1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Ana Rita Horta Estevinha da Silva

Computational Tools for the Development of Dynamic Visual Identities

Dissertation in the context of the Master in Design and Multimedia, advised by Professor João Miguel Andrade Proença da Cunha and by Professor Tiago Filipe dos Santos Martins and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2023

Computational Tools for the Development of Dynamic Visual Identities

→ **Rita Estevinha Silva**
Advisor: João M. Cunha
Co-advisor: Tiago Martins

→ **Master in Design & Multimedia**
Faculty of Sciences & Technology
University of Coimbra
September 2023

Agradecimentos

À minha mãe, porque tudo o que eu sou é o seu reflexo. Ela é, e será sempre, o meu porto seguro.

Ao meu pai por todo o orgulho que demonstra. Tudo o que eu faço é sempre, também, um bocadinho dele.

Ao meu irmão, por caminhar ao meu lado.

À Dra. Carina, porque sem ela este caminho seria bem mais sofrido; porque me ensina a ser quem eu sempre quis ser.

Às minhas amigas da vida toda, porque se há coisa certa são elas.

À Pipa, por ser quem é. Palavras não chegam para lhe agradecer a amizade.

Às amigas e amigos que fiz neste percurso. Em especial, à Carolina, por me acompanhar desde o início; à Catarina, por ser a ótima amiga que é; e ao grupo de DT, pela partilha e convívios.

Ao professor João Cunha, por me ter limpo as lágrimas mais vezes do que eu gostaria; por estar disponível; pela confiança. Ao professor Tiago Martins, por ter sempre algo útil a acrescentar. Nem sempre foi fácil, mas valeu a pena.

A todos os professores que me acompanharam e que fizeram de mim melhor designer.

Por fim, ao João, por me ensinar as coisas que realmente importam: o amor e a vida; por ser a minha pessoa favorita todos os dias.

Resumo

O *Branding* é uma área proeminente do design de comunicação, que se tem vindo a adaptar, ao longo da história, às mais diversas transformações tecnológicas mundiais. As tecnologias e os novos media deste novo milénio fomentaram o desenvolvimento de um novo tipo de identidade visual: Identidades Visuais Dinâmicas. Estas identidades já não são estáticas, ao invés, têm a capacidade de se adaptar continuamente a novos contextos.

Uma abordagem, cada vez mais notável, às Identidades Visuais Dinâmicas é o desenvolvimento de identidades visuais baseadas no sistema — um tipo de identidade visual que consegue garantir o reconhecimento da marca através da sua linguagem visual e design integral, sem necessitar de replicar uma marca gráfica específica. No entanto, nota-se que, apesar dos avanços neste domínio, o *Branding* continua bastante enraizado em concepções tradicionais e modernistas.

Face a estas evidências, esta dissertação propõe uma ferramenta que intui fornecer aos seus utilizadores um conjunto de ferramentas e diretrizes para o desenvolvimento de sistemas visuais dinâmicos. Isto será alcançado ao permitir que os utilizadores incorporem várias formas de dinamismo nos seus próprios elementos gráficos. O objetivo é que, através desta exploração, os utilizadores se familiarizem com métodos para desenvolver as suas próprias linguagens visuais. Para além disso, a ferramenta consegue demonstrar também o potencial e utilidade da automatização de processos de design para alcançar resultados diversos.

Dada a intenção de pretender democratizar esta abordagem ao maior número de pessoas e designers, a ferramenta foi desenvolvida como uma aplicação que pode ser acessível através da internet. A nossa ferramenta pode ser utilizada para experimentação livre — podendo contribuir para encontrar novas possibilidades criativas que poderão ter aplicação em projetos pessoais — ou para desenvolver materiais que poderão responder às necessidades de um projeto de design específico.

Embora não se considere que a ferramenta seja adequada para desenvolver identidades visuais concretas, os resultados demonstraram grande potencial no desenvolvimento de linguagens visuais.

→ Palavras-chave

Identidades Visuais Dinâmicas
Identidades Visuais Flexíveis
Branding
Design Gráfico
Design de Comunicação
Design Computacional

Abstract

Branding is one prominent domain of the communication design field. This domain has been historically adapting to the world's technological developments. The technologies and new media from this millennium have shaped the development of a new kind of visual identity: Dynamic Visual Identities. These are identities that are not static anymore, instead, they have the capability to continuously adapt to new contexts.

One increasingly prominent form of Dynamic Visual Identity is the development of system-based visual identities — an approach that ensures brand recognition through its overall design and visual language, without needing a replicable graphic mark. Nevertheless, the branding field seems to remain firmly rooted in traditionalist and modernist conceptualisations, despite the advancements currently taking place.

In the light of this evidence, we propose a tool designed to provide users with a comprehensive tool kit for the development of dynamic visual systems. This is achieved by enabling users to incorporate various forms of dynamism into their own graphic elements. The goal of this exploration is that users become familiar with methods to develop their own visual languages. Moreover, the tool can demonstrate the potentiality of automating design processes on a parametric basis to achieve diverse outcomes.

