problem occurred where the user could not distinguish the box they were editing (Figure 99) after adding several text boxes. Therefore, we added the same settings box and also highlighted the selected text box. When a new box is selected, the values of the settings box change to display its settings. The blue highlight of the input also matches the outline of the text box in the artboard, helping the user make the connection (Figure 100). Initially, the button to add a new text box was below the settings but, since users could not find it, we ended up placing it above the settings.



**Figure 99.**  
Close up of the *Text* panel interface. Old version (left) and updated version (right).

**Figure 100.**  
The final result of the Text section.

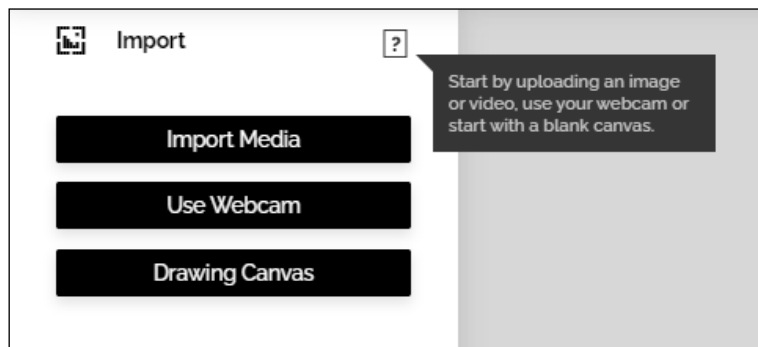


In order to help users understand the features, tooltips were placed in most of the features. They can be accessed simply by hovering over the label or the tooltip signs left on the top right corner of each panel( Figure 101).

Finally, to tie all of the pieces, we created a logotype for the platform. It revolved around the simple concept of using the tiles to write the website's name, making for an easily dynamic piece. The logotype can be seen in the top left corner of the website, and it also appears on the loading screen. In Figure 102, we can see the final result for the logotype.

Due to the extension of the platform, some improvements that would elevate the usability were planned for future work. For example, letting the user navigate the system with shortcut keys, communicating the system status when there's an error and improving the system's flexibility in general.

**Figure 101.**  
Tooltip for the *Import* panel.



**Figure 102.**  
Logotype for the platform *Digitiles*.

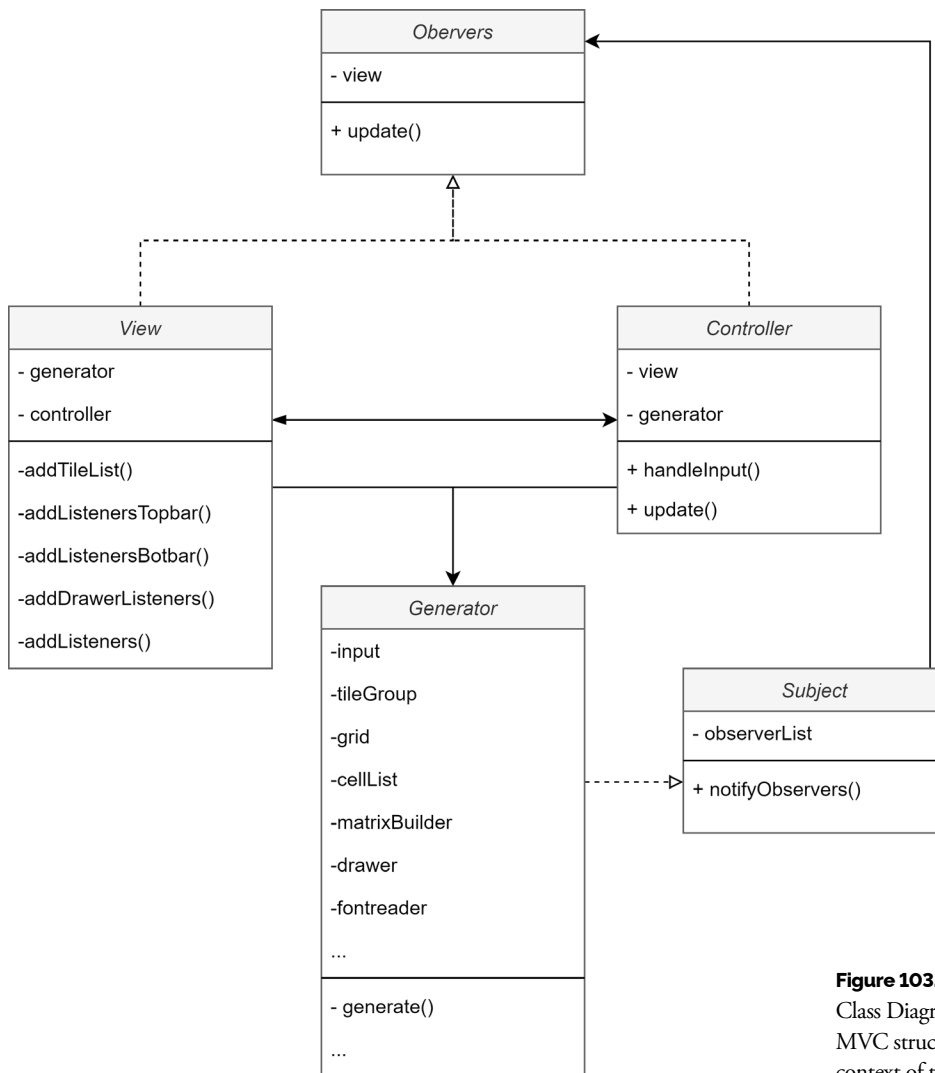


## 4.2.5 Architecture

Throughout the project, and with the addition of new features, the architecture suffered several restructures. Due to the number of features and iterations, and since most objects have a similar structure, we will not explain all the architecture in this section. Instead, a general picture and important points will be highlighted. To better visualise the architecture, this section also provides class diagrams.

The goal of this architecture was to make the final structure of the system as modular as possible so it would be easier not only to adapt it for different purposes but also to do it without destabilising the current system.

To achieve this, the system starts with a simple *Model View Controller* Pattern in which we have three main objects: *View*, *Controller* and *Generator* (Figure 103). The first one deals with changes in the website's appearance (what the user sees). However, it does not deal with the drawing of the generated artefact. The *Controller* deals with the communication between *View* and *Generator*, by receiving inputs from the first one and ordering

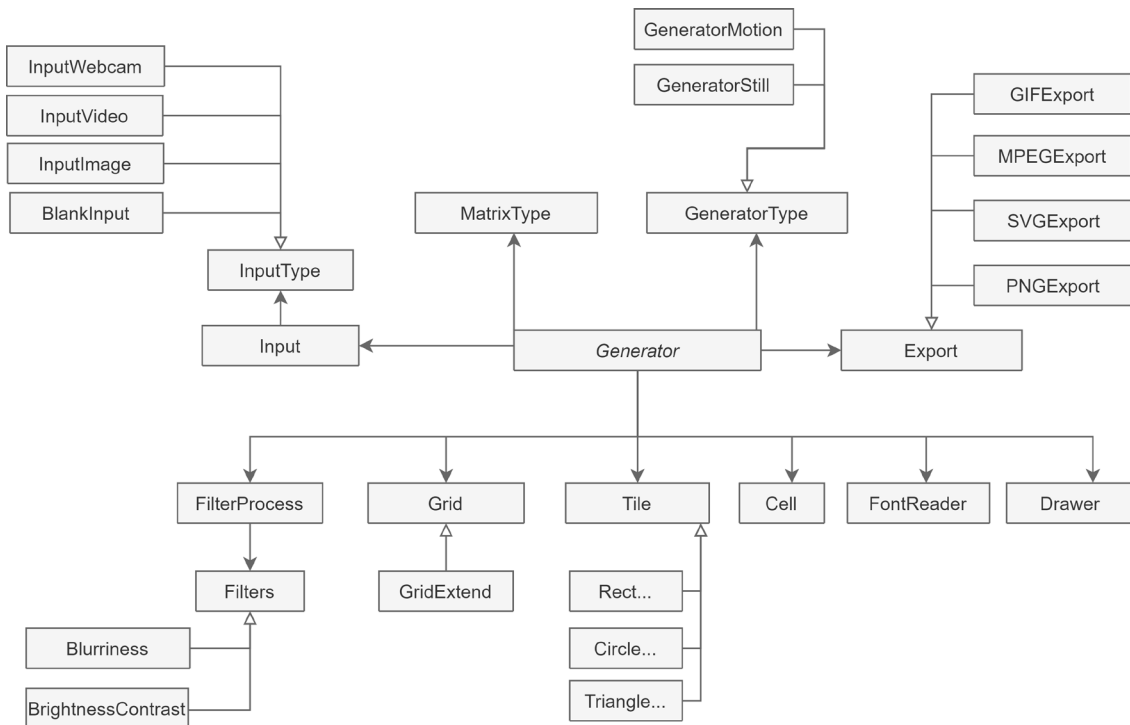


**Figure 103.** Class Diagram (simplified) of the MVC structure being applied in the context of the project.



the *Generator* how to respond. Lastly, the *Generator* manages all of the information necessary to generate the mosaics.

By zooming into the class, we see that the most complex structure is inside the *Generator* object. Different objects that contain varied parameters are managed and accessed through the *Generator*, such as *Input*, *Grid*, *Tiles*, *Cells*, *MatrixBuilder*, *GeneratorType*, *Filters*, *Export* (Figure 104).

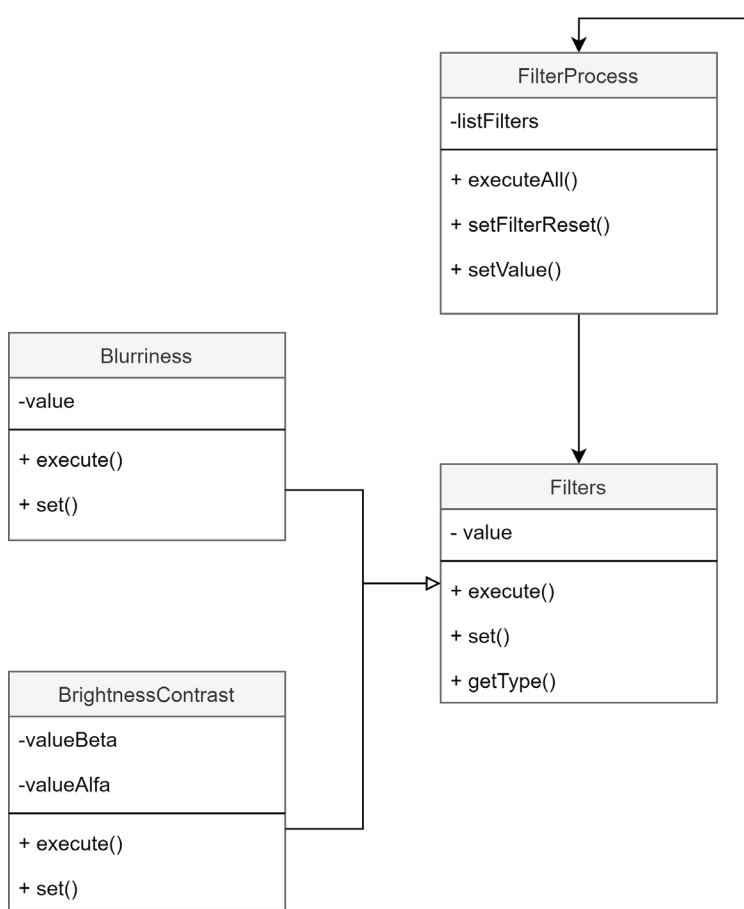


**Figure 104.**

Class Diagram for the structure of the *Generator* object.

Most of these objects follow a State Pattern to switch their behaviours depending on the parameters chosen by the user. For instance, the *Export* has several states such as *PNGExport*, *SVGExport*, or *PDFExport*; the *GeneratorType* has *StillGenerator* state and *MotionGenerator*. It is through the *Generator* that we can switch those states depending on the input.

We also needed to have objects that managed groups of objects with similar characteristics. Therefore we can find the Composite Pattern in some parts of the architecture. For example, the *Filter* object groups and equally treats components such as *Brightness*, *Contrast* and *Blurriness* and consequently make it easier to add new ones if desired (Figure 105). In this case, all of these components have the same function, however, their behaviour is different.



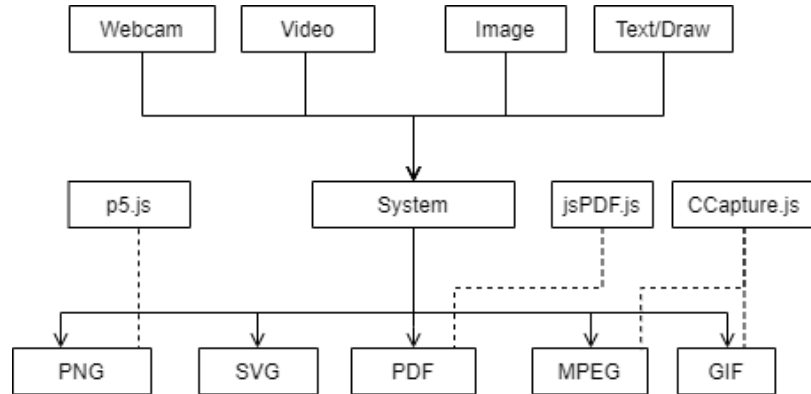
**Figure 105.**  
Class Diagram for the Filter components.

To generate and draw the results on the canvas, the *GeneratorType* was created. This object receives the needed parameters to draw the final result. Instead of looping the drawing only to display the same result (when dealing with still images), it would be more optimal to do it only when the user changed something. However, we also had to deal with videos and live capture that required calculating the values and drawing the results constantly. Therefore, the system has two strategies that inherit from *GeneratorType*. When using static images, the system creates a *GeneratorStill* object and executes it once. When the system deals with live capture or videos, it simply changes strategy by creating a *GeneratorMotion* object. When executed, this last object loops the drawing function.

Finally, since the system uses some libraries and has several forms of input and exports, a pipeline for the generation of the final result can be seen in Figure 106. This representation shows the external libraries that we used and where they are used throughout the generation of the output.

The last remark towards this architecture is that it is not as optimised as it could be. This could be a reflection of the constant introduction of new ideas. Resources were lost into adapting it to fit these ideas and allowing it to be easily expandable for the future. This event may also reflect the exploratory character of the work and the earlier confusions concerning the goals we were trying to achieve.

**Figure 106.**  
Pipeline for the generation of  
outputs.

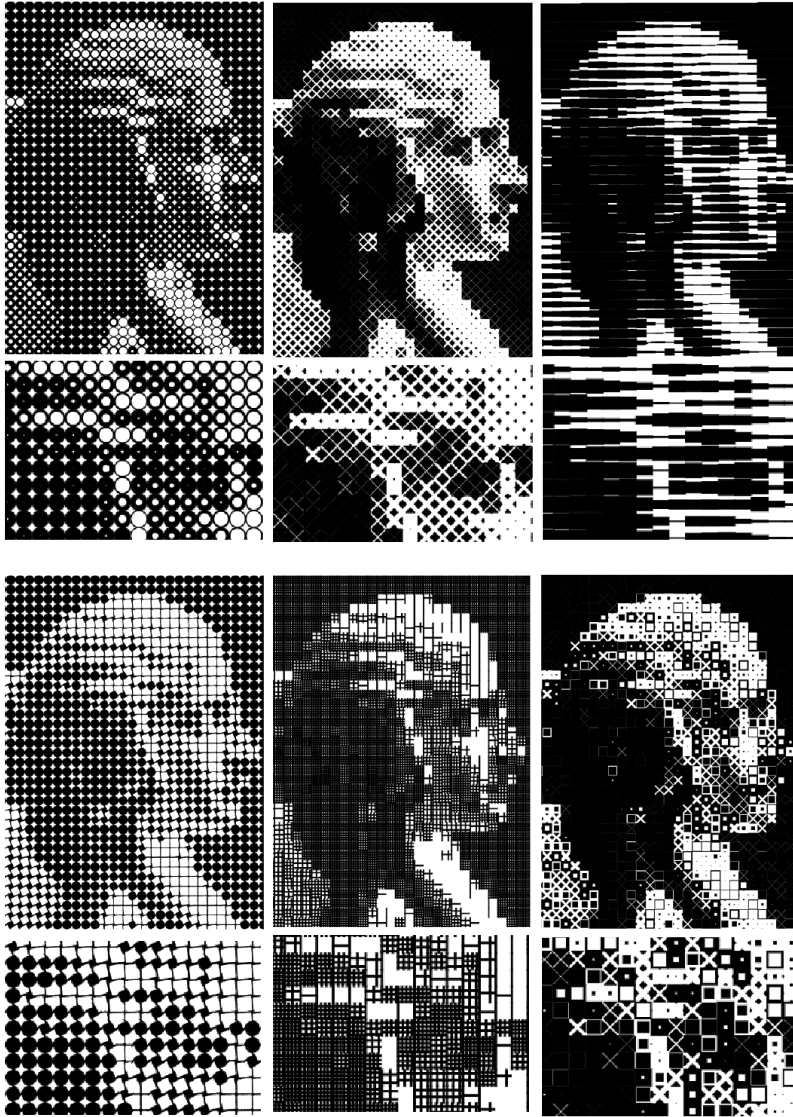


## 4.3 Output

### 4.3.1 Parametrization

Although one of the goals is to describe the system, it is also relevant to explain the implication of its different forms of usage on the outputs. Therefore, this section contains considerations regarding the application of the system, when approaching its parametrization of different functionalities and how this influences the results. It is important to study this perspective to obtain the results closest to the desired outcome and understand which situations one can take advantage of the choices made.

Firstly, by varying the tile the user can change the stylisation of the image (Figure 107). The choice of the tile can also add context to the image. This said context can come from an abstract concept: the tile can be a dynamic logo or symbol, therefore, connecting all the images composed with this tile to a graphic identity or idea..., or it can also come from a physical limitation, for example, the resolution of a screen in which the tiles must be made of rectangles because that is the screen's unit. The tiles are the component that allows the system to encapsulate the capacities of representing visual coherence (by applying the same one to different images) and diversity (by applying different tiles). As mentioned before, more than one tile can be selected to be used in the same composition.



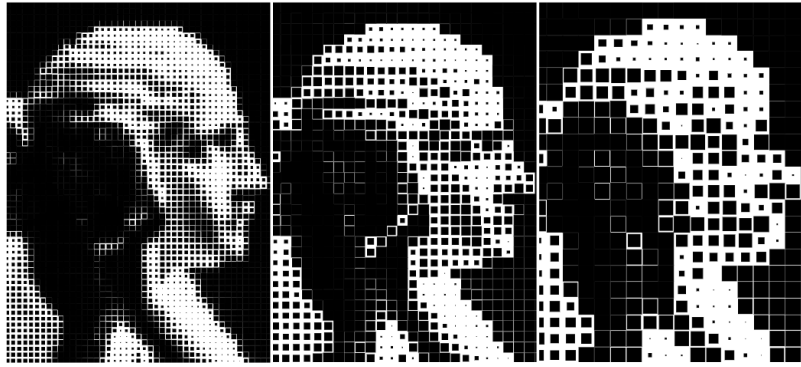
**Figure 107.**

Images generated using different variable tiles. In the last example, more than one tile is selected.

Secondly, by varying the resolution of the pattern, the resulting image can have different levels of detail. The higher the resolution, the closer the output is to the input, meaning the image becomes more recognisable. This is useful when handling images with more than one element. The lower the resolution, the less recognisable the image gets in comparison to its original, by ignoring and erasing a lot of detail that could be essential. This also means the attention will be dragged to the tiles themselves and their form. This last format works better with simpler images (Figure 108).

Different images produce, therefore, different success levels of readability. If an image, such as a landscape, has a lot of small details, it would need more grid resolution to be recognised as such, than a simpler composed image by, for example, an object against a white background.

**Figure 108.**  
Images generated with different grid  
resolutions.



There is another added layer of possibilities by varying the tile and background colours. This offers more aesthetic options and new opportunities to create more contexts (such as visual identities) by combining different shapes of tiles and now, colours (Figure 109).

Although still monochromatic, the colour option opens possibilities, for the future, to overlap different images with different colours, and to work with different values besides brightness.

More parameters are available, such as tile size which alters the contrast of the image. Regarding rotation, this parameter can make the pattern more interesting and also help the user recognise the image faster (if the rotation is based on the weight of the tile).