Given the intention of democratising this approach to visual identities, the tool was developed as a web-based application accessible to a wide range of users. The tool can be used for free experimentation, which can contribute to finding novel creative possibilities for the development of personal designs and visual identities; or to develop materials that can answer the needs of a specific design request.

Although the system may not be the most suitable for designing realistic identities with all their intrinsic requirements, it has shown great potential in the development of visual languages.

→ Keywords

Dynamic Visual Identities
Flexible Visual Systems
Branding
Graphic Design
Communication Design
Computational Design

0.

Index

1.

Introduction

Acknowledge that we live in a branded world.
(Wheeler, 2013, p. 11)

Brands are present in almost every aspect of our lives. Brands can be products, services, companies, individuals, even ideas. Achieving a desired brand image — referent to the public's mental perception of the brand (Costa, 2004) — is the goal of any visual identity design project (Raposo, 2008). Although a brand is much more than its visual representation, visual identities can, not only, fuel recognition and amplify differentiation as well as make big ideas and meaning accessible (Wheeler, 2013). Great visual identities make a brand excel and stand out from others (Wheeler, 2013).

Technological progress always affects the way brands are managed and how graphic design is practised (Costa, 2004). The fast pace at which the world is evolving demands adaptation from brands, companies and institutions. Today we are living in a globalised and technological era, in which its main characteristics are velocity and mutation (Kreutz, 2007). These concepts are translated into new types of visual identities.

Traditionally, brands strongly relied on a static graphic mark; a symbol or a logotype were the greatest expression of a brand identity and the "quintessential expression of graphic communication" (Blauvelt, 2011, p. 191). These types of visual identities fulfilled their role to unify a company. However, in today's world, they are way too rigid (Blauvelt, 2011). Allied to a remarkable technological development, people started desiring more meaningful experiences and connections from brands (Blauvelt, 2011; Costa, 2009; Nes, 2012), which led to the development of a new era of visual identities: Dynamic Visual Identities (DVIs) (Blauvelt, 2011; Felsing, 2010; Guida, 2014; Lorenz, 2017; Nes, 2012).

Language-like visual identities — a term used by Felsing (2010) — are one form of Dynamic Visual Identity: a type of visual identity that bases its design on the visual language; the visual system guarantees the identity's recognition without needing a graphic mark (Lorenz, 2021). These identities answer better to today's digital demands because they can adapt to different messages, contexts and formats (Lorenz, 2021). This new type of development seems to be an increasing trend in graphic design (Lorenz, 2021), due to the potentialities of automatization in design processes (Guida & Voltaggio, 2016) but also due to the introduction of kinetic possibilities within design (Coelho, 2022).

The present dissertation introduces DynamicIDioms: Towards Visual Languages, a web-based tool that aims to provide users with a comprehensive tool kit for the development of dynamic visual systems based in the system. This can be achieved by allowing users to employ different variation mechanisms to manipulate their own graphic elements. Furthermore, this project aims to demonstrate the benefit of using computational tools for developing contemporary visual identities.

The tool's website can be accessed through the following URL: https://anaestevinha.dei.uc.pt/

1.1
—
Motivation

Brand identity is in the core of the graphic and communication design domain. As the world evolves, so are brands. In such a fast changing world designers must remain adaptable and responsive to change in order to properly answer the demands brought by the contemporary world. The way brands have changed in the past years has been manifested in dynamic approaches, aligning with the use of computational tools in their development.

The use of these tools in the graphic design domain seems to be increasingly prominent and relevant, as it allows novel opportunities to designers. However, there seems to exist a gap between what "cutting-edge" design studios are developing and what designers are learning in schools (Conrad, 2023). Having felt this from past experiences has motivated me to conduct the present project.

Learning computational approaches to design during my Master's degree in Design and Multimedia from the University of Coimbra, made me realise the possibilities of computational tools for contemporary branding, which I did not have previously. In addition, learning about the topic led me to realise that, whilst there is significant theoretical and practical support for traditional visual identities — based on graphic marks —, the current knowledge of system-based visual identities seems to be shorter and more inconsistent. We are determined to produce knowledge in this domain and to create a tool for the exploration and development of contemporary visual identities.

By conducting this dissertation, our primary goal is to enhance the field of Dynamic Visual Identities (DVIs) in a way that not only adds depth to the discipline itself but, more importantly, enriches the understanding and experience of users within this domain. We hope to address the concerns expressed before — the gap between what is being developed and what is being learned — by developing a democratic design tool.

It is our motivation to foster knowledge on this domain so that users can broaden their landscape of expertise to apply to their personal projects and then choose the best practices according to the needs of their clients and/or design requests.

1.2
—
Context

> *"One day I noticed that it doesn't make sense, you make a signet and always add it somewhere. The design itself must take the place of the signet."*— Karl Gerstner, 2007
> (Felsing, 2010, p. 21)

In this dissertation, we aim to address the type of Dynamic Visual Identity that anchors the dynamism in its overall design, usually, dismissing a specific graphic mark. These types of identities can guarantee recognition and coherence through their visual system and graphic elements in all applications and communication content without the need to replicate a logo or a symbol (Felsing, 2010; Lorenz, 2021).

Dynamic Visual Identities that are anchored in the system are usually based on a set of rules that predict the variation and combination of the identity visual elements (Felsing, 2010). These rules allow the system to be in constant change while maintaining a high degree of coherence.

What makes this paradigm increasingly relevant, apart from its customizable and adaptable features, is its potential to be automated on a parametric basis as Karl Gerstner (1964) predicted. Using programming for the design process, not only allows designers to reach higher degrees of complexity and innovative results but also makes it possible to spend more time on experimentation. Moreover, it can guarantee that results are more consistent with the design requirements (Guida & Voltaggio, 2016; Reas & McWilliams, 2010)).

For the development of these visual identities, graphic designers have to define a set of rules and a framework to shape the coherence of the visual content — the designer becomes the one who sets parameters to generate forms (Guida & Voltaggio, 2016), rather than the one who creates them.

It is within this context that we have developed this project: employing computational tools for the development of system-based visual identities.

1.3 — Goals

In recent years, researchers have developed guidelines and models for the evaluation and development of DVIs. Even though most authors acknowledge that their research may not be final, we find it relevant to understand how these proposals can be applied for teaching and exploring these types of visual identities.

The main goal of this project is to develop a computational tool for users to become familiar with processes and mechanisms to design coherent visual languages, through a parametric manipulation of different visual elements. We expect that by freely interacting with the tool, users will not only learn guidelines for the development of dynamic visual identities but they will enrich their creative landscape.

Although the tool might allow the production of visual results to apply in realistic contexts, the main purpose of this tool is, above all, to be exploratory and didactic. We are not committed to developing a tool that produces a visual identity in its full complexity, but rather to enriching users about contemporary visual identities.

We have identified a set of goals to be met in the present dissertation:

1. to study and analyse existing models and guidelines for the evaluation and development of Dynamic Visual Identities, to support our project development;
2. to study and analyse system-based visual identities, understanding their characteristics and requirements;
3. to develop a computational tool for the generation of coherent visual systems;
4. to evaluate the tool and apply results in different applications, to assess the system's potential.

1.4

—

Document Structure

→ 1. Introduction

This chapter introduces the project's subject and addresses the motivations behind the development of this dissertation, as well as its context and goals.

→ 2. Development Plan

This section addresses the plan that was formulated to successfully complete this project, as well as the alterations that had to be made along the development process. Moreover, it presents the work methodology that was adopted throughout the system implementation.

→ 3. State of the Art

In this chapter, we present a brief history of Dynamic Visual Identities; we review guidelines and models proposed by diverse authors for the comprehension and development of Dynamic Visual Identities; we address visual systems and their application throughout design history; finally, we showcase computational tools that were developed for the context of system-based visual identities.

→ 4. An Analysis of DVIs

We present an analysis of a sample of dynamic identities that anchor their dynamism within the system in order to better understand their features and possibilities.

→ 5. Project

We present the body of work that has been developed since the beginning of this dissertation. Mostly, we showcase the tool that was developed highlighting its functionalities and potentialities. Additionally, we show some generated results, which were beneficial to understand the tool's potential. We also discuss possible future work to be developed regarding this project.

→ 6. Conclusion

Lastly, final considerations about the work developed are discussed.

2.

Development Plan

2.1
—
Development Plan

Four main tasks were found crucial for the development of the present dissertation. For each task, we predicted the associated workload and duration. These tasks were: (i) bibliographic review and research; (ii) dissertation writing; (iii) project development; and (iv) results and system evaluation.

→ Bibliographic Review and Research

This task concerned the investigation of the current state of the art on the domain of dynamic visual identities. We looked at their history; reviewed existing perspectives and proposals for the evaluation and comprehension of this kind of visual identity; looked into the concept of visual systems, and demonstrated how technology can enhance the development of more complex visual identities and optimise design processes through the analysis and presentation of case studies.

→ Dissertation Writing

The writing task was split into two phases: an intermediate and a final writing. The intermediate phase was relative to the state of the art and theoretical research developed in the previous task. Moreover, it included an initial system definition as well as the presentation of a preliminary project related to the purpose of the present dissertation. The final phase entailed writing the project development, documenting its results and writing the conclusion. Additionally, during this phase, we refined the earlier state of the art and introduced new concepts, considered relevant for the domain of this dissertation.

→ Project Development

The project development was divided into two tasks: an analysis of system-based visual identities; and the system development. The analysis consisted in reviewing a sample of dynamic visual identities, examining their variation mechanisms and dynamic components in order to provide us with insights for the next task, the system development.