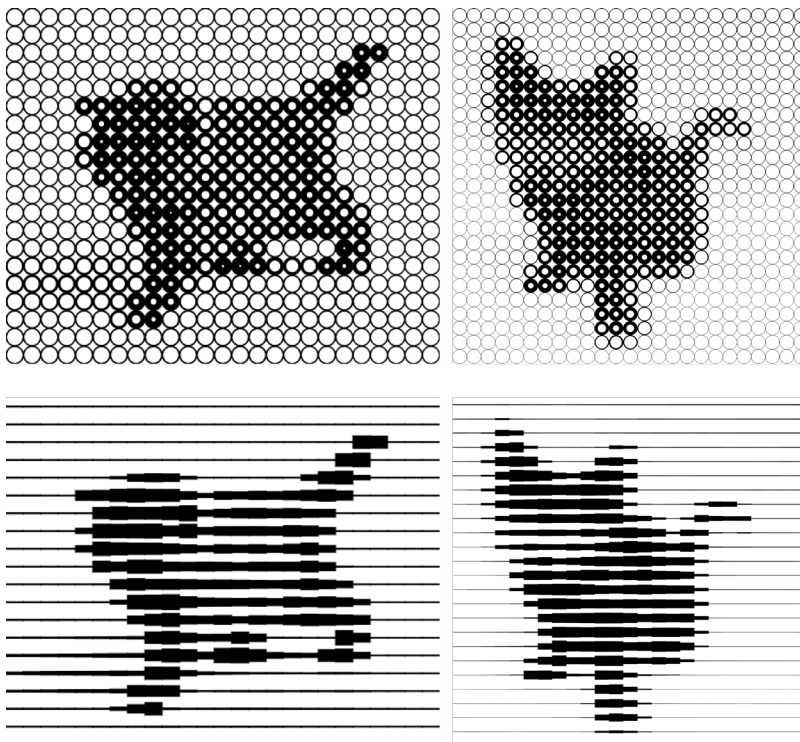
The system can also be used to either achieve visual diversity or visual coherence. This means that it can generate different patterns using the same image by exchanging parameters, such as chosen tiles and grid size, to achieve diversity. However, it can also achieve coherence by giving the same parameters to different input images, making the results share the same style (Figure 110). As it is, the system can represent any tone in an image, but this does not mean it represents them all. This is also limited by the imposed resolution that we want to work with. Nonetheless, given a very high resolution, the system will smoothly portray the tonal values. This will affect the perception of the whole image.

**Figure 109.**  
Images generated with different  
colours (top) and a close-up of the  
tiles (bottom).



The images generated demonstrate the system's ability to create an illusion of a range of continuous tones and, consequently, patterns with the underneath image being recognisable by combining all these factors. As said before, this capacity to recognise the image is enhanced when the grid has a higher resolution, however, the tiles will not be so easily perceived as individual elements (unless seen up close or zoomed in). In the case of wanting to display or share the results, for example, on social media, this becomes important. Since the image may be displayed on a phone screen, this means there will be bounds and limits to where we can perceive the image and the tile. Nonetheless, both represent different paths that fit into different goals.

The outputs can be exported as a rasterized image (PNG) or a vector image (SVG). Since the user can adjust the image resolution as well as the grid resolution, the results can either be exported with high, medium, or low quality. These adjustments help to control the result to better suit different situations and purposes. For instance, the user may want to print a wide poster or make a small avatar for a website. In this way, they can be used for different purposes. The possible applications of a framework as such go through representing any type of visual artefact, from still filtered image mosaics to video animations, modular visual installations, generating modular typefaces. The possibilities can look immense, and that is why it was relevant to analyse the behaviour of different patterns.



**Figure 110.**

Images showcasing how the platform can portray both diversity (vertically) and visual coherence (horizontally).

### 4.3.2 Patterns

To compare the resultant patterns, we present a table (Table 1) where the horizontal axis represents an increase of the grid resolution and the vertical axis presents a variation of the tiles. This list represents only a sample of the available tiles and variations in the grid.

Looking at this table, it is clear how some designs result in patterns with higher contrast such as examples F and H, while others result in a lighter pattern, such as C and B. This is an indicator of which tiles can be combined: lighter tiles work better with similar and the same applies to heavier tiles. From this sample, we can also see how easy it is to get diversity of patterns by analysing it vertically.

Since the patterns have different results when applied to text, we also show results with typography through a similar table presented in Table 2, where the horizontal axis present variation in font. In this case, we can see that the same logic of lighter and heavier pattern applies, however, without affecting as much our ability to recognize each glyph.


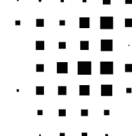




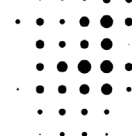
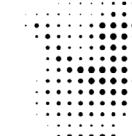



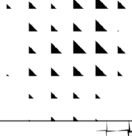
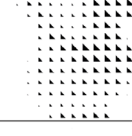

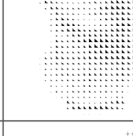
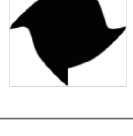
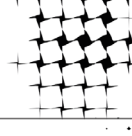
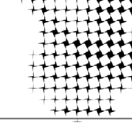
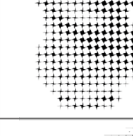






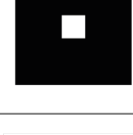
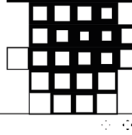
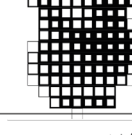
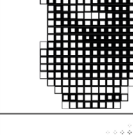
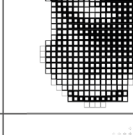

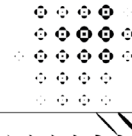
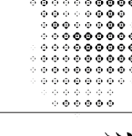
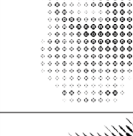
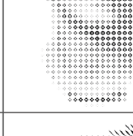
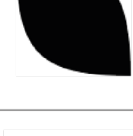





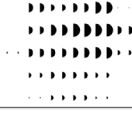
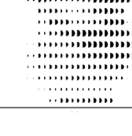

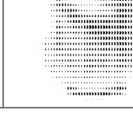
Moving to individual compositions, by looking at Figure 111, we can see how images with simpler compositions and with more balanced levels of black generate better outputs. Contrast and exposition play an important role in the quality of the result. Furthermore, having a more known figure or an isolated and distinct object in the input image also increases the success of the output.

The system also generates interesting outputs that are only based on typography (Figure 112). Although geometric and with a pixelated style, the results easily change their character when using other tiles (Figure 113).

Lastly, we have an image of the swiss flag made of a grid with cross modules (Figure 14). This showcases how the system can be used for a specific cultural space by appropriating a symbol and adapting it into a variable tile. Besides cultural symbols, the same could be done with logos or glyphs. However, this output also shows how images of landscapes do not work as well as isolated figures. Nonetheless, by adjusting the image we managed to get a good result. Furthermore, we are able to combine it with typography in a harmonizing way since both share the same grid and tile. More examples can be consulted in Appendix C.

**Table 1.**




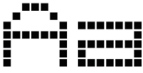





















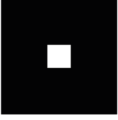



















Table with a sample of the results using image:vertical axis presents the change in the tile, while the horizontal axis presents the change in the number of columns.

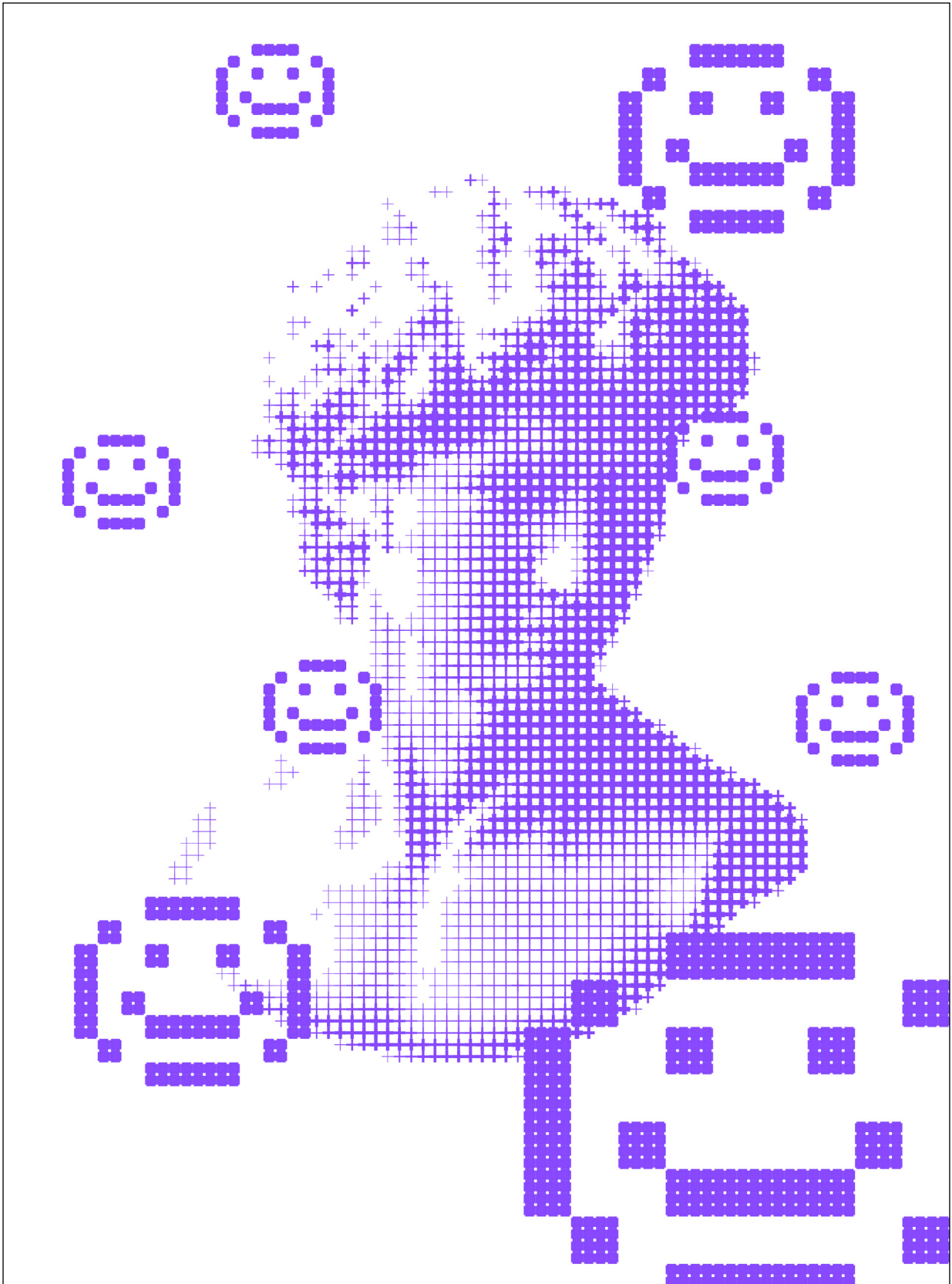
	1	10	20	35	50
A					
B					
C					
D					
E					
F					
G					
H					
I					



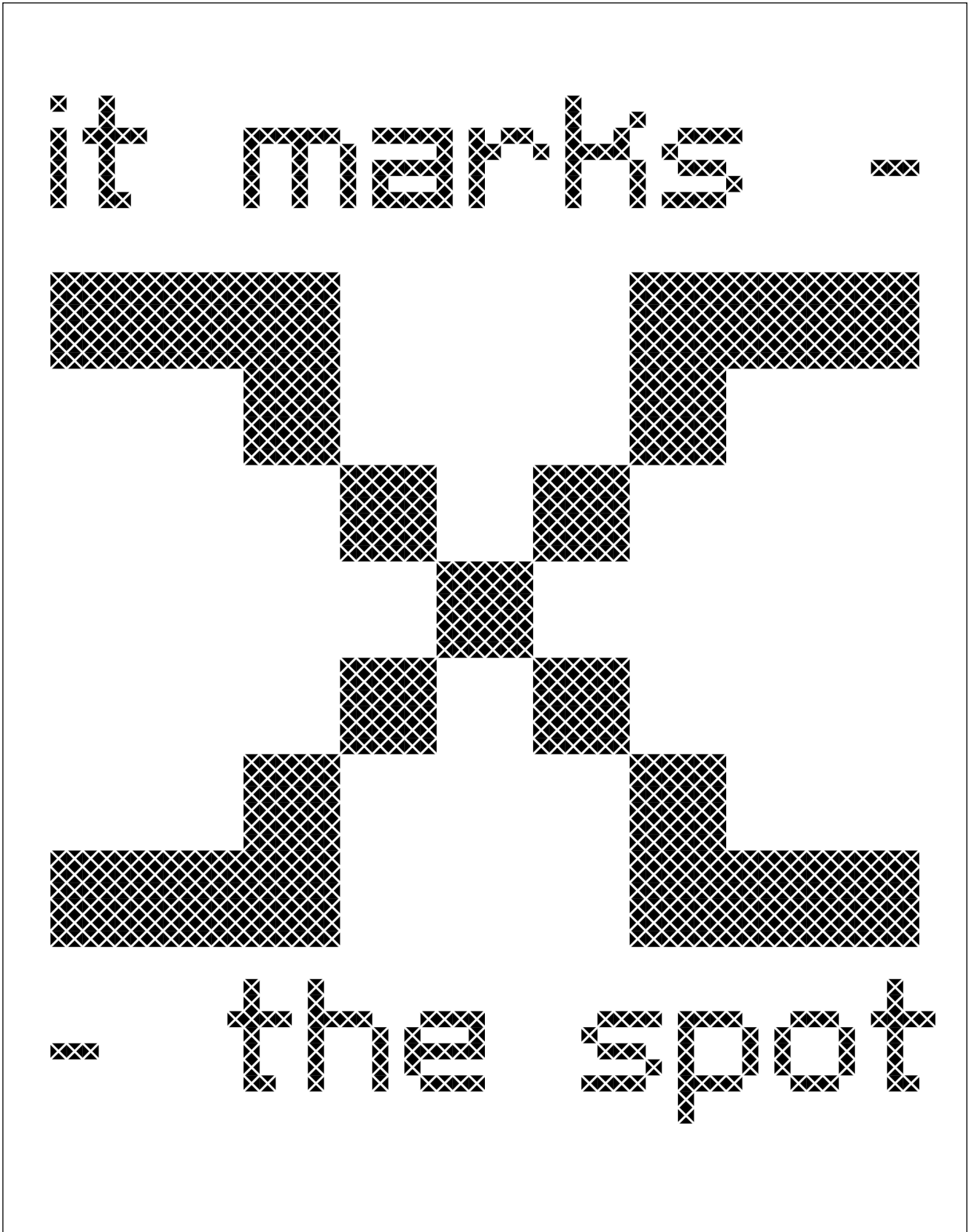
**Table 2.**

Table with a sample of the results using typography: vertical axis presents the change in the tile, while the horizontal axis presents the change in the typeface.

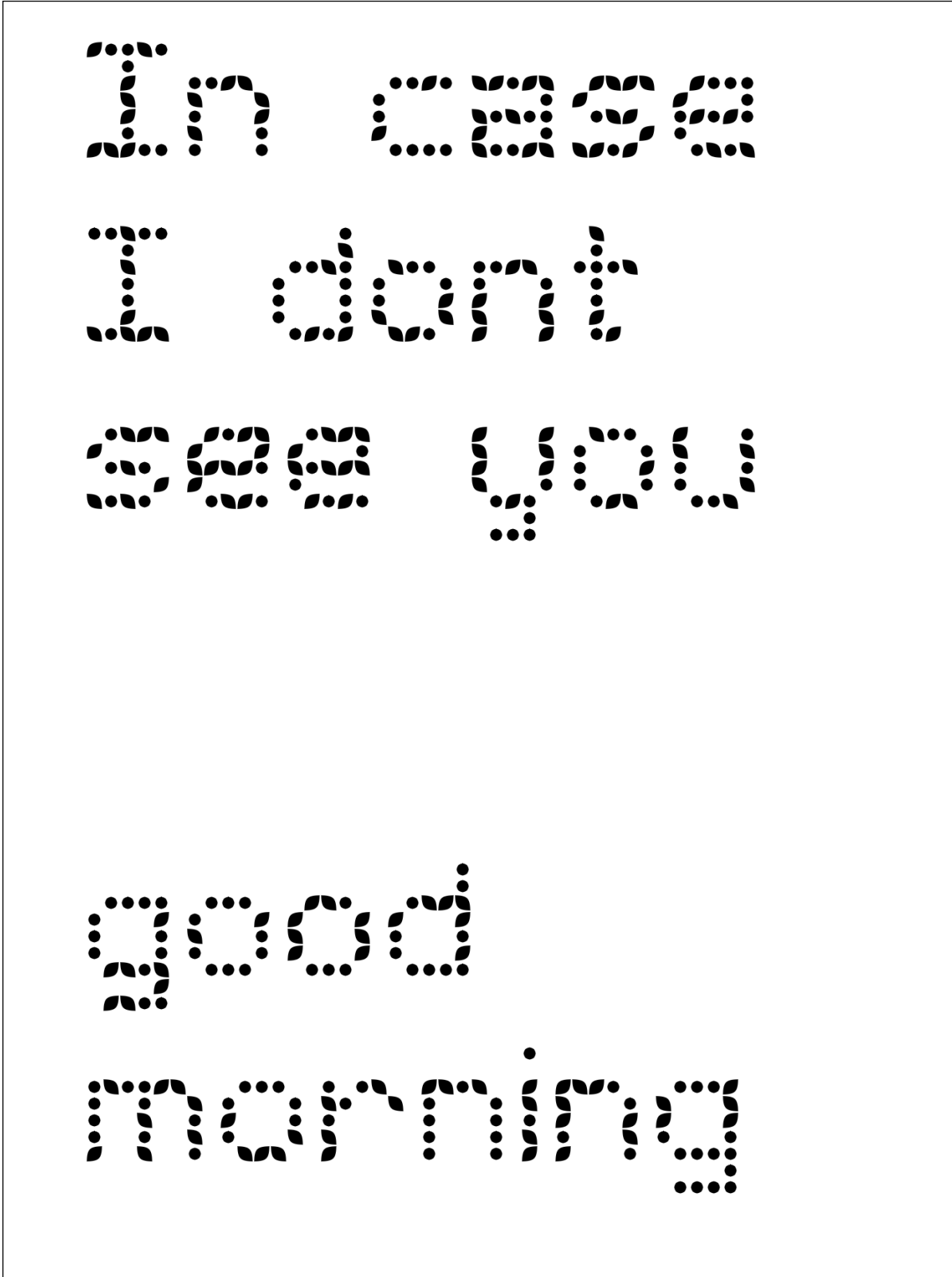
	Tile	Sans Serif	Serif	Mono	Small
A					
B					
C					
D					
E					
F					
G					
H					
I					



**Figure 111.**  
Poster generated mixing image and symbols  
created for the system.



**Figure 112.**  
Poster generated using only the typefaces  
created for the system.



**Figure 113.**  
Poster generated using typography and two  
different tiles.



**Figure 114.**  
Poster generated using a crossed Tile to  
portray the swiss flag.

## 4.4 Evaluation

The evaluation phase took place throughout the final sprint of development when some indecisions about the visual interface were still present but the product was stable enough to obtain feedback. Evaluating the project was crucial for its success: it helped test its concept and quality of execution. To explain this phase of the process, we will outline the profile of the audience, then we will establish the goals for the tests, and, finally, we will expose the procedure developed for evaluating and analysing data that was gathered.

### AUDIENCE

To better establish the context of the tests, is it important to explain in which socio-cultural context this system is placed. As previously hinted, the platform was developed for designers and artists with experience using image editing tools. Their intentions should be to generate, from varied inputs, artefacts located between the visual, artistic, and digital world.

Additionally, we also envisioned the system to be able to adapt to different cultures and spaces. These can be physical and public spaces, like urban streets, museums, or galleries, where installations can be set. It also fits private and virtual spaces such as websites and social media, where users can share the generated results. The user may adapt the parameters to elevate the qualities of the space it is being used in. For example, changing the grid to fit a smaller display, using new tiles with a certain symbolism, or using a set of colours to fit the theme of an exhibition. The framework can be used in other possible and imaginable media contexts.

Finally, it is important to mention the people that encounter the generated artefacts. These can be the visitors of the built physical spaces such as installations or they can be the ones who find digital artefacts generated with the system, such as a post on a social network. This group accounts for an important part of the evaluation of the developed framework. While designers and artists provide a perspective of the product they are using, pointing to the effectiveness of the system, architecture, and bugs, the viewers' perspective centres on the quality of the artefacts. This provides validation of the proposed concept and gives pointers to improve the efficiency of the program.

### GOALS

Different techniques were used to evaluate the work's usability. We applied them to better understand the context, guide the design towards the usage goal, and compare design solutions. More specifically, we intended to test parameters such as efficiency, effectiveness, performance and speed, quality and robustness of the interface, and evaluate the concept. The testing phase

also allowed us to detect dysfunctions in the platform and resolve usage problems along the process.

Concerning the visual interface, we wanted to discover if the users learned how to navigate through the platform, if the users agreed with some imposed restrictions, how the user should insert and edit text and if the user could understand all the icons and to what function they corresponded.

Apart from this, it was also relevant to know how valuable the concept appeared to the user, if it was perceived as being well executed and if it was effective, meaning interesting artefacts were easy and quick to generate.

Among the techniques used to evaluate we picked lab tests, which allowed us to gather data. A heuristic evaluation could also be conducted in future work with more evaluators. This would involve having them analyse the system's interface and judge it based on a set of recognisable principles called "heuristics". Although these heuristics were taken into consideration while designing the interface, a closer examination of the result with more than one person would be impactful.

## USER TESTS

After finishing the design of the platform, it was necessary to test it with potential real users. These tests would provide relevant feedback, find errors and understand if the website had good usability.

The initial intention was to apply iterative design and testing to make informed decisions and to better understand the users' needs from the beginning. However, only one wave of tests took place over the final development of the project. This wave had ten people testing the platform. Although the majority of people matched the target audience, there were also experiments with people outside the target audience. The goal in doing this was to test the robustness of the interface and test how hard the curve of learning would be for someone who was not aware of the vocabulary used in the area of Design.

The tools that were used during these tests included Google Forms, Zoom and Google Sheets. The first one allowed us to share the User Questionnaire and to easily organise the information collected. The second tool (Zoom) provided better conditions to reach a bigger pool of people and also to easily record the tests for a second analysis.

The tests' preparation included establishing questions that reflected on doubts about the interface, the execution and the concept. More specifically, the questions queried about the efficiency of the platform, its speed, the quality of the interface, the quality of the resultant artefacts, the value of the proposed concept and if the context and user profile were well established. The final questionnaire was a mix between Likert scales (where users have to pick between a 7 point scale) and development questions. After having the questions, we prepared a set of tasks that the user would have to perform and consequently would conduct them throughout the

test (for more details on the tasks, see Appendix B). These tasks tried to test the most important features as well as clarify the doubts. The preparation also involved preparing worksheets to collect notes and results.

The procedure for each user test began with a brief explanation of the session and gathering consent for recording it. Next, the users followed the tasks to test, and they were incentivised to think aloud as they were performing them. In the end, they filled the form and were queried about the parts of the test where we observed problems.

Different types of data were collected, such as the number of errors per task, time to complete a task and the number of users completing a task. These were extremely useful to solve urgent errors in usability. Several small errors were found which were fixed immediately after the first wave of tests. In the next section, we will expose the errors that were found and fixed as well as the results for the user questionnaire. Following this, we'll analyse these results.

## **RESULTS**

This section begins by exposing the errors that we found throughout all the tests. The full list of found errors can be found in Appendix B where we can see their description and urgency. Most problems were not urgent and were easy to solve. However, some issues were more complex, such as the pencil not drawing the whole path, which required an algorithm to calculate the tiles between the last detected tile that was mouse hovered and the current, or the interface of the Text panel.

There were also problems with optimization. There were too many input calls when the user tried to change the colour input by dragging the cursor, and too many drawing calls when trying to generate the result. These were fixed by replacing the colour input with one that only changed colour on the first click and by calling the drawing functions when a change was made.

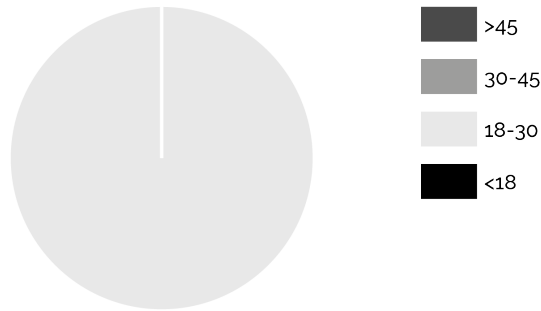
After correcting the errors, the answers to the user questionnaire were studied. To better show the results of the user questionnaire, the answers to each question were organised and displayed in graphics.

The first phase of the test helped to recognise the users' profiles that tested the platform. All users were between 18 and 30 years old and had some experience with image editing tools. Regarding the areas of study/work of the users, 70% belonged to Design and Multimedia and Art. This meant that the majority of the users that tested the website matched the intended user profile (Figure 115).

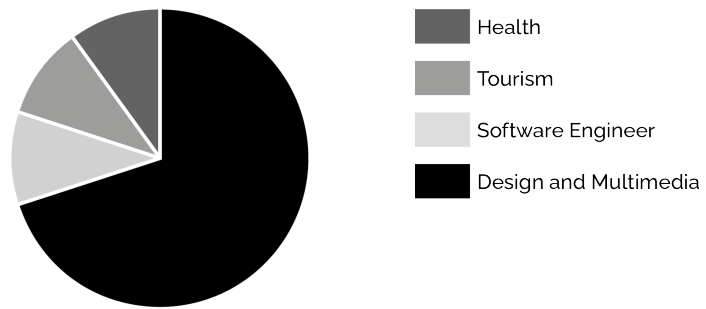


**Figure 115.**  
Distribution of users' answers regarding their profile.

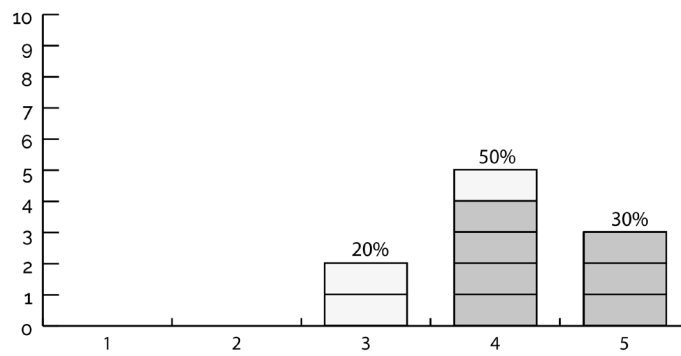
**Age Group**



**Area of study/work**



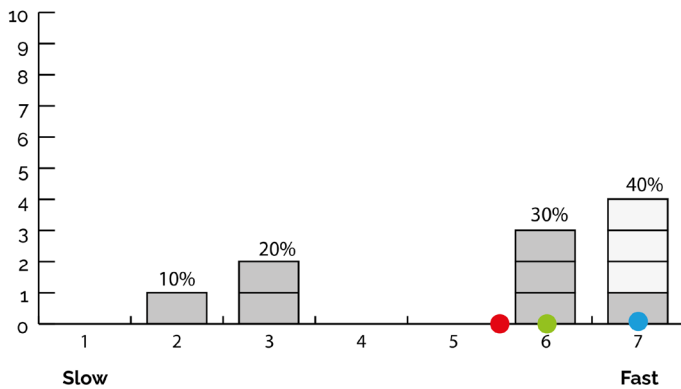
**Level of experience with image or graphics editing tools.**



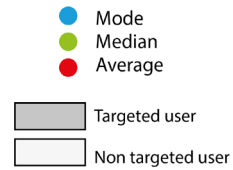
When questioned about the speed of the platform, the answers were divided. In the first four tests, some errors were recurrently found, and before continuing, they were corrected, therefore improving the results on the last five tests. Due to this, we can see the clear division in the presented graph (Figure 116). The answers that were given with 6/7 values were from users that tested after improving the platform.

Additionally, a wide variety of browsers were used during the tests, however, no correlation was established between the performance and the used browser.

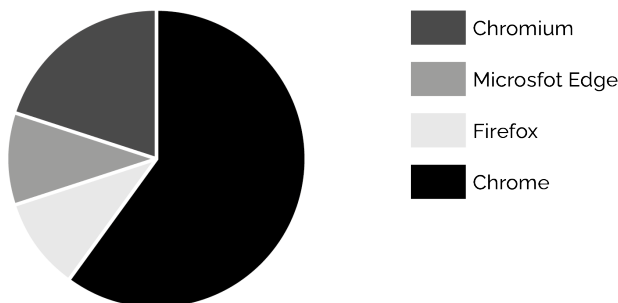
**The platform runned:**



**Figure 116.** Distribution of users' answers regarding the performance of the website.



**Browser used during the test**



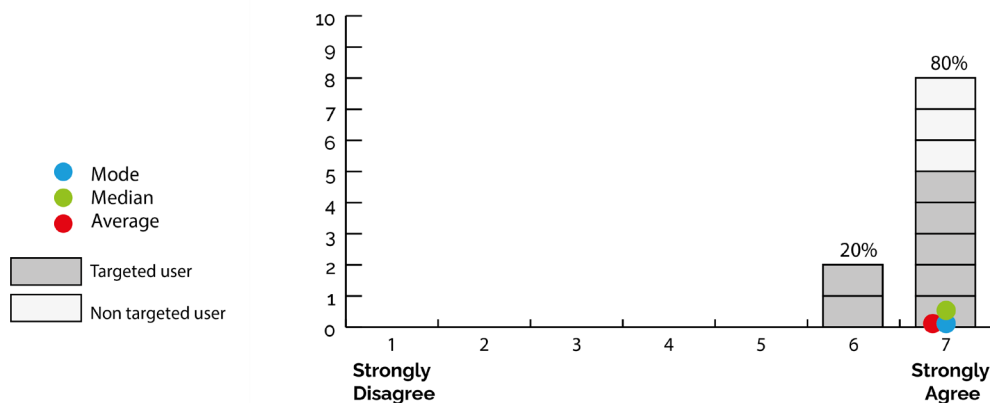
The second part of the user questionnaire intended to discover the interface's usability. In this part, the most frequent suggestion was about the Text interface. People tried to drag the text, and not only this but they would also be confused on how to select a text box and how to edit its text. Nonetheless, the ratings were positive (Figure 117) and users agreed that the platform was easy to learn and conductive. They also agreed that they could understand all the system's features and how to use them.

An option to write which features were not understood was left in the questionnaire. In this option, only two users filled in the answer. The answers were "Editing the text inside the text boxes" and "Blank Canvas".

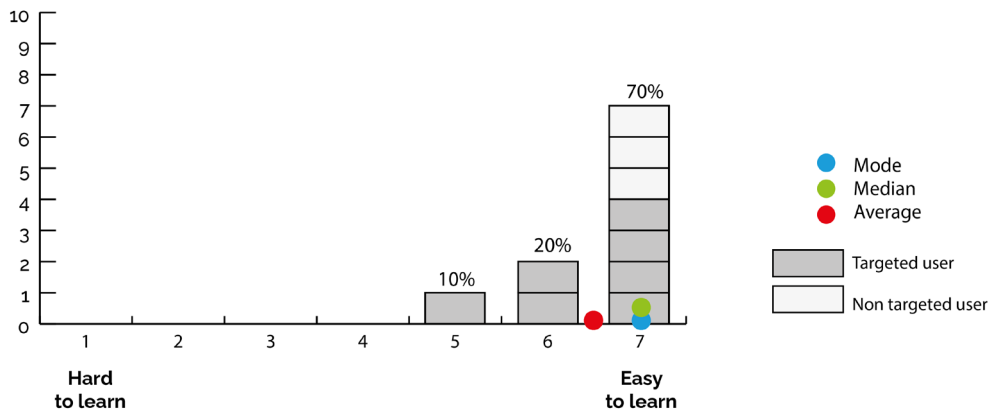
The first answer refers to the moment when a user did not understand how to edit the textbox text. They first tried to click on the canvas. This occurred two times, which indicates that in future work, a more standard interaction with the text boxes should be programmed, allowing the user to double click and write. The second answer refers to the label on one of the buttons on the inputs which previously was "Blank Canvas", meaning the user could choose to create an empty artboard. The user, however, thought that this cleared the text and drawings on the screen rather than creating a white artboard. As suggested by the same user, by replacing this label with "Drawing Canvas", future users could automatically picture not only the white canvas but also its intention.

Other suggestions were given throughout the tests, however, they were not written in the answers to the questionnaire. One user suggested it would be really useful to be able to take a snapshot or "pause" the live capture and video. This would allow the platform to run faster while editing instead of having to generate the video in real-time. Another suggestion was to add the option for the user to create their tiles. However, this feature would imply a lot of restructuring and research to be stable. Some suggestions also indicated the users wanted to see more tiles. To conclude the feedback on the interface, users agreed that it looked attractive.

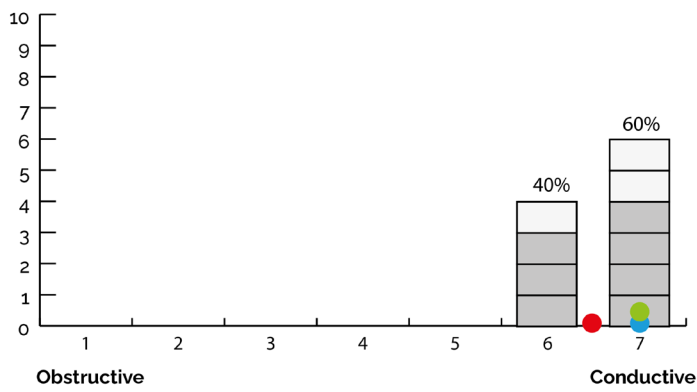
**Figure 117.**  
Distribution of users' answers  
regarding the user interface of the  
website.



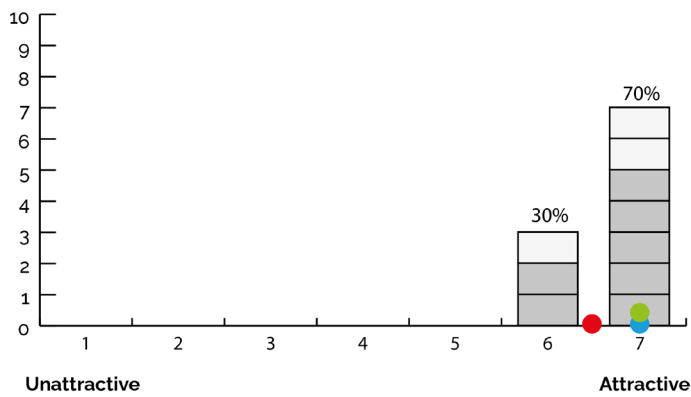
The navigation through the platform was:



The navigation through the platform was:

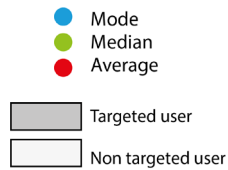


The interface looks:

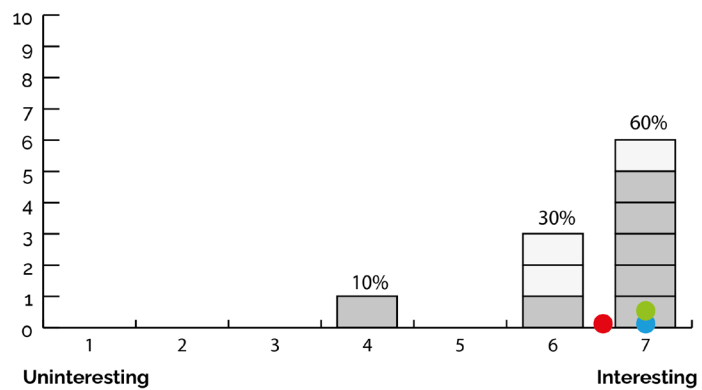


Concerning the effectiveness of the platform, the user was also asked several questions (Figure 118). The first one questioned whether the tool was capable of producing a new interesting artefact from any visual input, to which the users agreed. They were also queried on whether the process of creating a new artefact was fast or slow. To this question, 90% of the users answered above 6, however, we could not figure out the reason behind the 5 given by one of the users. The third question asked if the results were interesting, and once again, users answered very positively.

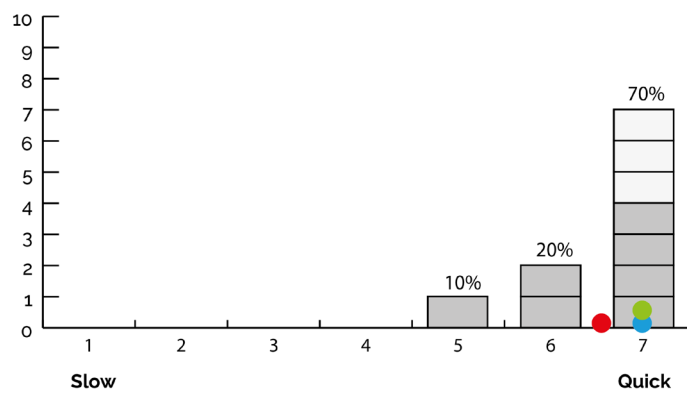
**Figure 118.**  
Distribution of users' answers regarding the effectiveness of the platform.



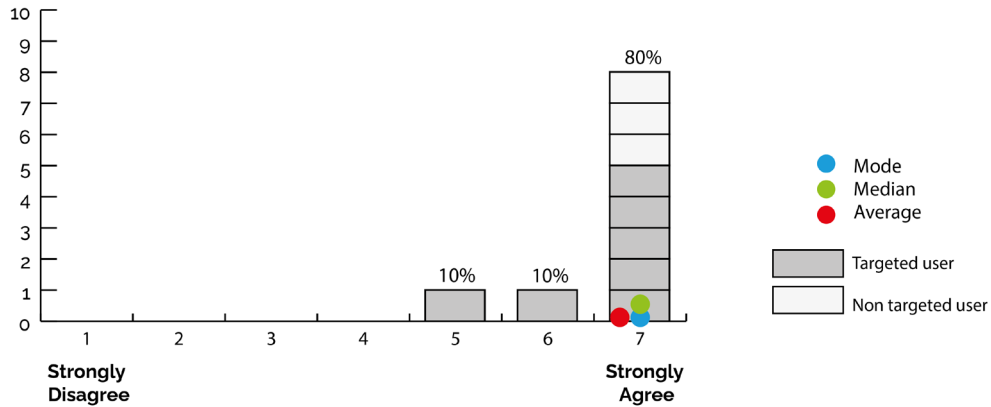
**The generated artefacts look:**



**The process of creating a new artefact was:**

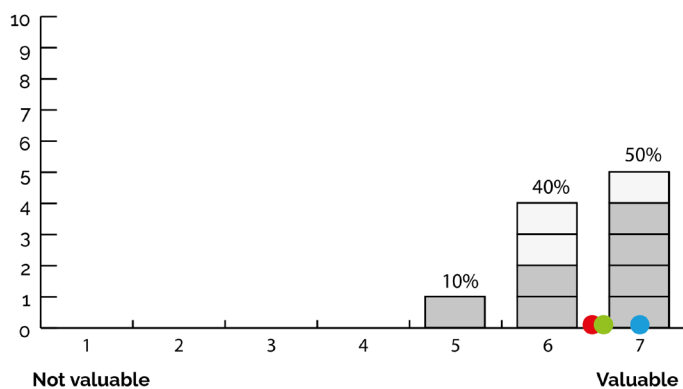


**This tool facilitates the creation of a new visual artefact that is interesting from any image.**



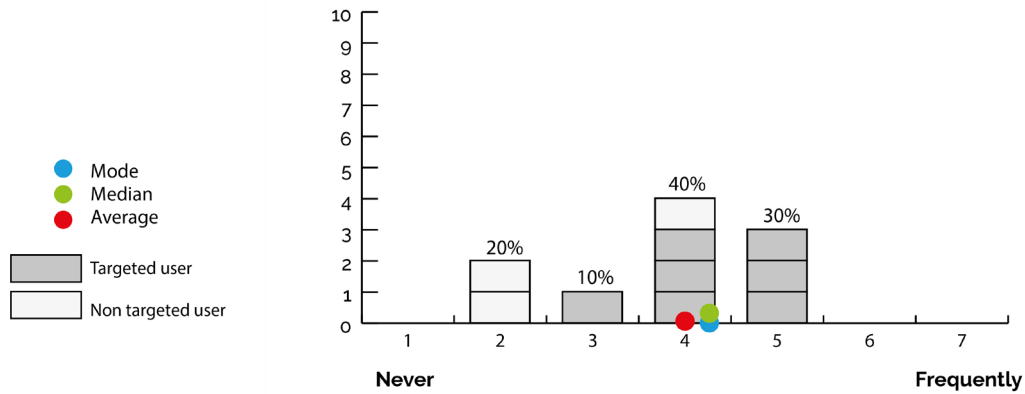
Answers regarding the concept of the platform were also positive (Figure 119). When asked about recognising the advantages of a modular system as the presented one, all designers and artists answered above 6. Only the three users that did not belong to the targeted audience answered 4. Nonetheless, all agreed that they thought the concept and execution of the platform were valuable, giving answers above 5. To conclude the evaluation of the concept, when asked about the frequency with which they would use the platform, users that belong in the target audience answered between 4 and 5, and the rest answered between 2 and 3.

**I considered the concept and execution to be:**

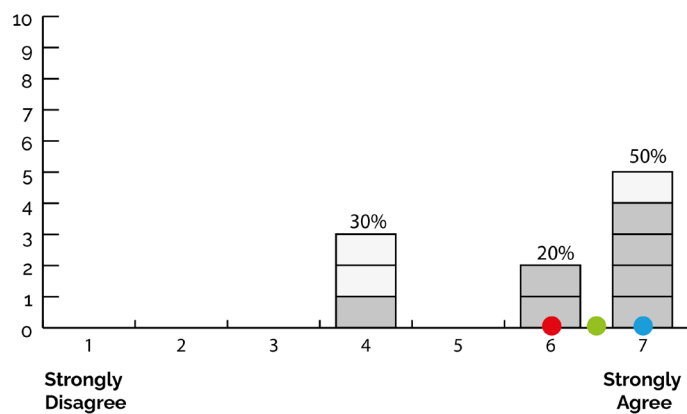


**Figure 119.** Distribution of users' answers regarding the validation of the concept.

### How often would you use this platform?



### I recognise the advantages of using a system like this.



As for the context of the platform, users answered that the project fitted the areas of New Media Art and Design (Figure 120). Users were consistent and certain that it would fit in the area of Design, however, they were less certain about New Media Art. The lowest rating was given by a designer and the second-lowest by a non-targeted audience user.

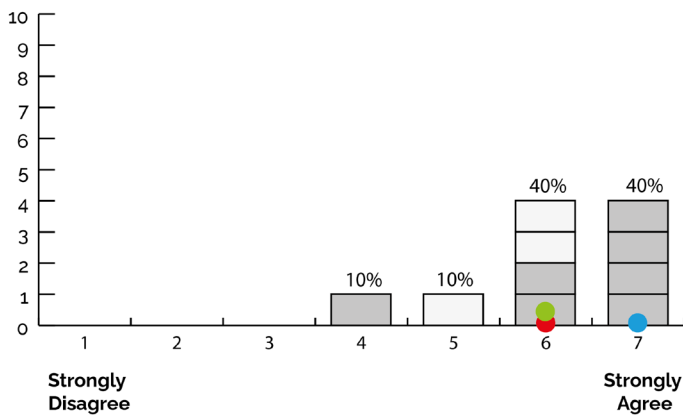
Finally, the users were also questioned on which usages they would give the generated artefacts, to whom they would recommend the platform and how they would describe it.