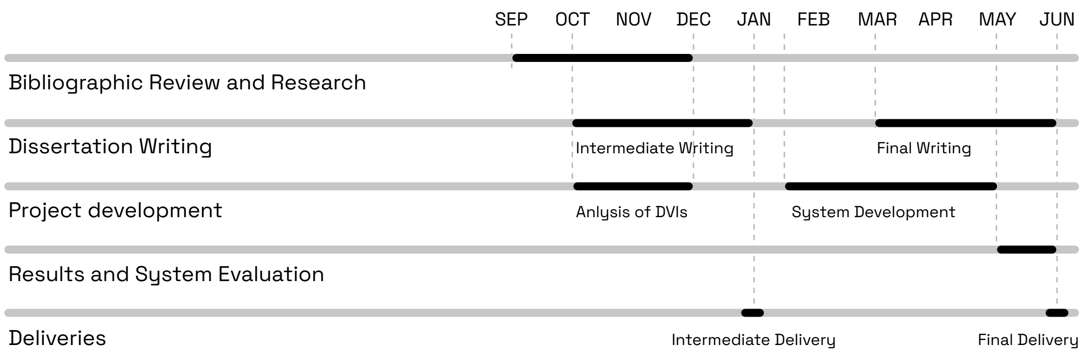
The system development consisted in building a web-based tool. Given the dimension of this task, it required its compartmentalisation in smaller tasks: design and conceptualisation; testing and implementing variation mechanisms; testing and implementing the grammar concept and sequencing of actions; implementing the remaining features; and web development. In the end of these tasks we expected to have a complete system and functional tool.

→ Results and System Evaluation

This task served to evaluate the system and assess its potential. The system was tested in different contexts to understand its versatility in answering the goals we had predicted initially.

A chart was developed to predict the duration of each of these tasks. At the beggining, we had planned to deliver in July (Figure 1). However, the delivery had to be extended as seen in Figure 2. This happened due to personal programming limitations. It is important to note that I have started this project with short knowledge of programming, and almost none regarding web development programming languages. The fact that we had to switch from p5.js to JavaScript (see section 5.2) had delayed our process, as it required a great investment of time and effort to get the necessary knowledge for the system development.

Figure 1. Initial chart with predicted tasks.



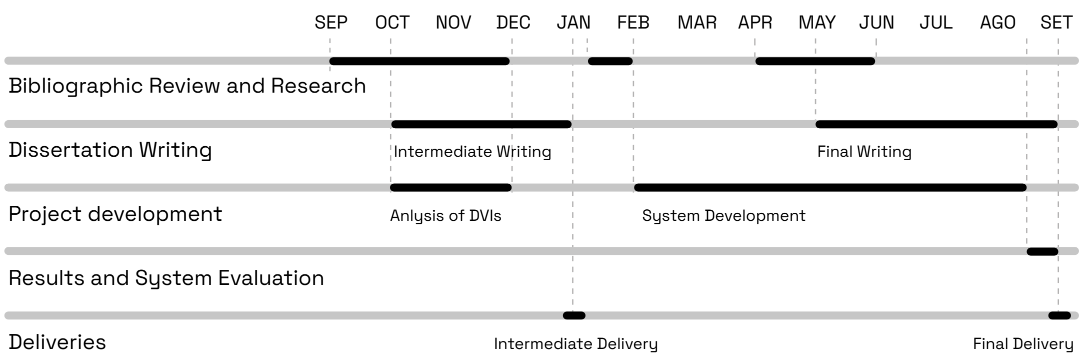


Figure 2. Realistic chart of the dissertation's development.

Furthermore, we had intentions of conducting tests with users, but since the development of the system took longer than expected, we have decided to generate personal results and perform a personal evaluation. Although we could not conduct them within the given time, we predict to continue testing the tool for future purposes.

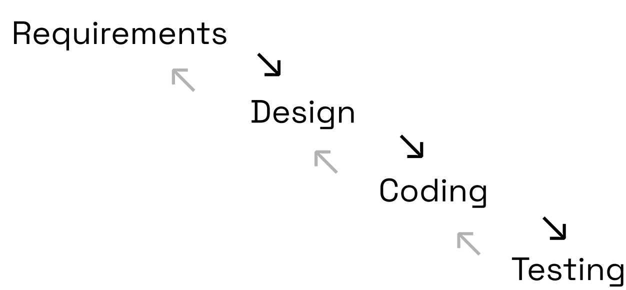
Nevertheless, we implemented features into the system that were not initially anticipated, which, along the process, we considered more valuable for the project than strictly following the initial plan. Even though the planning did not occur exactly as estimated, we overcame the obstacles and exceeded our expectations.

2.2 — Methodology

Considering that the practical project was mainly focused on software development, we have adopted a *Waterfall Lifecycle* methodology (Dubberly, 2004). This method is a model for software development that suggests that transitions to the next tasks can only be made once the previous task is completed, as the achieved output from one phase serves as input to the next (Dubberly, 2004). This approach also implies that when an issue arises in one stage, it is necessary to revisit the previous phase to address the problem and guarantee a coherent development (Dubberly, 2004). Adopting this methodology was beneficial because it forced us to maitain a focus on the system requirements that were initally defined.

The model notices 4 phases: (i) requirements; (ii) design; (iii) coding; and (iv) testing (Figure 3). We have adjusted this methodology for the context of this dissertation. Requirements consisted in defining comprehensively the system functionalities after analysing the state of the art and the sample of DVIs. In the design phase, we began conceptualising the project and crafting the system's architecture as well as interactive possibilities. The coding phase was focused on system development. While developing our system, the stages "design" and "code" involved a back and forth process, with continuous iterations and adjustments. Lastly, the testing phase consisted in generating various results in order to analyse the system interaction and the potentialities of its outcomes.