To the first question, there was a wide variety of answers. Users stated the platform could be used for posters, exhibitions (interactive digital mirrors), decoration, social posts, crochet patterns, book covers, memes, digital art, create typefaces, videos...

However, two answers stood out from this pool. One user, a design student, pointed out they would use the platform to teach Design since

it was a good sketching and prototyping tool to teach basics. The second answer came from a user in the area of Tourism, who pointed out the platform for public use. He described the creation of installations that used tiles that resembled symbols of the city where they were being displayed. They gave an example of the Portuguese pavements.

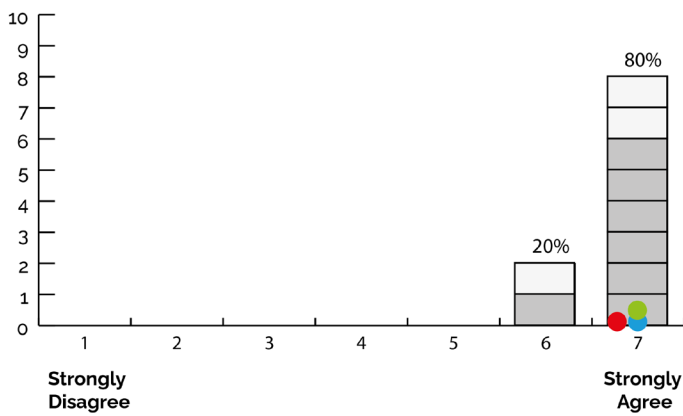
**This tool fits the area of New Media Art.**



**Figure 120.**

Distribution of users' answers regarding the context of the project.

**This tool fits the area of Design.**





## ANALYSIS

Overall, the results were better than expected, most doubts were successfully answered and we understood the urgency with which some features were needed. A critique however needs to be applied to this phase of the project: the number of tests was low and even lacked an earlier phase that could answer some of the earlier doubts about the concept of the project. With this in mind, the obtained results were positive and revealed consistency in the value of the product.

More specifically, we will now analyse the possible reasons for the occurrence of some errors and some special events that were not predicted. We will establish the connection of these to the context of the situation, such as the profile of the users or other conditions.

The first four tests were the ones with the most errors. These were technical and focused on programming mistakes. However, from the first test, it was clear that most critiques were going to be focused on the interface associated with the text features. Due to this, since this instance, we began to program new interactions and a better interface. After applying changes, the last six tests had a lot of improvements. Surprisingly this was also the wave of tests where we tested the users that did not belong to the target audience. Nonetheless, these managed to be some of the fastest times regarding the completion of tasks.

About the frequency to which users would use the platform, it was not surprising to find that it would not be often. Even though users outside of the target audience were the ones to rate the frequency with the lowest values, designers and artists still kept their answers between 4 and 5. This event may be due to the restricted pixelated look in the artefacts, which is not always desired in projects, and also due to the lack of more user freedom, for example, to create and add tiles. With this in mind, the average grade of the frequency with which users would use the platform still managed to be 3.8, and when considering only the target audience users the average is 4.4. This may show that although this model would not be as regularly used as Illustrator or InDesign, it would still serve as a quick production model for some occasions.

Moreover, a lot of positive feedback was collected regarding the interface's usability and aspect, the effectiveness of the platform and the opinion on the produced artefacts. The answers regarding the possible usages also showed a great range of ideas. All of the results evaluate the success of the initially established goals, which were to build a modular platform that could quickly transform different inputs into interesting artefacts for varied purposes.

Another unexpected and positive point was the reaction from the users with no Design or artistic background when they got to experience the live capture feature. They were enthusiastic and kept moving in front of the camera. This, alongside their answers about the artefacts' interest (which were positive), helped to get an idea of the reaction we would get from the receivers referenced in the Audience section. The people who will most

likely encounter the generated artefacts, may not have a design or artistic background but their reaction will have an impact on the success of the platform. If the artefacts are well received, the user that produced them may come back for more.

Regarding the system's modularity, we found that it was hard to establish a limit to its possibilities. Users requested new features that would open a world of possibilities. However, these deviated from the goal, making it difficult to tell when the system would be complete. This is where the idea of having a module enters. A module should be a fixed element: it should have limitations or its definition becomes hard to grasp.

Readaptations can be made from this framework to better suit other purposes. However, it is important to define its boundaries.

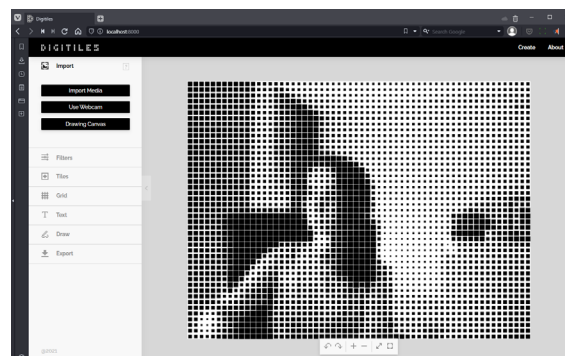
## 4.5 Applications and Dissemination

Besides knowing how to manipulate the system, it is also relevant to think about the situations it could be used in. This section intends to talk about its validation and showcase different usages to which the system could contribute or elevate a project in areas such as Design or New Media Art.

### 4.5.1 Applications

A modular and generative system allows itself to be re-adapted and approached in different ways. Firstly, we will explore the usages that can be given to the engine within the format of a website. Then we will explain other applications that could be reached from simple adaptations of the current model.

The online website is a good way to reach more people and provide them with a good extension of the engine, allowing users to quickly manipulate and explore the system's functionalities (Figure 121).

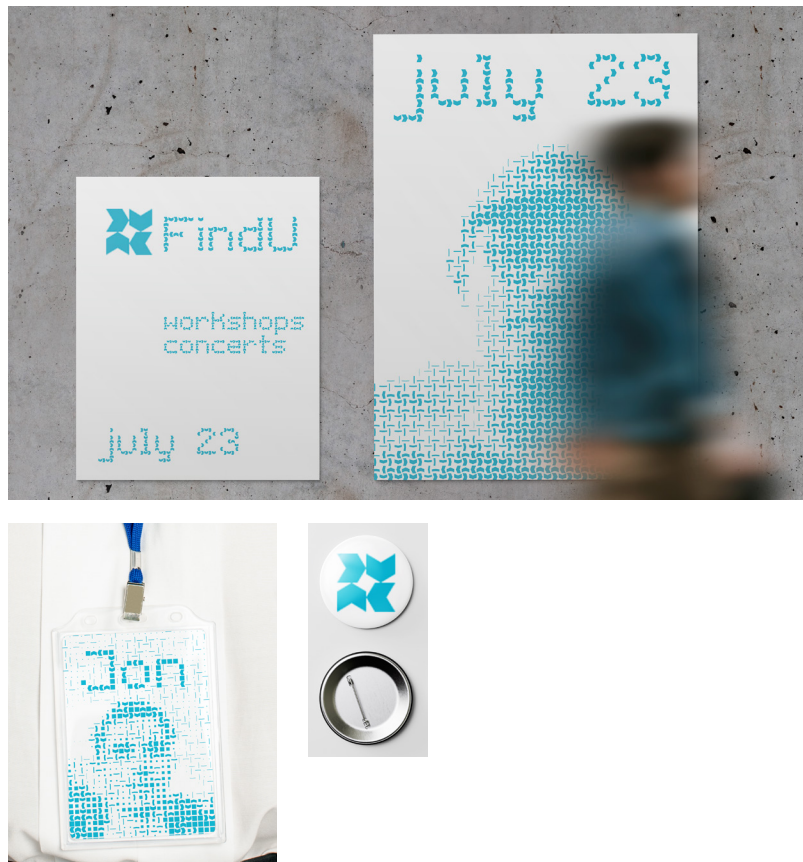


**Figure 121,**  
Website Digitiles capturing image.

Regarding the area of Graphic Design, a website like this can be used to produce artefacts for a visual identity that relies on modularity, therefore achieving visual coherence faster. The tile can be a symbol, a logo, or another reference that helps to transform and link a random artefact to the identity (Figure 122). The results can be used as posters, banners for events or even personalised gifts for the visitors. It can also be used to compose visuals that perfectly fit into a specific display. In a more exploratory approach, it can be used to produce fun visuals, avatars to gifs or even transformed drawings.

Since the website gives the option to export as SVG, the results can also be used in the design of responsive digital applications. The PDF export can also be used to do printing or even laser cuts.

**Figure 122.**  
Mockups for an identity created  
for an imaginary event  
using the system.



As it is now, the system can portray different glyphs and emojis. This also brings a future possibility of adapting the system to better deal with typography, handling type skeletons, and generating modular types that can be exported as actual fonts or even to generate variable fonts.

Apart from this, the system can also enrol in New Media Art. Since captured images and videos are just sequences of frames, and since the system is fast, it enables rendering transformed videos in real-time. This opens space to use the system to create installations or interactive mirrors (Figure 123). By displaying the visuals in a larger display and using, for



**Figure 123.**  
Mockups of a mirror installation  
using Digitiles.

example, a Kinect, we can reprogram the tiles to respond to certain movements, proximity or other types of interaction, generating new visuals that turn into an immersive experience. These visuals may or may not be more abstract, depending on the desired outcome. As an example, instead of just translating the image's brightness values, the system may be programmed to only portray detected movement, using its speed as input for the tiles' weight instead.

Another usage can be imagining the system as an interactive drawing tool, in which the user picks the tile and colour and creates any composition (Figure 124). The weight that the tile takes can be the pressure the user applies when using a tablet or even the speed to which he draws. This idea could also be applied as a sketching tool, used for example for teaching design concepts, as suggested by a user. Since it does not require skill, the students could focus on composing and understanding other basic concepts. Ultimately, we can mix any of the last ideas and create an interactive body drawing experience. The advantage of this modular visual system is that it becomes simpler to deal with different inputs and to generate several results with varied styles and sizes.

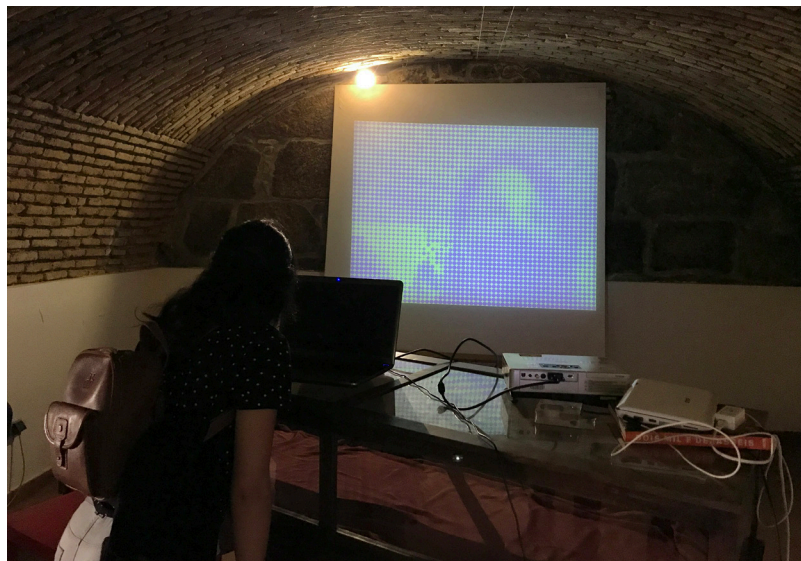


**Figure 124.**  
Mockups of a drawing application  
using Digitiles.

### 4.5.2 Dissemination

Earlier in the development phase, one scientific article was written and submitted to the conference Artech 2021 (2021.artech-international.org). The article was accepted and it will be presented in October of 2021.

Additionally, an opportunity appeared to participate in a set of installations for the Romaria Cultural 2021 in Gouveia. This event consisted of a collection of cultural shows and expositions. This was a great opportunity to exhibit the work as an interactive installation. The process of adapting the existing system was simple and low-cost. The goal was to have a mirror installation in which, from minute to minute, the tiles and colours would randomise. The list of colours matched the palette of the identity of the event. Furthermore, between the randomization, a sentence saying "Romaria Cultural " would appear. The result can be seen in Figure 125.



**Figure 125.**  
Installation of the platform  
Digitiles for Romaria Cultural  
2021 in Gouveia.

5.

**Conclusions**

As designers and artists, one of the most challenging tasks is to keep discovering how to push new systems to different uses. With this in mind, we start to reach for the concept of modularity, with the goal of using the knowledge we have of the available pieces to form new combinations.

From this context, we developed a website that generates variable image patterns using a modular structure. It deals with different inputs and interprets them by simplifying their understanding. In this case, it transforms the images into tiles of varying shape. By accepting a set of limitations, in this case, the use of tiles, the system quickly creates artefacts that can be used for different purposes.

The presented dissertation began by analysing the bibliography about the history of modularity in the context of Image Synthesis, New Media Art, Design, and Typography. The research impacted our perception of visual information and its adaptability to different media outlets. The research uncovered techniques, case studies and ideas that could be added.

After the research phase, the first prototype was built and, from it, we understood what functionalities would be needed to improve the project. In order to expand the flexibility of the tile in its ability to interpret results we added functionalities such as image adjustment, adding text and drawing.

The next phase of the project was the development of the website. It started with the projection of different wireframes and mockups and proceeded to its implementation. Besides the proposed system, several features were added and, in the end, we managed to build a generator of variable mosaics that serves different graphical purposes. Next, we developed an user interface which would allow users to manipulate the system's settings to generate their own outputs.

Finally, we evaluated the concept and its execution by testing it with potential users. We managed to solve different errors as well as improve the interface's usability but most importantly, the system proved to be effective. The results were overall positive but we also received a considerable amount of suggestions that could not be met in time. These did not interfere with the goals set for this dissertation, however, it is interesting to mention these improvements for future work.

First, the user interface would need refinements. This would include (i) setting better error prevention and communication with the user, adding (ii) better interaction with the text on the canvas, (iii) new and more complex tiles, (iv) and more tooltips. In addition, (v) the website also needs to be optimised for more browsers, such as Safari. Other smaller

improvements would include (vi) showing the seconds of a recording, (vii) and being able to set a timer to record. More complex future work would include (viii) giving the user the possibility of creating their tile and creating smooth transitions. Although not pointed out by the users, it would be also useful to (ix) save and export certain configurations that later could be loaded into the system and instantly used in other inputs. This would serve well when dealing with, for example, visual identities, where visual coherence is always required.

To conclude, the presented website sits at a current version and can be used for Design purposes such as to achieve visual coherence in the generation of visuals with the same pattern. Since it offers many types of export formats, they can be applied to different scenarios such as print, display, laser cuts, social media. The system also can enroll in New Media Art since it can capture the users' image and give back a unique interpretation of it. In this sense, and even though future work is in sight, the project met the proposed goals for this dissertation. We managed to successfully generate variable image patterns and we also added more layers to them such as text and drawings that would be presented in the same visual world through tiles.





# References

Accept & Proceed: Grace FO display. (2020, March 1). Creative Review. <https://www.creativereview.co.uk/grace-fo-display/>

Albright-Knox. (2018). *Computer Nude (Studies in Perception I)*. <https://www.albrightknox.org/artworks/p20142-computer-nude-studies-perception-i>

ALFAZETA. (n.d.). *Advantages of electromagnetic flip-disc technology*. Retrieved 24 January 2021, from <https://flipdots.com/en/advantages-of-electromagnetic-flip-disc-technology/>

Benson, R. (n.d.). *Digital halftones and stochastic printing | The Printed Picture—A survey of the process*. Retrieved 23 January 2021, from <https://benson.readandnote.com/videos/digital-halftones-and-stochastic-printing>

Bil'ak, P. (2004, November). *Typotheque: Karel Martens*, graphic designer by Peter Bilak. [https://www.typotheque.com/articles/karel\\_martens\\_graphic\\_designer](https://www.typotheque.com/articles/karel_martens_graphic_designer)

Bolter, J. D., & Gromala, D. (2003). *Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency*. MIT Press.

BREAKFAST. (n.d.). *Flip-Discs Canvas*. BREAKFAST. Retrieved 24 January 2021, from <https://breakfastny.com/flip-discs-canvas>

Cartwright, M. (2019, September 2). *Mosaic*. Ancient History Encyclopedia. <https://www.ancient.eu/Mosaic/>

Cassel, D. (2018, March 11). *The Surprisingly Rich History of ASCII Art – The New Stack*. <https://thenewstack.io/surprisingly-rich-history-ascii-art/>

Christou, C. N. (2008). *Optimal Dither and Noise Shaping in Image Processing*. Universidade de Waterloo.

Chu, H.-K., Chang, C.-S., Lee, R.-R., & Mitra, N. J. (2013). *Halftone*

QR codes. *ACM Transactions on Graphics*, 32(6), 217:1-217:8. <https://doi.org/10.1145/2508363.2508408>

Dickinson, D. (1972). *A handbook of intaglio techniques*.

*Digital Mona Lisa | Database of Digital Art*. (n.d.). Retrieved 19 January 2021, from <http://dada.compart-bremen.de/item/artwork/651>

Dunbabin, K., & Dunbabin, K. M. D. (1999). *Mosaics of the Greek and Roman World*. Cambridge University Press.

Ernest, P. (2003). *Mathematics Education and Philosophy: An International Perspective*. Routledge.

Finkelstein, A., & Range, M. (1998). *Image Mosaics*. Proceedings of the 7th International Conference on Electronic Publishing, Held Jointly with the 4th International Conference on Raster Imaging and Digital Typography: Electronic Publishing, Artistic Imaging, and Digital Typography, 11–22.

*Flip-disc display inventors*. (n.d.). Edubilla.Com. Retrieved 21 January 2021, from <http://www.edubilla.com/invention/flip-disc-display/>

Furht, B. (Ed.). (2008). *Dithering*. In *Encyclopedia of Multimedia* (pp. 198–199). Springer US. [https://doi.org/10.1007/978-0-387-78414-4\\_18](https://doi.org/10.1007/978-0-387-78414-4_18)

Gerstner, K. (1964). *Designing Programmes: Programme As Typeface, Typography, Picture, Method*. Lars Muller Publishers.

Gilchrist, L. Alan (1994). *Lightness, Brightness and Transparency*. Psychology Press,

Gooch, B., & Gooch, A. (2001). *Non-Photorealistic Rendering*. <https://doi.org/10.1201/9781439864173>

Gosling, E. (2015, April 27). *Muir McNeil's typography-led LCC Summer Shows identity*. <https://www.itsnicethat.com/articles/muir-mcneil-lcc>

Hausner, A. (2002). *Simulating Decorative Mosaics*. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, 1. <https://doi.org/10.1145/383259.383327>

highlike.org. (n.d.). ARAM BARTHOLL. Highlike. Retrieved 24 January 2021, from <https://highlike.org/text/aram-bartholl/>

Hybe. (n.d.). Retrieved 24 January 2021, from <http://www.kinetica->

[museum.org/artists/members/hybe.html](http://museum.org/artists/members/hybe.html)

Jansen, S. (1997). *STAR WARS ASCIIMATION* - Main Page. <http://www.asciimation.co.nz/>

Knowlton, K. (n.d.-a). Ken Knowlton. Retrieved 24 January 2021, from <https://www.kenknowlton.com/pages/03shortbio.htm>

Knowlton, K. (n.d.-b). Knowlton Mosaics—Portraits by Computer Assisted Art Pioneer Ken Knowlton. Retrieved 24 January 2021, from <https://www.knowltonmosaics.com/>

Kyprianidis, J. E., Collomosse, J., Wang, T., & Isenberg, T. (2013). *State of the “Art”: A Taxonomy of Artistic Stylization Techniques for Images and Video*. *IEEE Transactions on Visualization and Computer Graphics*, 19(5), 21.

Lau, D. L., & Arce, G. R. (2018). *Modern Digital Halftoning*. CRC Press.

Lava. (2012). *Moscow Design Museum—Russia’s first design museum*. [http://lava.nl/projects/moscow\\_design\\_museum](http://lava.nl/projects/moscow_design_museum)

Lechêne, R. (n.d.). *Printing | History, Techniques, & Facts*. Encyclopedia Britannica. Retrieved 23 January 2021, from <https://www.britannica.com/topic/printing-publishing>

Liberatore, S. (2016, November 29). *Watch the amazing ‘flip-disc’ display in action*. Mail Online. <http://www.dailymail.co.uk/-/article-3983466/index.html>

Lipshitz, S., Wannamaker, R., & Vanderkooy, J. (1992). *Quantization and Dither: A Theoretical Survey*. *Journal of the Audio Engineering Society*, 40, 355–374.

Lupton, E., & Phillips, J. C. (2008). *Graphic Design: The New Basics*. Princeton Architectural Press.

Mackenzie, C. E. (1980). *Coded Characters Sets, History and Development*. Addison-Wesley Publishing Company. [https://textfiles.meulie.net/bitsaved/Books/Mackenzie\\_CodedCharSets.pdf](https://textfiles.meulie.net/bitsaved/Books/Mackenzie_CodedCharSets.pdf)

Madrigal, A. C. (2014, January 30). *The Lost Ancestors of ASCII Art*. *The Atlantic*. <https://www.theatlantic.com/technology/archive/2014/01/the-lost-ancestors-of-ascii-art/283445/>

Martens and Martens. (n.d.). Martens-martens. Retrieved 22 January 2021, from <https://martens-martens.com/dutchcloud.html>

Meggs, P. B., & Purvis, A. W. (2016). *Meggs' History of Graphic Design*. John Wiley & Sons.