Figure 3. Methodology adopted for the development of this dissertation. *Waterfall lifecycle*, Philippe Kruchten, 2004.



3.

State of the Art

3.1

—

From Static to Dynamic

Trademarks, symbols, signs have been around since the time people created goods to trade or sell (Landa, 2005). Brands have emerged in a trading context, due to necessity, and continued to evolve until today in complex ways (Costa, 2004). Joan Costa (2004) refers to four different brand "births" throughout history, which not only represent brand evolution but also depict important periods of modern societies and cultures. These stages consider functional, utilitarian and semiotic features of brands that remain altering through time (Raposo, 2008).

The author (Costa, 2004) considers that brands started in Antiquity, when signs were used to identify agricultural products and cattle. They evolved in the Middle Ages, as economic activity shifted towards craftsmanship and manufacturing, brands began to symbolise quality and trust. The third birth is referent to the Industrial Economy, when an increasing market culture implied higher differentiation from brands. This modern brand culminated in "corporate brands", in which the visual identity became an integral part of a company, reflecting its culture, values, mission and personality. Some well-known visual identities developed by Paul Rand, Saul Bass, Otl Aicher, Josef Muller-Brockmann and Muriel Cooper are benchmarks of this period in history (Meggs & Purvis, 2012).

In the final stage, which represents the contemporary scene, brands transcend their commercial function and become an agent of emotional experiences as well as cultural and social transformation (Raposo, 2008). Bauman (2000) uses the concept of liquidity to translate the present phase of history: a world where institutions, ideas and relationships change very quickly and unpredictably.

In the final stage, which represents the contemporary scene, brands transcend their commercial function and become an agent of emotional experiences as well as cultural and social transformation (Raposo, 2008). Bauman (2000) uses the concept of liquidity to translate the present phase of history: a world where institutions, ideas and relationships change very quickly and unpredictably.

> *What all these features of fluids amount to, in simple language, is that liquids, unlike solids, cannot easily hold their shape. Fluids, so to speak, neither fix space nor bind time. While solids have clear spatial dimensions [...], fluids do not keep to any shape for long and are constantly ready (and prone) to change it;*
> (Bauman, 2000, p. 2)

The author expresses that the solidity of the past, that should represent predictability, security and trust, started to slowly disintegrate.

This led to distrust in institutions. So modern times started showing the "wish to discover or invent solids of — for a change — lasting solidity, a solidity which one could trust and rely upon" (Bauman, 2000, p. 3). This reinvention is translated in the transition from solid to liquid: a melting.

These concepts are very clearly reflected in branding history. The corporate brand — a symbol or a logotype unifies the visual identity system and is the most fundamental element — that once reflected reliability, stability and uniformity is way too rigid for the contemporary world (Blauvelt, 2011; Felsing, 2010). Static and standardised visual identities no longer answer the demands of contemporary communication (Lorenz, 2021), given their limitations in adapting to an increased complexity. Moreover, the conceptions behind this corporate paradigm are increasingly conveying a sense of scepticism and distrust (Grant & Vodeb, 2023). Brands and people are demanding transparency, emotional experiences, participatory approaches, commoning and diversity (Grant & Vodeb, 2023).

The concept of variable visual identities is not entirely novel. For example, in 1915, the graphic mark for the publishing house Alfred A. Knopf used a variation of a borzoi symbol for each book (Figure 4) (Martins et al., 2019; Pearson, 2013). Nevertheless, this approach was not the main paradigm, mainly because the technological limitations and lack of resources of that time narrowed the possibilities for crafting variable visual identities (Lorenz, 2017).

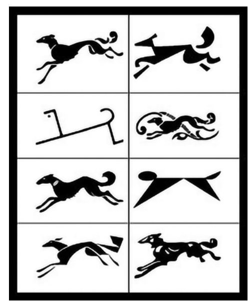


Figure 4.
Borzoi graphic mark, Alfred A. Knopf, 1915.

Later, in 1981, Manhattan Design studio presented the MTV logotype. They created an identity capable of addressing thousands of variations of colours, patterns, textures, animations and illustrations (Blauvelt, 2011; Meggs & Purvis, 2012; Nes, 2012). In the final decades of the past century, this identity gained increasing relevance as the world was entering a new technological era. The transition from traditional print media to new emerging digital platforms reshaped the nature of visual identities and the MTV logo (Figure 5) was clear evidence of a new visual identity paradigm (Meggs & Purvis, 2012).

> *The age of dynamic identity was born.*
> (Blauvelt, 2011, p. 194)

The turning of the millennium, marked by the internet democratisation, arising of social media platforms and technological breakthroughs, has witnessed an increasing number of dynamic visual identities (Blauvelt, 2011; Felsing, 2010; Guida & Voltaggio, 2016; Nes, 2012). This transition was mostly practical due to generative tools, as they allow the generation and management of visual artefacts in easier and novel ways (Guida, 2014; Reas & McWilliams, 2010).

Figure 5.
MTV logos from 1981-82, designed by Manhattan Design.