Muir McNeil. (n.d.). *TwoPoint Poster « MuirMcNeil*. Retrieved 24 January 2021, from <https://muirmcneil.com/project/twopoint/>

Nelson, J. (1939). *Artyping. Artyping Bureau*. <https://loriemersondotnet.files.wordpress.com/2014/01/artyping.pdf>

Ong, J. (2020, September 2). "Absence in design is very important": Karel Martens on paying attention to the things we don't see. <https://www.itsnicethat.com/features/karel-martens-re-printed-matter-graphic-design-020920>

Ostromoukhov, V. (1999). *Artistic halftoning: Between technology and art* (R. Eschbach & G. G. Marcu, Eds.; pp. 489–509). <https://doi.org/10.1117/12.373433>

Patros, A. (n.d.). *Digital Mona Lisa*. DIGITALMONALISA.COM. Retrieved 20 January 2021, from <https://www.digitalmonalisa.com/>

Pellacini, F., & Kim, J. (2002). *Jigsaw Image Mosaics*. 8.

Petrulis, A. (2006). *MetroPostcard Guide to Postcard Printing Techniques* 3. <http://www.metropostcard.com/tech3-half-tone.html>

Pil, F., & Cohen, S. (2006). *Modularity: Implications for Imitation, Innovation, and Sustained Advantage*. *Academy of Management Review*, 31, 995–1011. <https://doi.org/10.5465/AMR.2006.22528166>

Piller. (n.d.). *Escher & Treasures from the Islam. Escher in Het Paleis*. Retrieved 6 September 2021, from <https://www.escherinhetspaleis.nl/story-of-escher/escher-treasures-from-the-islam/?lang=en>

R2. (2010). *AGI Open Congress Porto 2010 Poster*. <https://www.r2design.pt/projects/agi-open-congress-porto-2010-poster/>

Ridell, A. (1975). *Typewriter Art*. London Magazine Editions.

Rozin, D. (n.d.). *Daniel Rozin Interactive Art*. Retrieved 18 January 2021, from <https://www.smoothware.com/danny/index.html>

Rozin, D. (1999). *Wooden Mirror*. <https://www.smoothware.com/>

danny/woodenmirror.html

Rozin, D. (2005). Daniel Rozin Circles Mirror. <http://www.smoothware.com/danny/circlesmirror.html>

Schofield, S. (1994). *Non-Photorealistic Rendering: a critical examination and proposed system*.

Silanteva, D. (n.d.). *Typographic music dina silanteva*. Retrieved 20 January 2021, from <http://www.ddina.com/index.php?/2011/typographic-music/2/>

Smith, T. (2013). *Star Trek: The original computer game*. [https://www.theregister.com/2013/05/03/antique\\_code\\_show\\_star\\_trek/](https://www.theregister.com/2013/05/03/antique_code_show_star_trek/)

Sullivan, M. (2011, November 7). *Dina Silanteva—Typographic Music*. <http://www.typpetoken.net/publication/dina-silanteva-%E2%80%94-typographic-music/>

Thinks Create. (2018, September). *Interactive Flip-Dot Display*. <https://thinkcreate.us/portfolio-interactive-flip-dot-display.html>

Tullett, B. (2014). *Typewriter Art: A Modern Anthology*. Laurence King Publishing.

Ulichney, R. (1999). *A Review of Halfioning Techniques*. 14.

Visnjic, F. (2017, September 5). *Gysin&Vanetti – O is not a letter, it's a circle*. CreativeApplications.Net. <https://www.creativeapplications.net/processing/gysin-vanetti-o-is-not-a-letter-its-a-circle/>

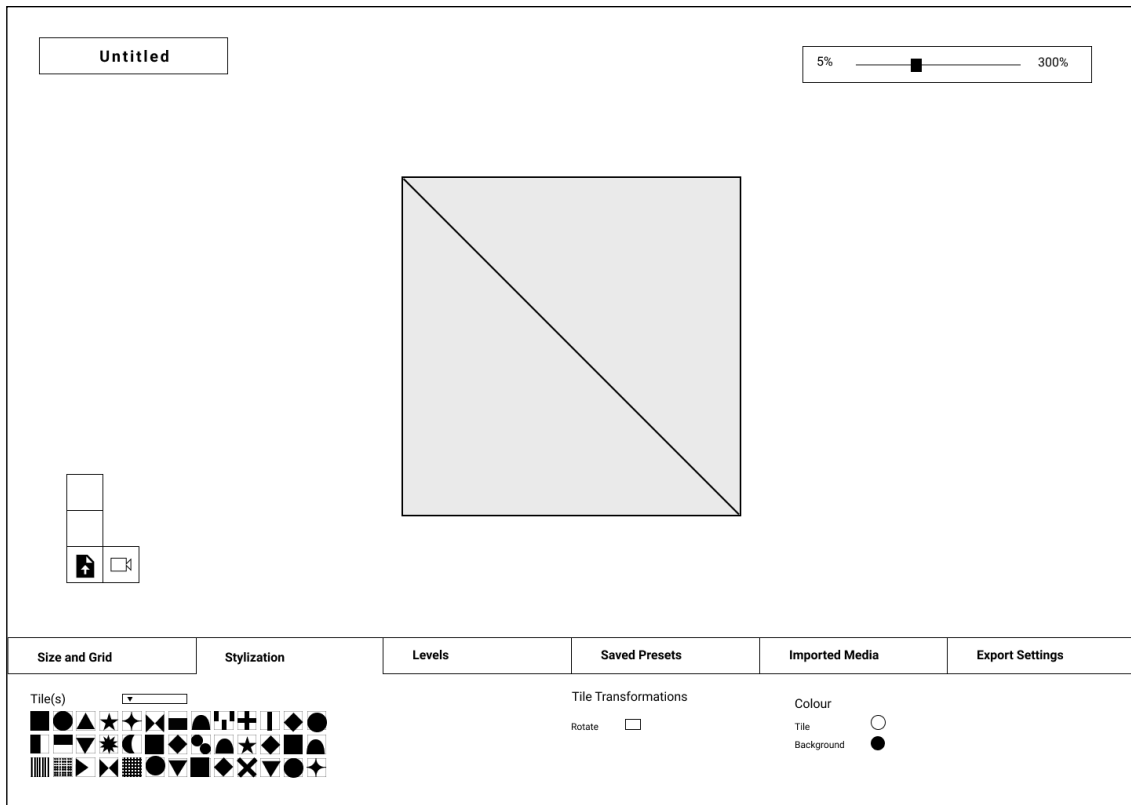
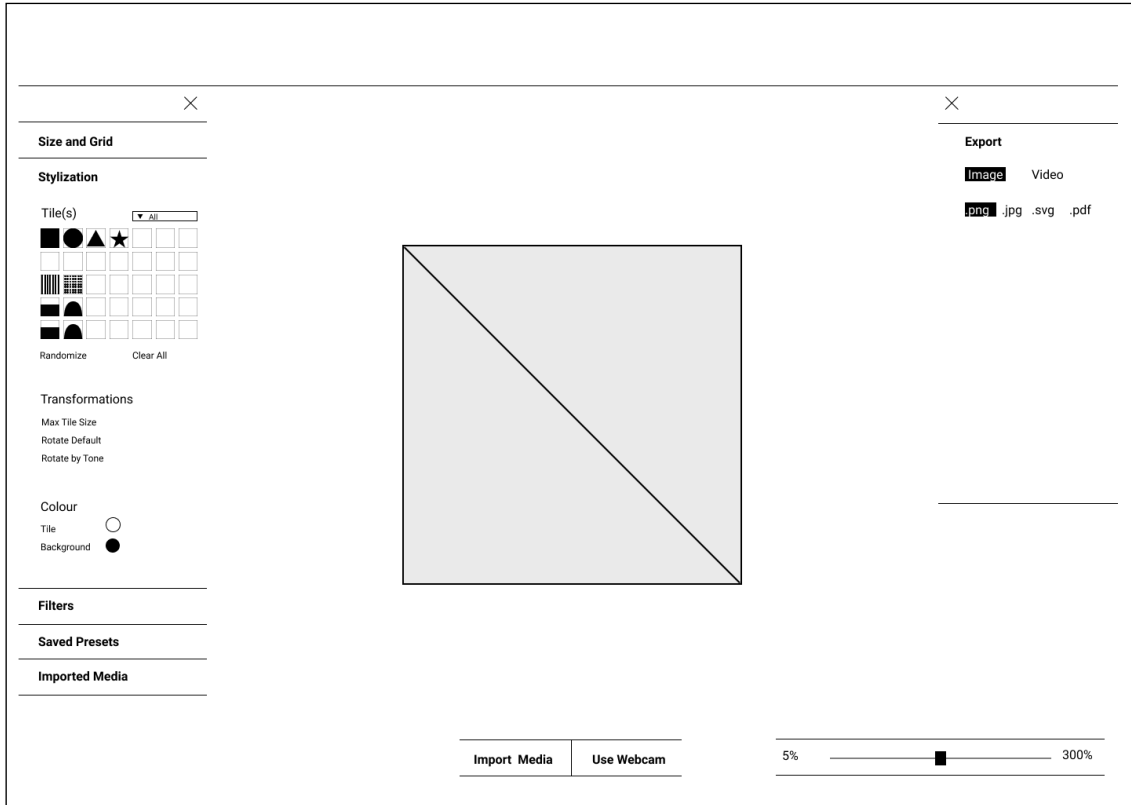
Wilson, M., Wilson, M., & Wilson, M. (2012, September 26). *This Giant Interactive Mirror Turns Viewers Into Pixels*. Fast Company. <https://www.fastcompany.com/1670870/this-giant-interactive-mirror-turns-viewers-into-pixels>

# Appendices



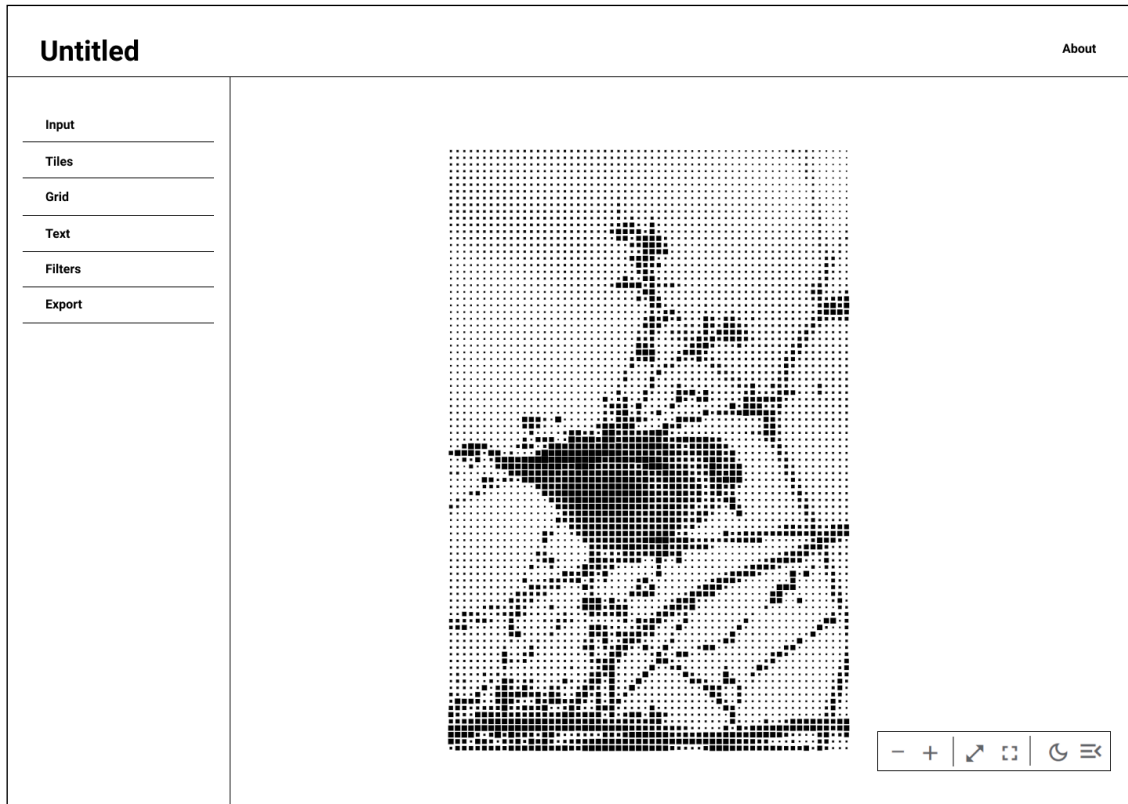
# Appendix A

This appendix contains some of the mockups developed for the user interface by chronological order. It serves as an extension of the Interface section.



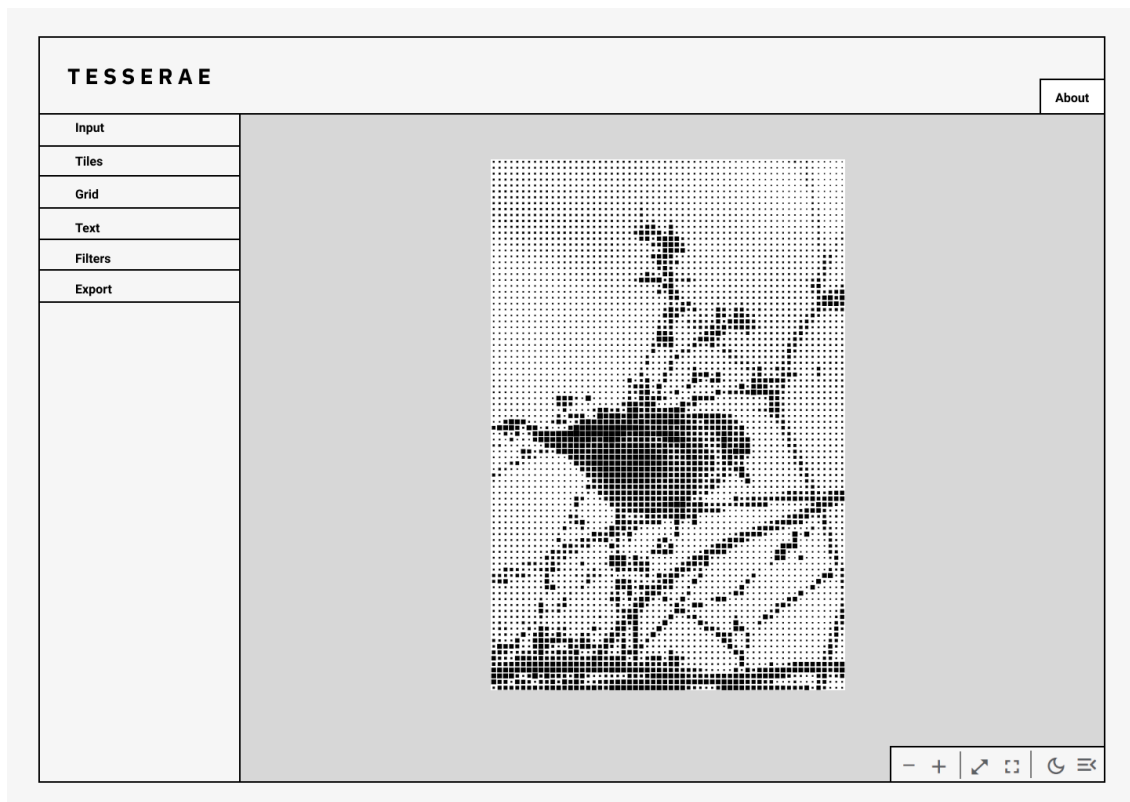
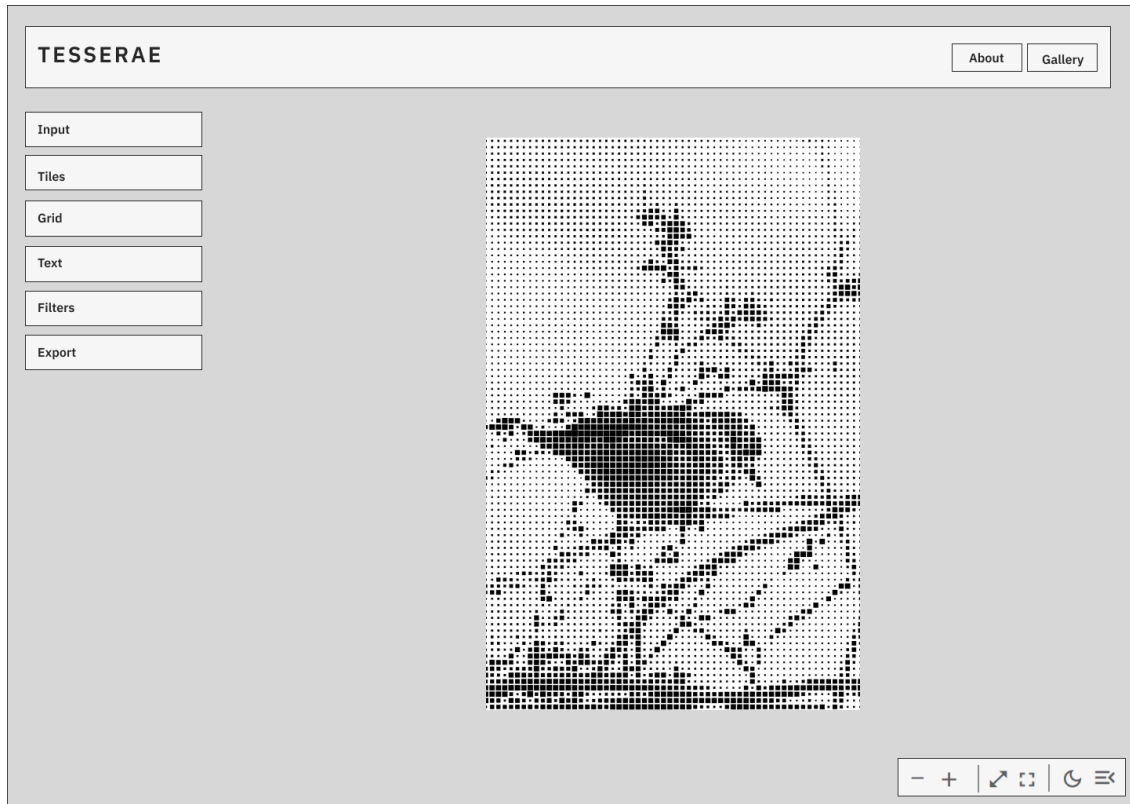
**Wireframes.**

First wireframes created for the project that helped visualizing the layout for the tile settings.

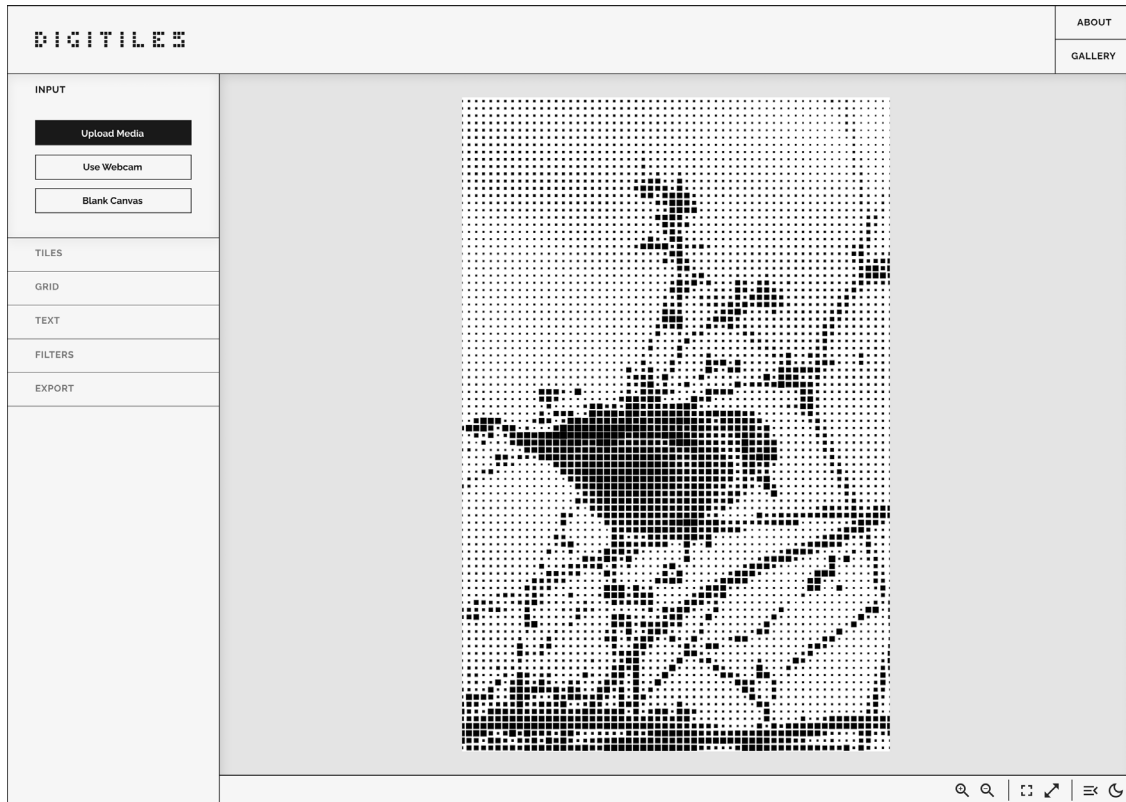


**Mockups.**

First mockup for the design of the website: full view (top) and text panel (bottom).

**Mockups.**

More studies on the design of the platform



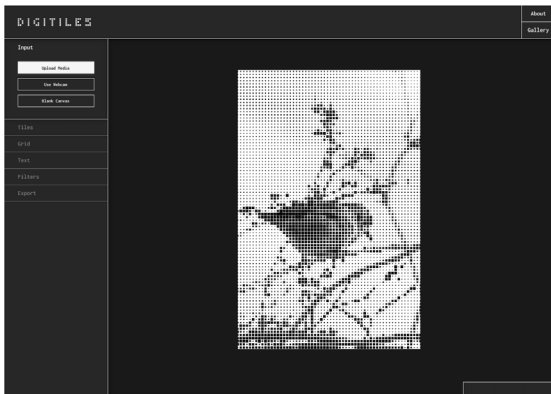
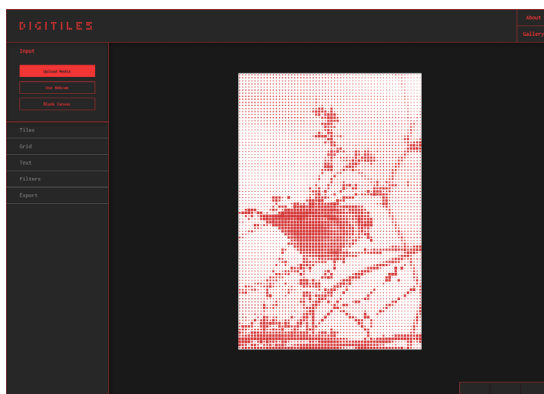
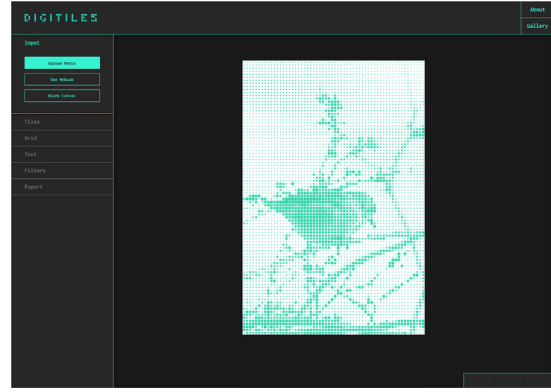
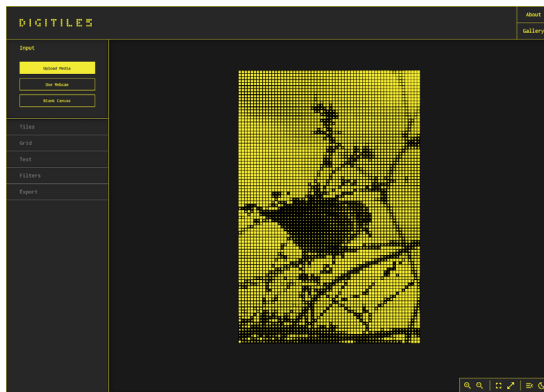
**Mockups.**

More studies on the design of the platform



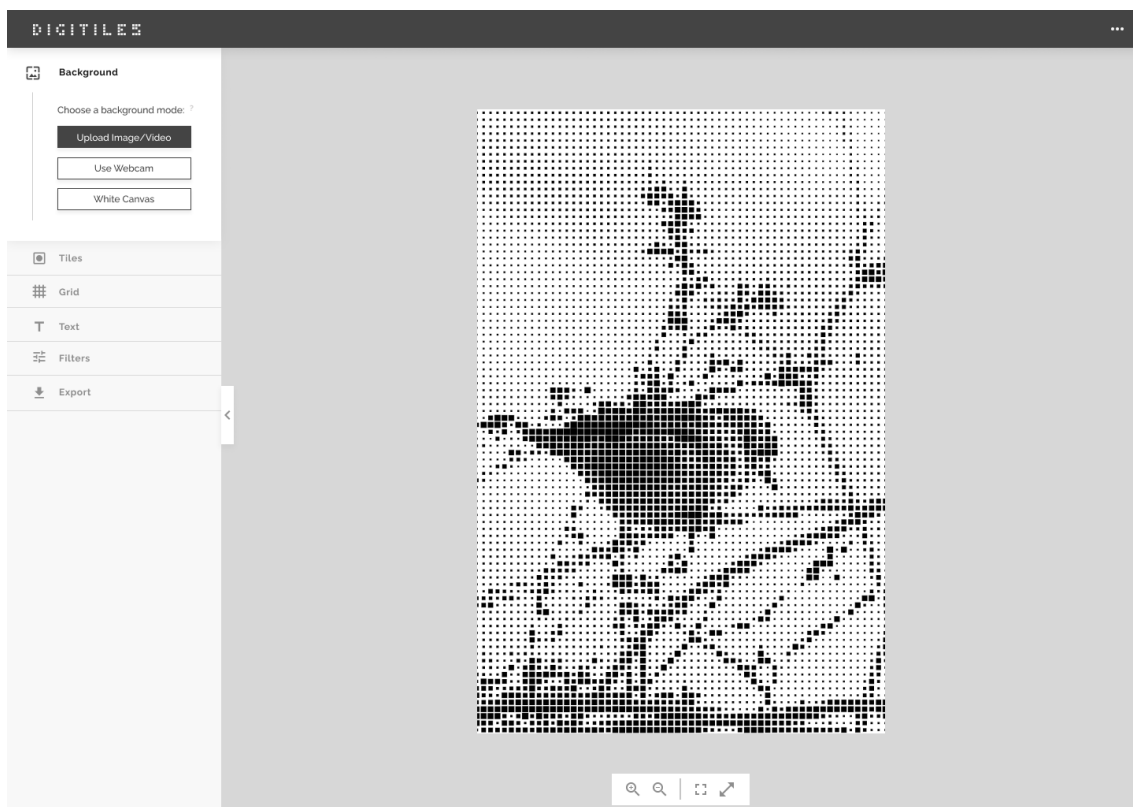
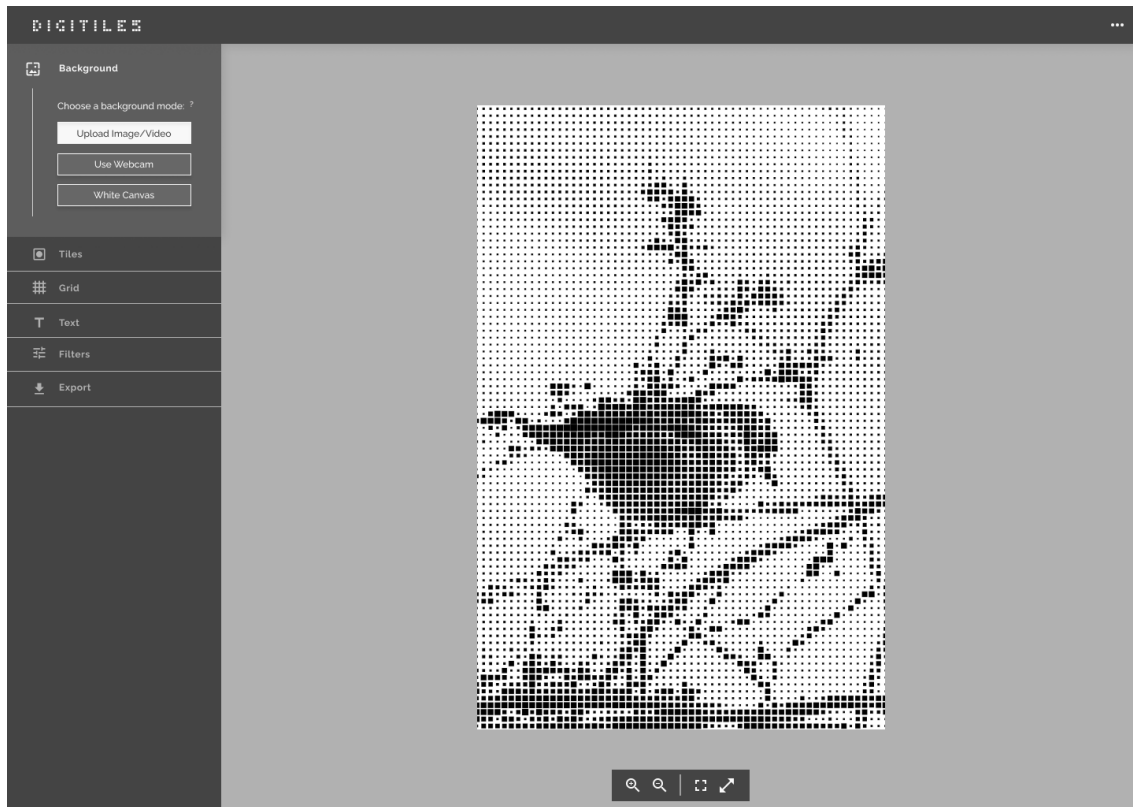
**Mockups.**

More studies on the design of the platform: experiment with shadows and high contrast.



**Mockups.**

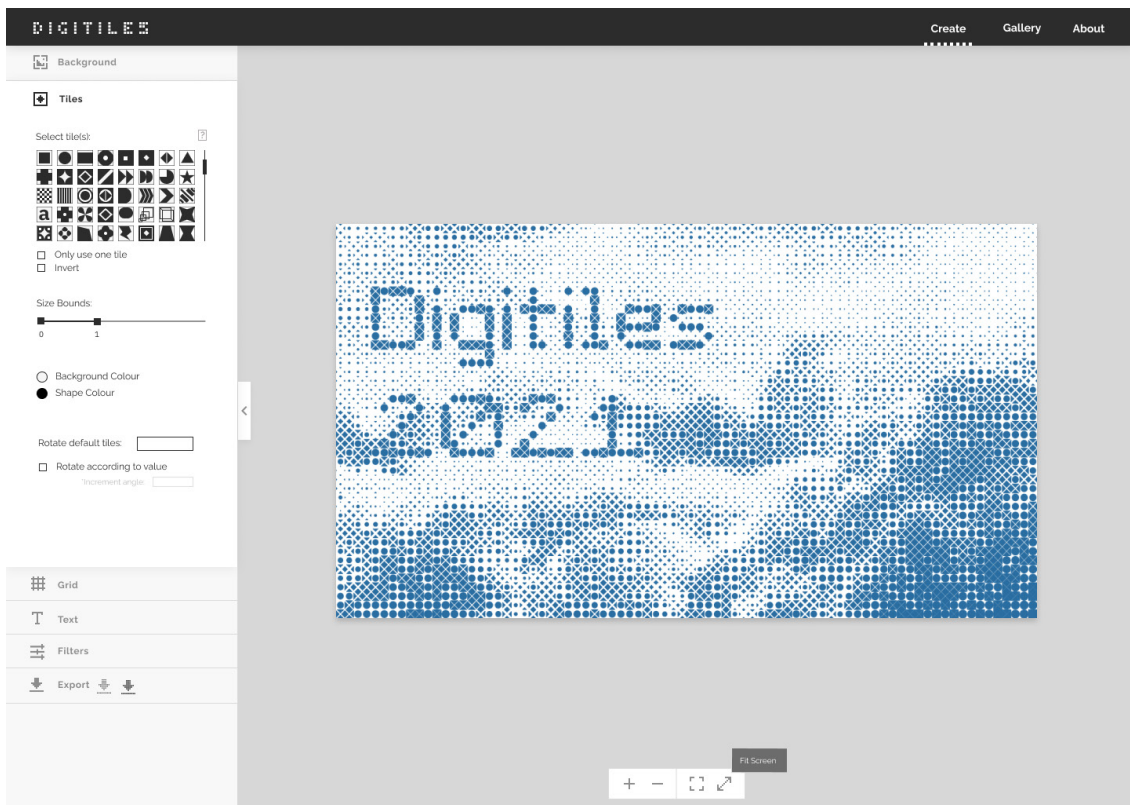
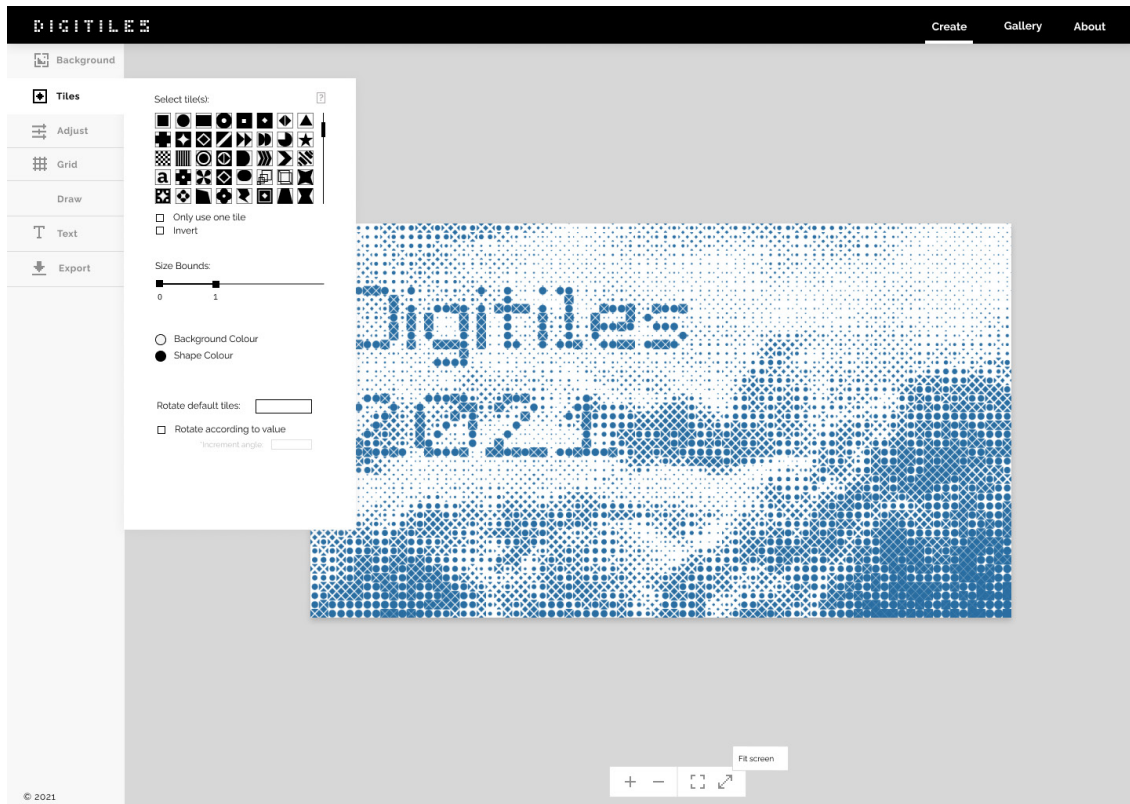
More studies on the design of the platform: experiment with coloured highlights.



### Mockups.

Last studies for the design of the platform: experiments using more neutral colours to centre the focus to the artboard.



**Mockups.**

Last studies for the design of the platform: different layout experiment (top) and final version (bottom).



## Appendix B

This appendix contains the results of the user tests made for the website Digitiles. Inclusive, it contains the set of tasks that were prepared for the user journey, errors found, and the individual answers from all users.

Task ID	Task Description
1	Access the platform through the website.
2	Upload an image from the computer.
3	Click in Filters panel and increase Brightness.
4	Zoom In and then Fit Screen to the available space.
5	Click in Tiles panel and choose one or more tiles.
6	Change the colour of the tile to blue.
7	Click in the Grid panel and change to 80 columns.
8	Increase the size of the tile in the Tiles panel.
9	Export the result as PNG.
10	Go back to Import panel and choose Drawing Canvas.
11	Click in Grid panel and adjust to 80 columns.
12	Click in Text panel and add two new text boxes.
13	Change the font of the first text box to Truchet Serif and the position to x:5 and y:5.
14	Change the font of the second text box to Truchet Sans Serif and the position to x:5, y:25
15	Edit the text to a random word.
16	Click in the Draw panel and choose the pencil.
17	Make a drawing.
18	Go back to the Import panel and choose Use Webcam.
19	Enter FullScreen and leave.
20	Explore the rest of the platform.

**Table.**

Task list developed for the user tests to perform on the website during the test.

Error ID	Description	Type of issue	Priority	Solution
1	Colour input updating slows down the program.	Technical	5	Colour input library
2	Dragging to change text input not working. One user said he's used to hover and scroll instead of click scroll"	Technical	2	Technical issue
3	Tile Size slider needs number input so the user can understand the proportions	Usability	3	Change the interface to fit a number input
4	Slows down on Brave or with an unknown extension	Unknown	1	-
5	Couldn't understand right away that they had to click the text box they wanted to edit	Usability	4	Allow the user to interact with them in the canvas
6	Frustration trying to understand which text box they were editing	Usability	5	Highligh the box that is selected
7	Users can not find where to write in the text box	Usability	3	Giving more contrast to the input so the user notices it
8	Not updating the proportion when using the scroll to change the input	Technical	2	-
9	If moving the pencil too fast, it doesn't draw the path	Technical	3	Implement path calculator
10	Sliders are backwards	Technical	4	Invert values

**Table.**

List of errors found during the user tests.

User ID	Age Group	Area of Work	Level of experience with image editing tools	Browser used
1	18-30	Design & Multimedia	5	Chromium
2	18-30	Design & Multimedia	4	Chromium
3	18-30	Design & Multimedia	5	Firefox
4	18-30	Design & Multimedia	5	Chrome
5	18-30	Informática/ Software	3	Chrome
6	18-30	Design & Multimedia	4	Chrome
7	18-30	Tourism	3	Microsoft Edge
8	18-30	Health	4	Chrome
9	18-30	Design & Multimedia	4	Chrome
10	18-30	Design & Multimedia	4	Chrome

**Table.**

Individual answers to the user questionnaire- Part1.

User ID	The instructions were clear (0-7)	Navigation was conductive	Navigation was easy to learn	The interface looked attractive	Concept and execution of the platform was valuable
1	5	7	7	7	7
2	5	6	5	6	6
3	6	7	7	7	7
4	7	6	6	7	5
5	6	6	7	7	6
6	7	7	6	7	7
7	7	7	7	7	7
8	6	7	7	6	6
9	7	7	7	6	7
10	7	6	7	7	6
Average	6.3	6.6	6.6	6.7	6.4
Mode	7	7	7	7	7
Median	6,5	7	7	7	6,5

**Table.**  
Individual answers to the user questionnaire- Part 2.

User ID	Creating an artefact was fast	The resultant artefacts looked interesting	During the test, the website ran fast	I recognize the advantages of a website as this	I understood all of the features
<b>1</b>	5	7	2	7	6
<b>2</b>	7	6	3	7	7
<b>3</b>	7	7	7	7	6
<b>4</b>	6	4	3	4	7
<b>5</b>	7	6	7	7	7
<b>6</b>	7	7	6	7	7
<b>7</b>	7	7	7	4	7
<b>8</b>	7	6	7	4	7
<b>9</b>	6	7	6	6	7
<b>10</b>	7	7	6	6	7
<b>Average</b>	6.6	6.4	5.4	5.9	6.8
<b>Mode</b>	7	7	7	7	7
<b>Median</b>	7	7	6	6,5	7

**Table.**

Individual answers to the user questionnaire- Part 3.



User ID	Functionalities that were not understood	This system facilitates the creation of a new artefact	How often would you use this system	This system fits the context of New Media Art	This system fits the context of Design
1	Edit the text inside text boxes	7	5	7	7
2	-	7	4	6	7
3	Blank canvas	7	2	7	7
4	-	5	3	4	6
5	-	7	4	5	7
6	-	7	5	7	7
7	-	7	4	6	6
8	-	7	2	6	7
9	-	6	5	6	7
10	-	7	4	7	7
Average	-	6.7	3.8	6.1	6.8
Mode	7	4	7	7	7
Median	7	4	6	7	7

**Table.**

Individual answers to the user questionnaire- Part 4.

User ID	What uses would you think this platform could have?
1	Exposição (moldura/espelho digital), Cartazes (design gráfico em geral), padrões para crochet, capas de livro
2	Cada pessoa usará seus artefactos para finalidades diferentes: Consigo imaginar aplicações simples como decoração e posts em mídias sociais, mas também aplicações maiores como criação de painéis para exposições ou outros usos similares.
3	Criação e influenciar linguagens visuais temporárias, na criação de identidades visuais
4	Para gerar grafismos e imagens para cartazes, sites, etc (especialmente se não tiver experiência em design)
5	Prototype Design
6	Criação de artefactos de design como cartazes, arte digital, etc... Para a área da tipografia, seria muito interessante a experimentação de várias fontes tipográficas para a criação de novos conteúdos, até em formatos como vídeo ou gif.
7	Desconheço.
8	Memes
9	Artefactos relacionados com Design Gráfico e de Comunicação, Instalações em espaços públicos com a utilização da webcam, criando um módulo semelhantes ao das calçadas de cada grande cidade, ex: Lisboa, Viseu, Braga, etc.
10	produção de imagens com base modular (fotografias, logotipos,...), edição de imagens experimentais e/ou abstratas

**Table.**

Individual answers to the user questionnaire- Part 5.