For example, the identity for Casa da Música, developed by Stefan Sagmeister, in 2007, was one of the first employing generative methods (Guida, 2014). A logo-generator allowed the output of diverse forms of the graphic mark, based on different perspectives of the building's architecture; colours could vary from uploaded images (Figure 6) (Blauvelt, 2011; Guida, 2014). Given the flexible features of the institution — hosting various and varied events — having a dynamic identity allows the brand to adapt to all these different contexts. The software offers a wide range of solutions not only for audiences but also for those who have to manage the identity every day (Guida, 2014).

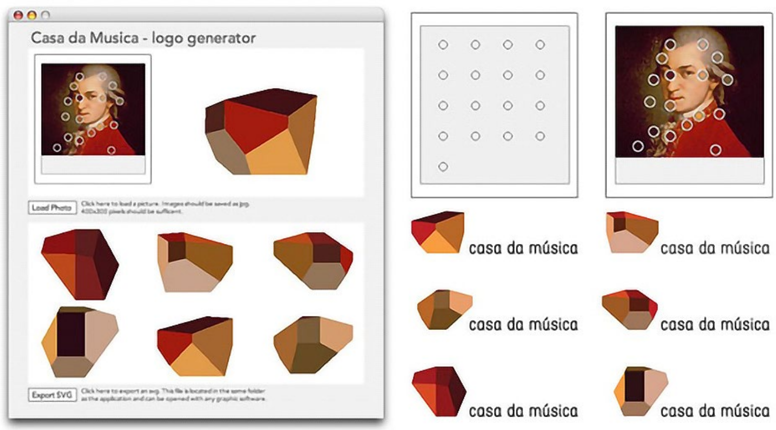


Figure 6.
Casa da Música logo generator, Stefan Sagmeister, 2007.

Another example is the identity for the Scandinavian peninsula of Nordkyn (Figure 7), which was designed by Neue Studio, in 2009 (Guida, 2014; Martins et al., 2019). This visual identity was based on a logo generator that would change according to real-life nature conditions (Guida, 2014; Visit Nordkyn, n.d.). Many other examples from this period are recognized as landmarks of dynamic visual identities such as Mit Media Lab's identity designed by TheGreenEyl, E Roon Kang and Richard The in 2010, that was capable to generate over 40,000 permutations of a logo (MIT Media Lab Identity, n.d.), or AOL's identity developed by Wolff Ollins in 2009, which changed its content endlessly (Martins et al., 2019; Nes, 2012), among others (Martins et al., 2019; Nes, 2012).

With the growing evolution of such visual identities, there has been a simultaneous effort to characterise, identify, and comprehend this new paradigm. The following section studies the proposals and thoughts on this domain for a better understanding of Dynamic Visual Identities.

Figure 7.
Nordkyn logo variations, Neue Studio, 2009.

3.2 — Dynamic Visual Identities: Existing Perspectives

Many authors have proposed different guidelines, models and proposals for the analysis and development of Dynamic Visual Identities (DVIs). Moreover, their nomination varies from author to author: dynamic (Jochum, 2013; Martins et al., 2019; Nes, 2012), flexible (Felsing, 2010; Hollington, 2011; Lorenz, 2021), fluid (Murdock, 2016; Pearson, 2013) mutant (Kreutz, 2007, 2012; Leitão et al., 2014). The following paragraphs address different perspectives on the regard of Dynamic Visual Identities in chronological order.

→ Mutant Brands, Elizete de Azevedo Kreutz

Kreutz (Kreutz, 2007, 2012) has distinguished Conventional Brands — a traditional conception of visual identities based on modernist design theories — from Mutant Brands — post-modern and contemporary visual identities. Conventional brands can be classified as Stereotypical or Arbitrary. Mutant brands are identities that are subject to transformations and, according to the author, can be Programmed — mutations occur in a determined period — or Poetic — mutations occur spontaneously without predetermined rules. These last identities can be flexible and adapt to new events.

Furthermore, Kreutz (2007) believes these identities can and should answer to contemporary communication demands such as: participating in the organisation as a whole; materialise the spirit and the emotions; capturing the expectations of the public; inciting the desire to participate; representing the common desire; have an enveloping structure; break with the mechanical, static view of a univocal image; have identity traits that allow the viewer to identify of their values in the observed object.

→ Flexible Visual Identities, Ulrike Felsing

The study of Felsing (2010) addresses diverse variation processes for the development of flexible visual identities (FVIs). The author points to the following processes: Content and Container: Masks / Grids; Element and Sequence: Movement / Change of Perspective; Theme and Variation: Transformation; Combinatorics: Rapports / Modules / Elementary Construction Kits; Element and Structure: Permutation; Interaction: Control Factors / Transfer / Open Form and presents case studies for each of these processes proposals.

Moreover, the author notices that these variation processes can be applied in a logo variation, in a sign family, or in a "visual language" — a language-like visual identity in which "nearly every aspect of this writing-based communication incorporates the constant of the visual identity" (Felsing, 2010, p. 21).

In her study, she argues that static visual identities have less potential to adapt to changing contexts as they run the risk of losing coherency. In flexible visual identities the constant elements assume the function of a graphic mark — ensure the recognition of the visual identity — while the varying elements can adapt to the particular changing aspects of a brand, "such as when companies merge or their product range expands" (Felsing, 2010, p. 220). A dynamic visual identity is forward-looking when encountering a future development of new contents (Felsing, 2010).