User ID	Who would you recommend this platform to and how would you describe it to them?
1	Pessoas no ramo de design gráfico. É um óptima forma de criar padrões e criar representações estilizadas de imagens.
2	Para mim, a plataforma remete a algo "retro," ao que poderia ser particularmente interessante para quem goste da tecnologia dos anos 80 ao início dos anos 2000 e das linguagens de design que cercavam o meio digital da altura.
3	sim, recomendaria. Boa ferramenta para exploração visual e oportunidade de resultados interessantes.
4	A quem queira criar imagens mais costumizadas e que não saiba usar ferramentas de edição de imagem. Descreveria como uma app de fácil uso que permite gerar imagens interessantes e que permite uma customização avançada (util para encaixar nas identidades visuais de uma entidade ou evento)"
5	Designers/Students in training that require a tool that allows them to create a fast prototype for a client/project.
6	"Designers gráficos, artistas, programadores Uma forma de criar arte digital simples, interessante e poderosa"
7	Pessoas da área de Design e Artes
8	A memers alternativos. Site onde transformas a tua fotografia em mini quadrados com diferentes formas
9	Principalmente a amigos e colegas na área da Arte e do Design, que possam ter interesse em ter uma ferramenta como esta para criar infinitas versões de artefactos gráficos num curto espaço de tempo e com uma grande facilidade.
10	Pessoas interessadas na interseção de design e tecnologia, média, edição de imagens descrição: plataforma para edição de conteúdos visuais com base em módulos e processos generativos"

**Table.**

Individual answers to the user questionnaire- Part 6.

User ID	Last suggestions
1	Muito bom!
2	-
3	Acrescentar a funcionalidade de arrastar e criar objectos no canvas
4	"Undo nao parecia funcionar; Erase apaga tambem o texto; Snapshot da camara para nao estar sempre a processar;"
5	Allow for the user to have more control and freedom over their own project (ex. Create their own Tiles)
6	Conseguir inserir todas as funcionalidades que faltam, colocar on-line e pedir a várias pessoas que criem plugins para novas funcionalidades que aumentem ainda mais o potencial desta ferramenta.
7	Intrigante.
8	Módulos em forma de emojis/fotos (exemplo)
9	-
10	algum foco para a barra de zoom e tamanho de ecrã, possivelmente dar possibilidade de não bloquear proporção ao exportar

**Table.**

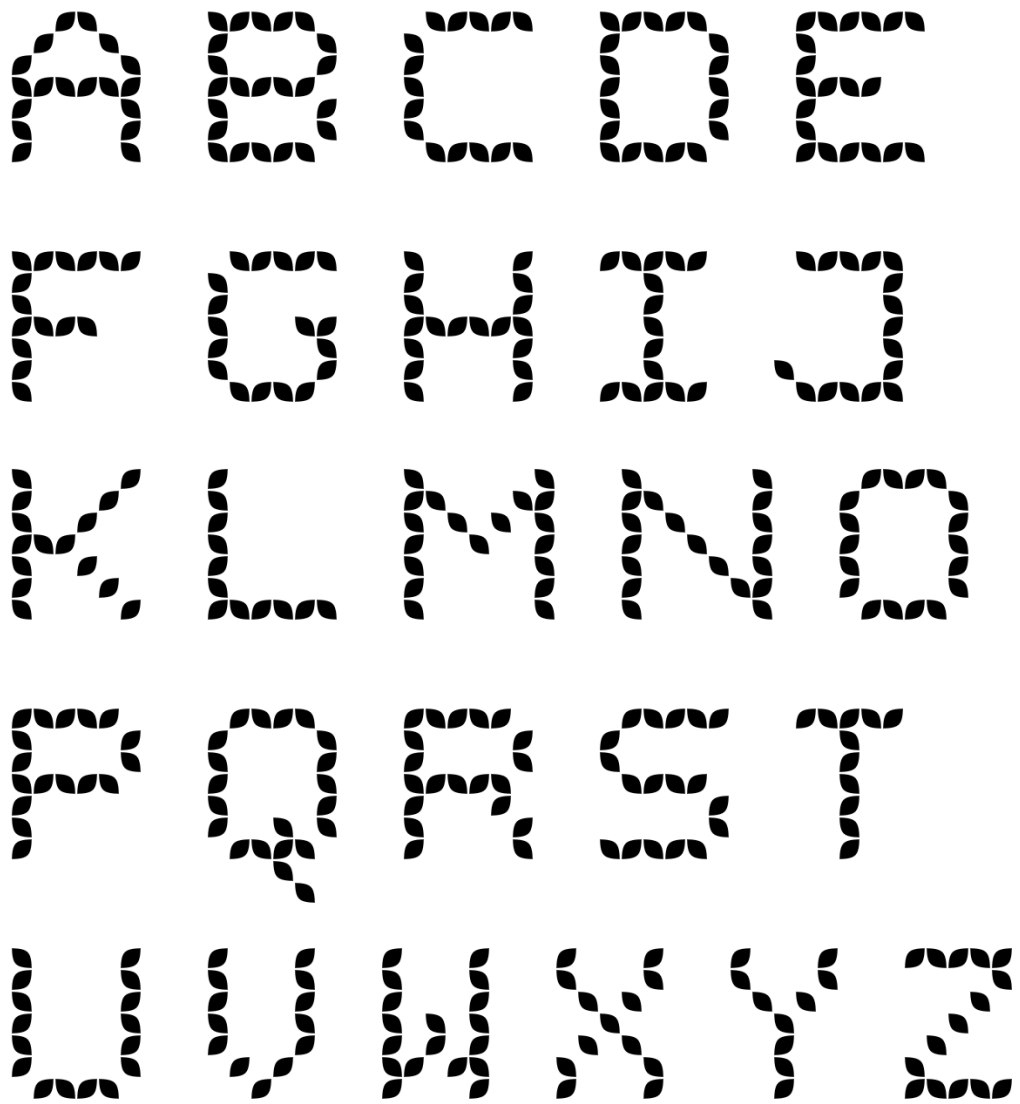
Individual answers to the user questionnaire- Part 7.





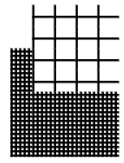
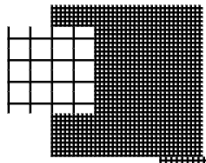
## Appendix C

This appendix contains images of generated outputs as an extension of the works shown in the Outputs section. Different settings and tiles were used to showcase the variability achieved through this system.





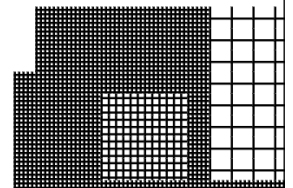
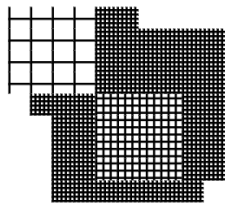
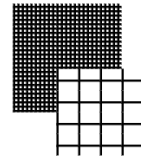
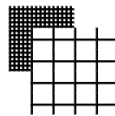
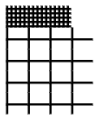
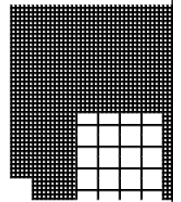
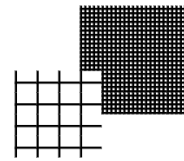