→ Flexible Visual Identity Systems, Stuart Hollington

Hollington (2011) points out six types of Flexible Visual Identities: Rearrangeable logo, Deconstructable logo, Variable content logo, Variable container logo, Single logo and Language-like system. Although five are rooted in logo dynamism, the author recognizes language-like systems as "non-logo based" (Hollington, 2011, p. 39), and the visual language defines the overall visual identity. The author also defines four types of flexibility: Adapt, Transform, Move and Interact which can be achieved by different types of Visual Identities. (Hollington, 2011)

A visual exploration developed by the author, using simple forms and combining different sets of fixed and variable mechanisms, shows a myriad of coherent patterns and outcomes that can be used in DVIs (Figure 8).

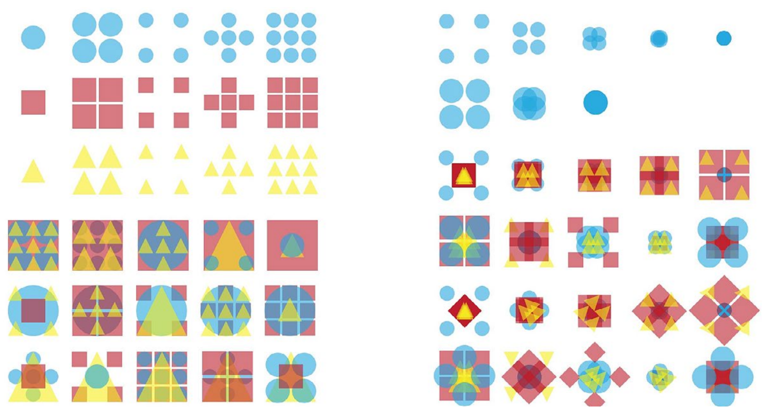


Figure 8. Visual experiments developed by Hollington (2011).

→ Dynamic Identities, Irene van Nes

Nes (2012) categorised six types of DVIs based on their construction: Container, Wallpaper, DNA, Formula, Costumised and Generative. For Nes, six components define an identity system: logo, colour, typography, graphic elements, imagery and language. Together these components build the brand identity. These elements can define different graphic variables that will add dynamism to an identity, as opposed to a mere logo variation. She presents a diagram with these six components that compose a visual system where pushing the limits of these components can be representative of where dynamism is applied within a system (Figure 9). One or more components can be variable, adding flexibility to the identity.

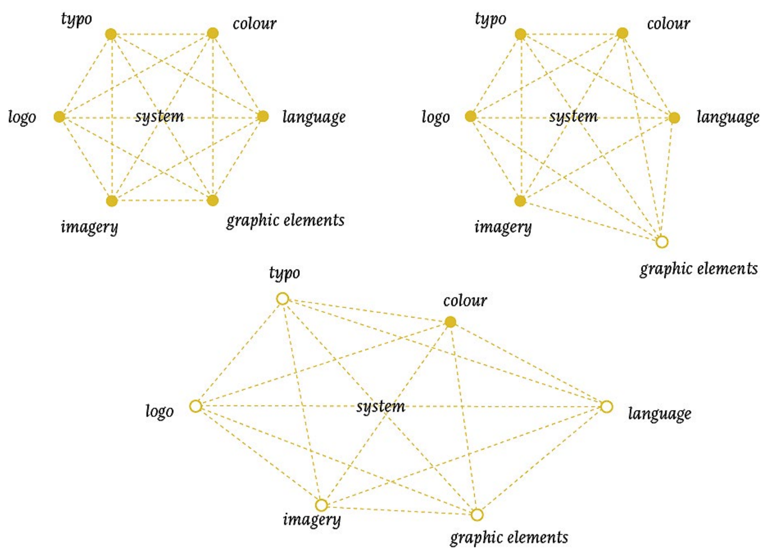


Figure 9.
Diagram of dynamic components (Nes, 2012).

Dynamic Identities require a balance between variable and fixed components to guarantee coherence. While variable components give room for the identity to live and evolve, fixed components help users to recognize their brand (Nes, 2012).

→ Dynamic Branding, Emanuel Jochum

Jochum (2013) defines his model based on a refinement of the variation processes proposed by Felsing (2010) and the categorization established by Nes (2012). The author explores how flexible design systems turn brands into dynamic visual identities and proposes six different methods to achieve that: Filling and Container, Background and Layer, Combination and Composition, Transformation and Adaptation, Customisation and Collaboration and Automation and Transfer.

Moreover, resorting to Nes' identity system definition, the author uses a flexibility chart to evaluate which components are variable and which are fixed showing the degree of dynamism of existing DVIs.

→ Fluid Marks, Lisa Pearson

Pearson (2013) established a taxonomy system for categorising different species of fluid marks. The author acknowledges that some fluid marks are hybrid, meaning they can exhibit characteristics of more than one species. Moreover, she does not consider this taxonomy to be closed, as she believes more species can be discovered. The author also discusses legal concerns brand developers need to consider when creating these marks, providing insights on how to avoid future risks and how to protect these types of brands.