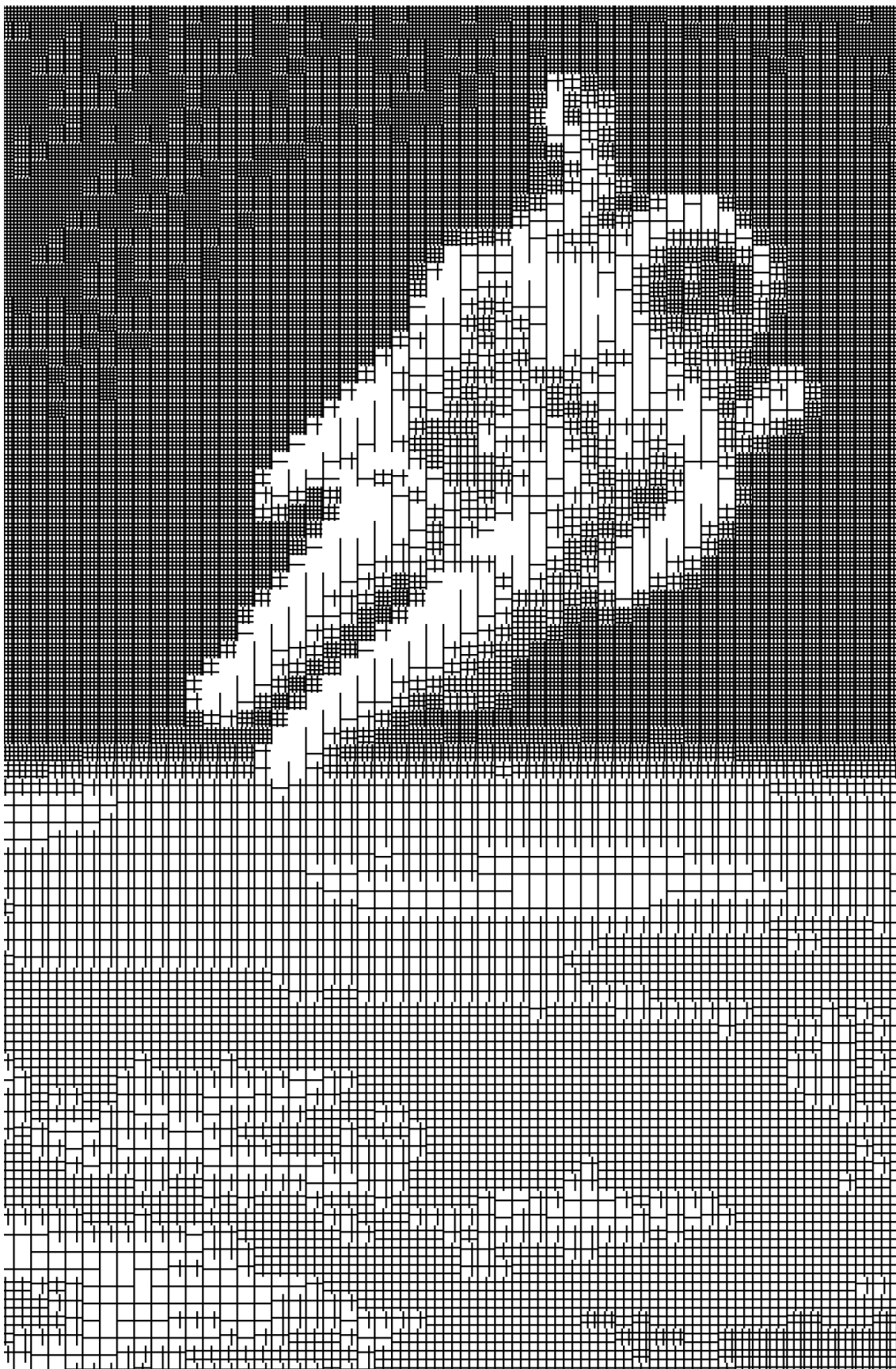


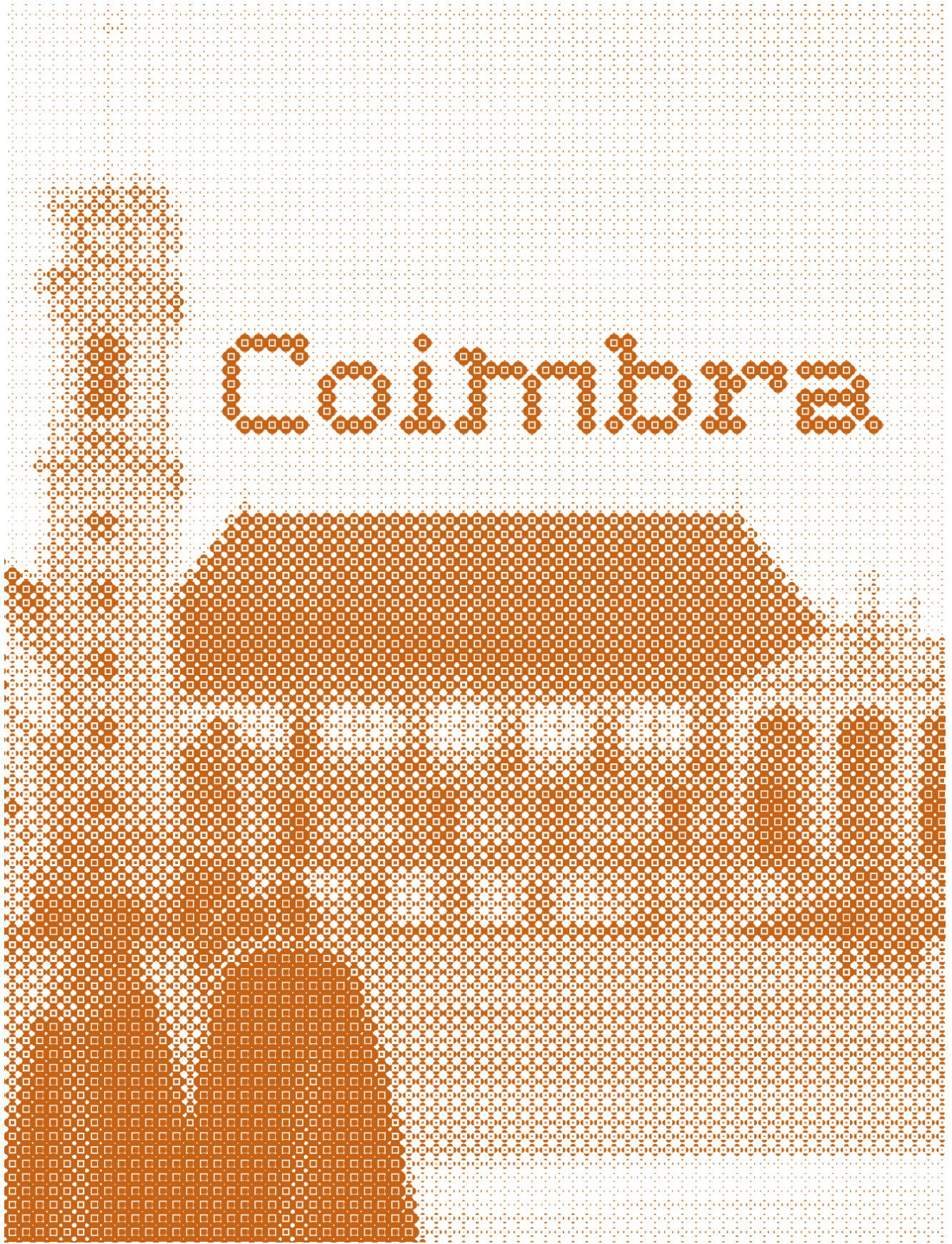
ARTHUR,

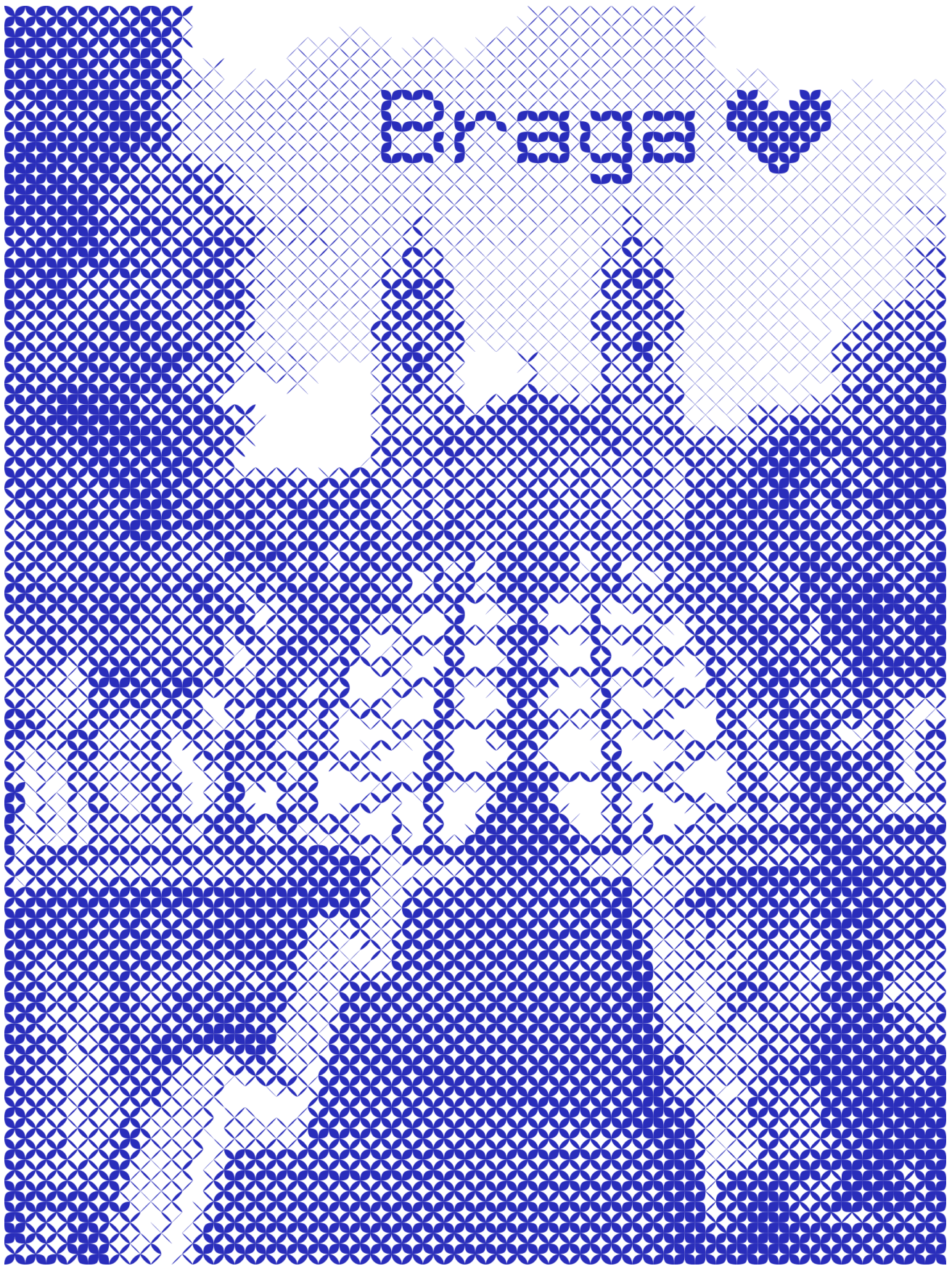
DONT

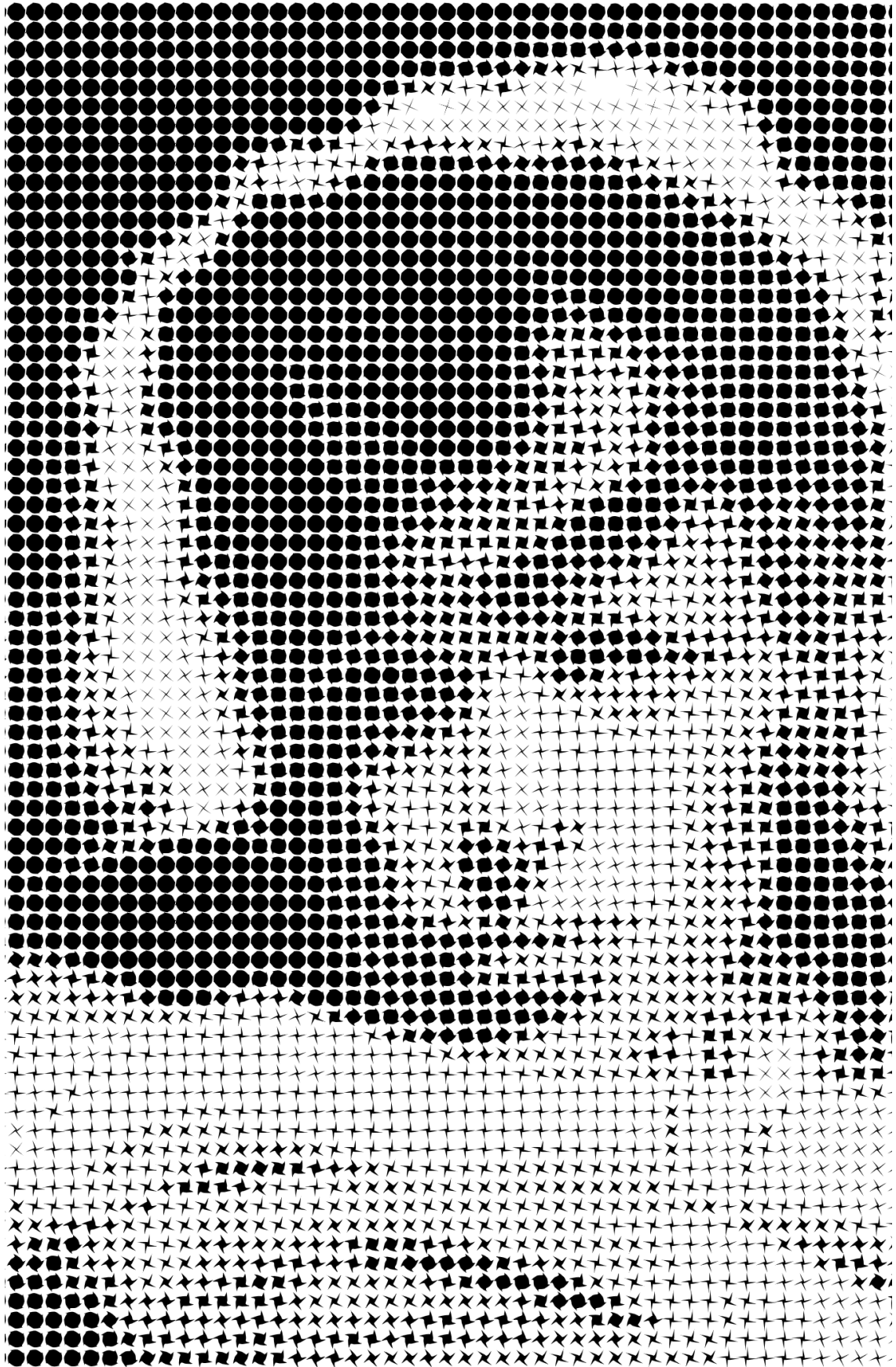
PANIC

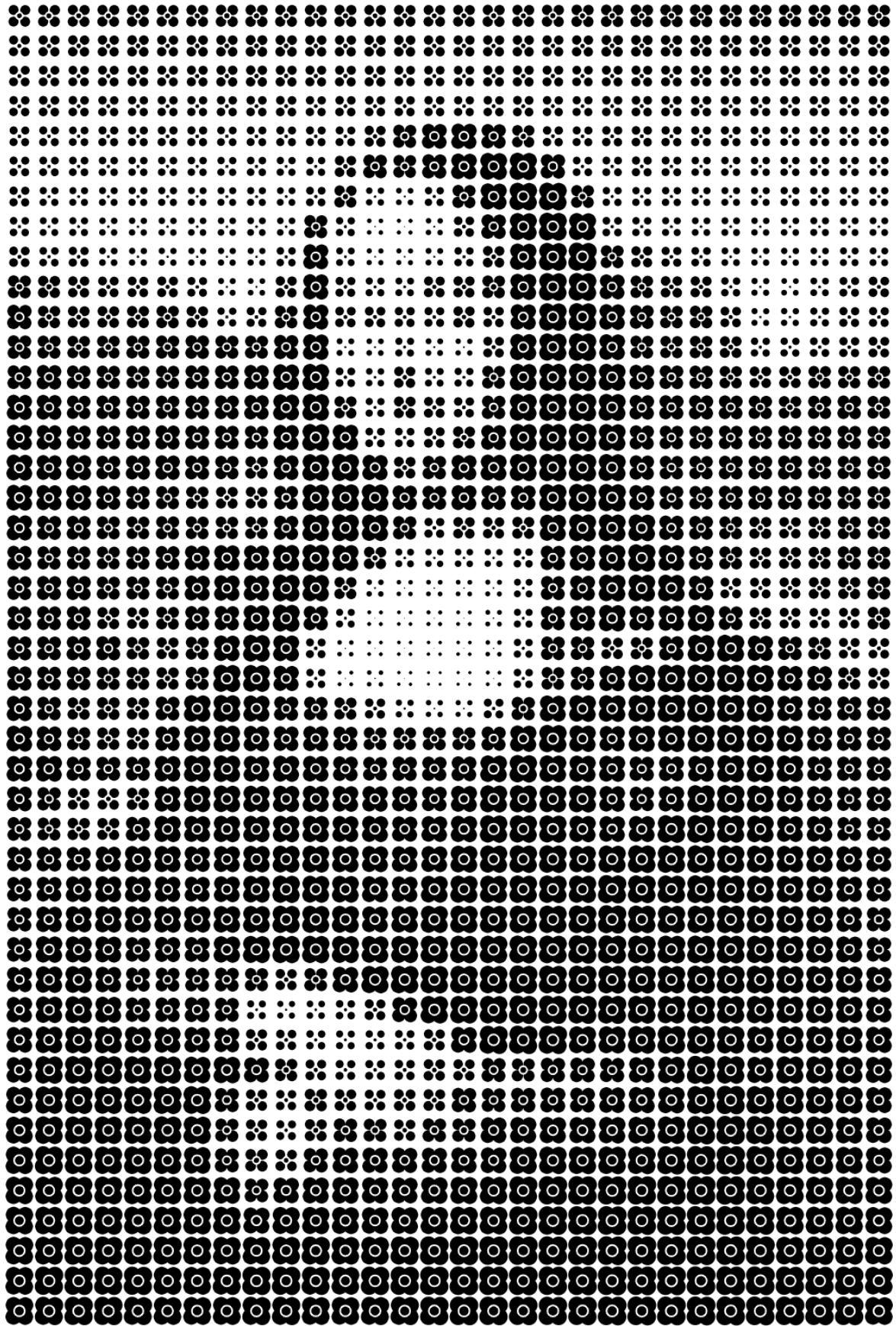






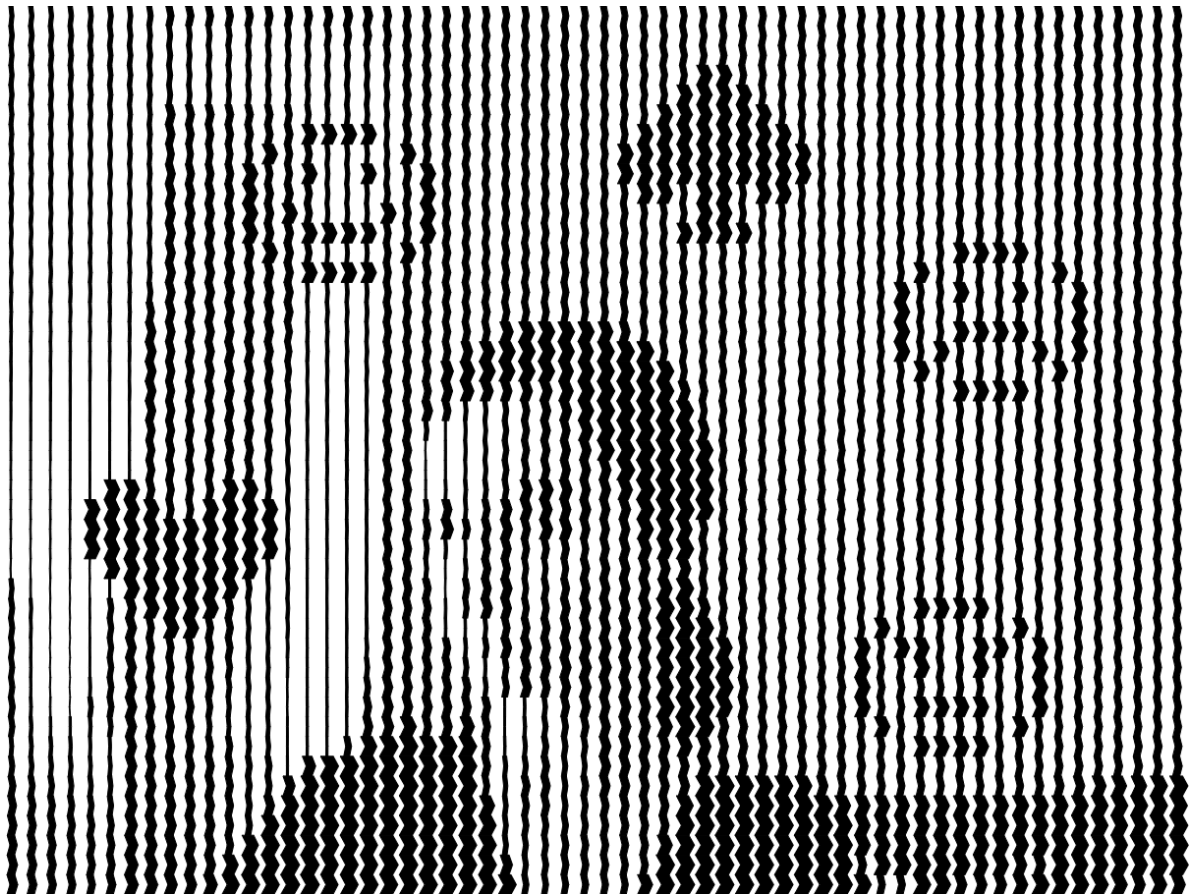


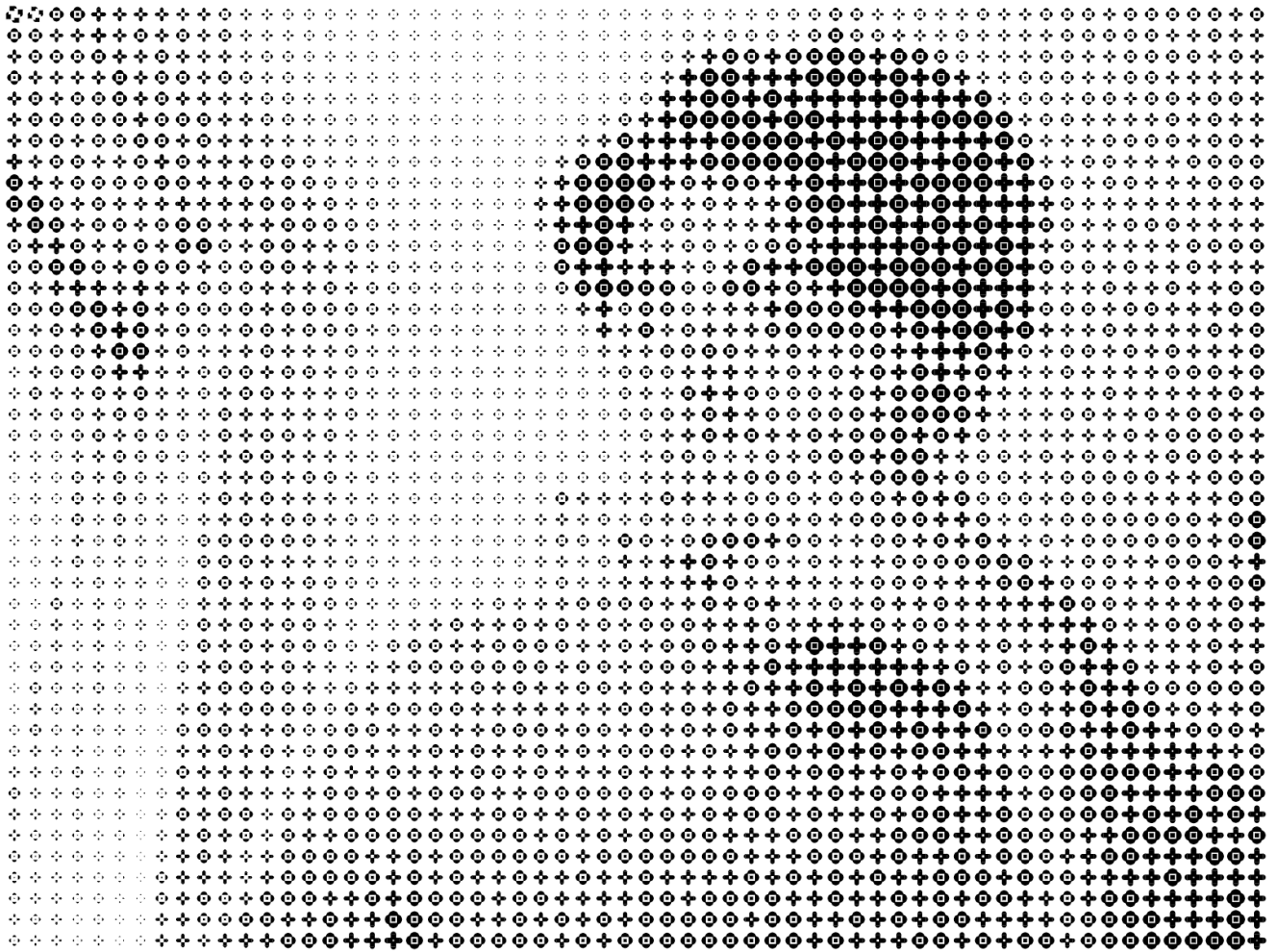


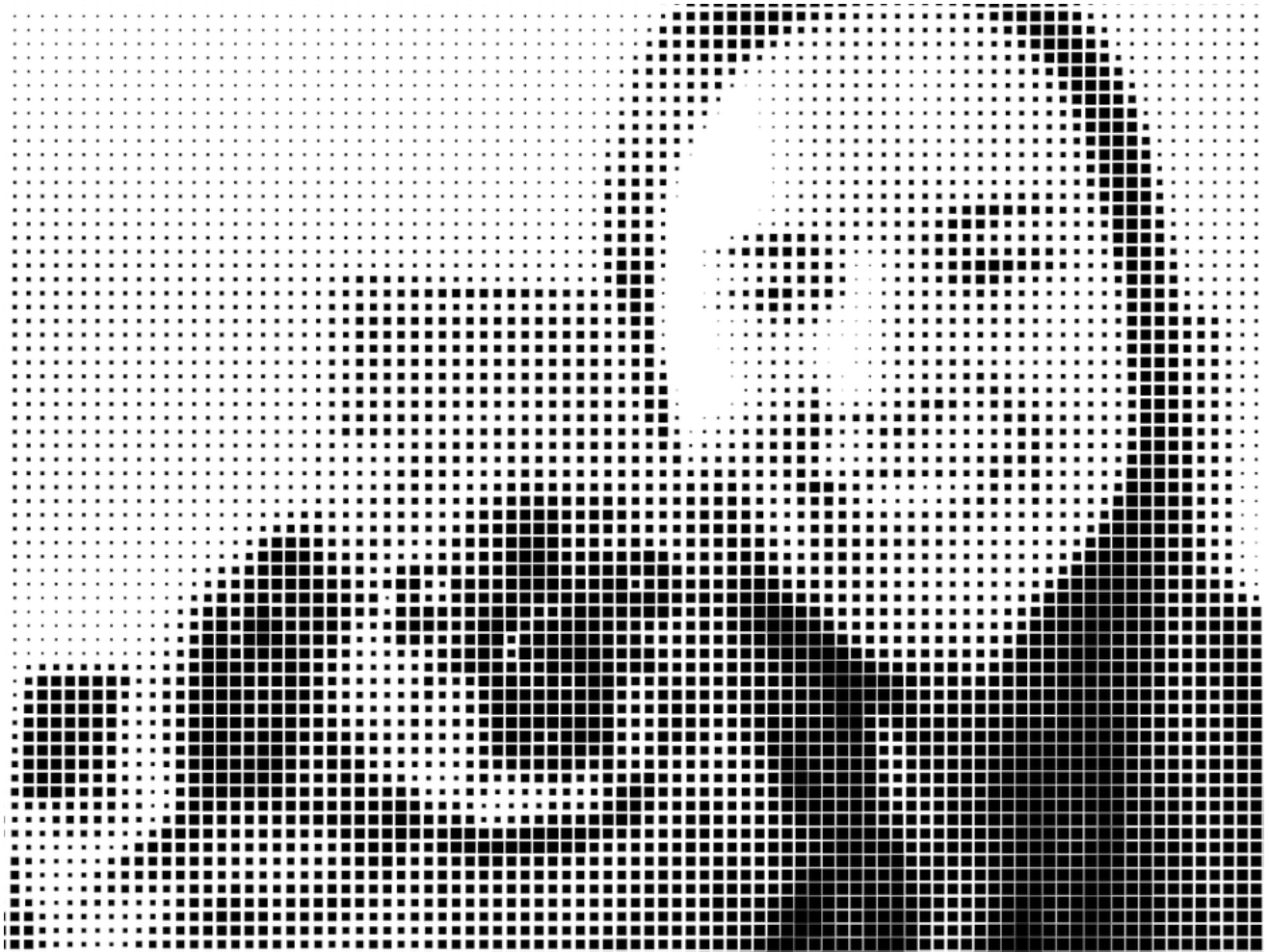




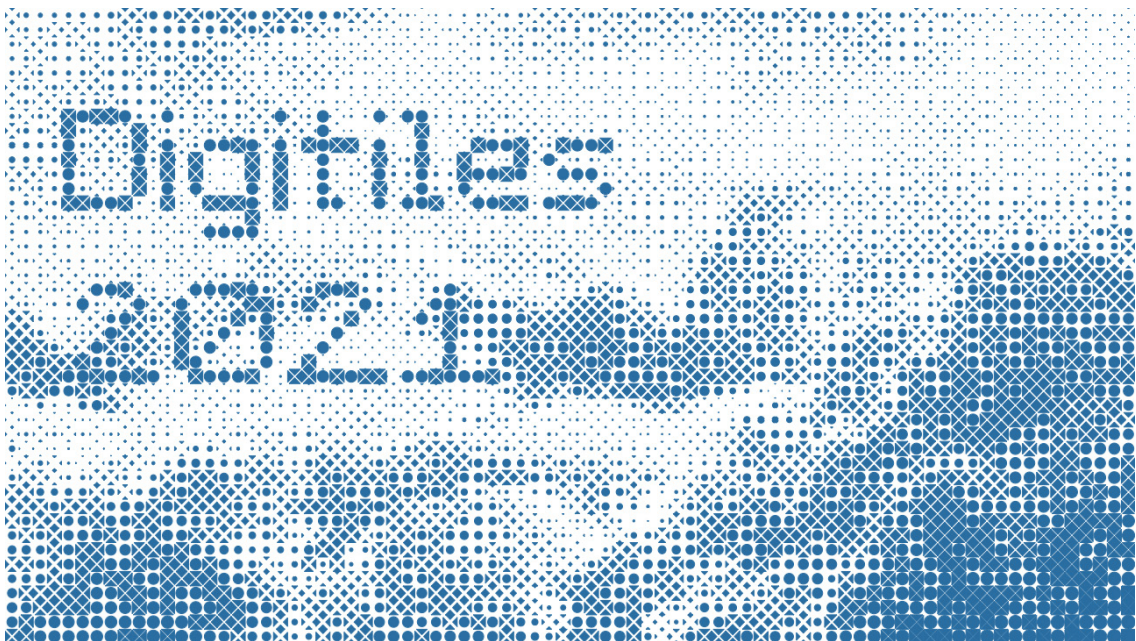


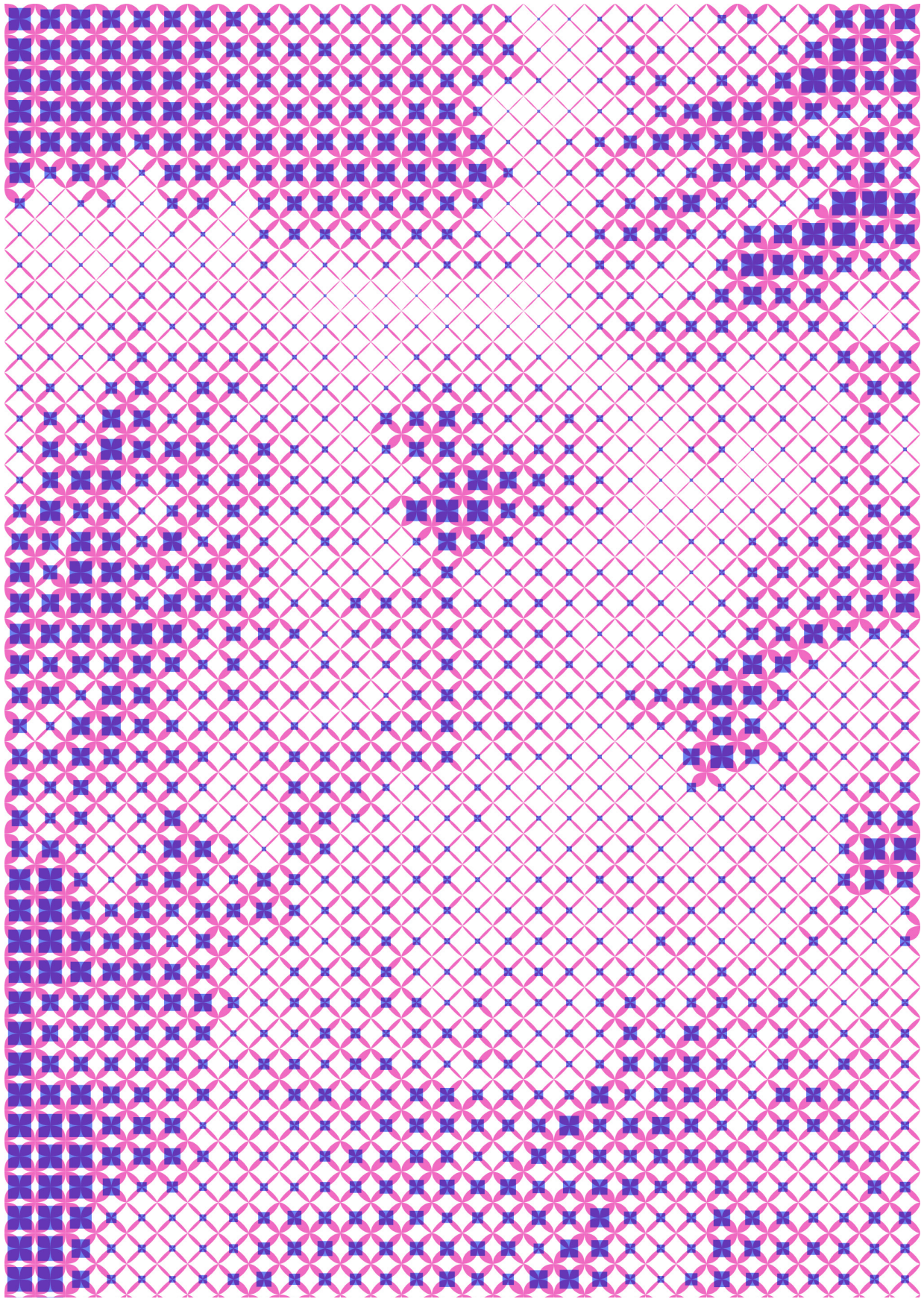












my brain  
is the key



that sets  
free