Pearson defines seven species that arise from the traditional mark: Ornamenting the mark, Reinterpreting the mark in different media, Filling a frame, Changing the background, Employing moving designs, Adopting multiple designs, Using ever-changing designs (Figure 10).

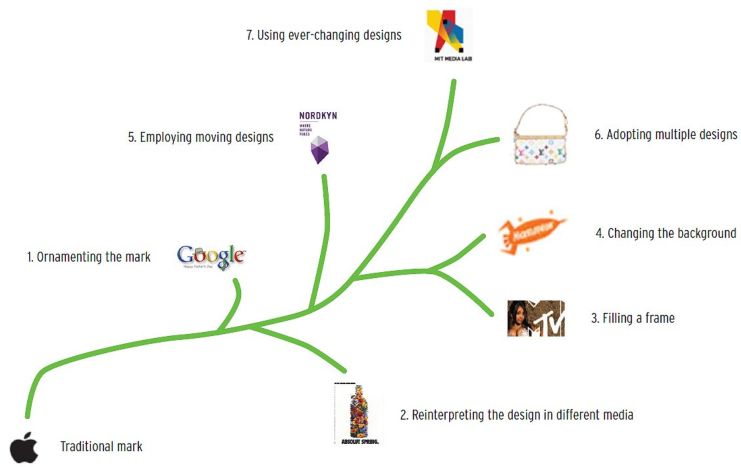


Figure 10. Pearson's taxonomy of fluid marks (Pearson, 2013).

→ Fluid Identity, Jason E. Murdock

Murdock's (2016) noted techniques to develop dynamic identities are Modularity, Permutation and Open form. These processes can vary from simple to complex and from more or less control of outcomes. Yet, they are broad enough to allow the designer to develop language-like visual systems — identities that base their recognition on the overall system rather than the logo — as the author believes these are the most basic form of DVIs. He supports these types of identities as a different paradigm from traditional branding based on modernist and industrial trends. "Dynamic visual identities, however, are often decentralised in their system architecture, and the logo — if one exists — plays no more important a role than any other element." (Murdock, 2016, p. 17). He considers this paradigm to be poorly studied by historians and designers.

→ Dynamic Visual Identities, Tiago Martins, João Cunha, João Bicker & Penousal Machado

Martins et al. (Martins et al., 2019) elaborated an in-depth model for evaluating Dynamic Visual Identities. The authors recognized seven features that can characterise a DVI — Flexible, Fluid, Generated, Informative, Participatory, Reactive, Unlimited — and eight variation mechanisms — Colour variation, Combination, Content variation, Positioning, Repetition, Rotation, Scaling, Shape transformation — that produce the dynamism using the components of the identity. Although these mechanisms seem basic in themselves, their combination allows high degrees of dynamism. The authors do an extensive analysis of dynamic visual identities, categorising their features and identifying the employed mechanisms.

The approach does not consider features or variations as being isolated concepts; instead it shows that features and variation mechanisms can co-exist in different ways, fostering different outcomes and opportunities. Furthermore, the study distinguishes identity focus: graphic mark focused or system focused (Martins et al., 2019), acknowledging that some visual identities base their dynamism on the logo, while others are system-based. Their analisis has a bigger sample on DVIs that are logo-based, denoting a higher development on this approach. However, their analysis only considers DVIS between 1959 and 2013.

→ Flexible Visual Systems, Martin Lorenz

Flexible visual systems are visual identities that dispense a graphic mark (Lorenz, 2021). The identity recognition is based on a coherent system of diverse graphic elements rather than in a repeatable, single logo. This approach goes beyond dynamic logotypes and it highlights the type of identities that Felsing (2010) referred to as "language-like visual identities", the same type of identity Karl Gerstner (1964) coined as "programmes".

The author (Lorenz, 2021) is committed to the belief that this type of developing visual identities is the future of communication as "we need contemporary visual languages to be able to solve contemporary communication problems" (Lorenz, 2021, p. 8).

For him, there are two types of Flexible Visual Systems: those based on forms — Form-based Visual Systems — and those based on transformations — Transformation-based Visual Systems (Lorenz, 2021, pp. 48–49). Although the author does not explore as much other types of flexible systems, he mentions other possibilities such as Object-based (3D symbols, objects and structures) and Interaction-based (intersections, exclusions, multiplications, etc.) (Lorenz, 2021, pp. 213–232).

Overall, these systems can be closed, not being able to add new elements, or open, allowing the identity to continuously evolve. The author argues that any flexible visual identity needs to receive an input to be able to produce an output that can be responsibly applied (Lorenz, 2021, p. 48). Moreover, he demonstrates four different ways to adapt to a grid of a format, these being: Position, Repetition, Distortion or Combination of the previously mentioned.

Because these systems are based on rules, the author advocates for automating everything. As he puts it, the designer's task "should be to develop concepts and systems, not to perform mindless repetitive tasks" (Lorenz, 2021, p. 19).

His book is a showcase of various forms of producing different types of visual identities being by the manipulation of different components able to create visuals, typography, frames through simple combinations; or by transforming its graphic components (Lorenz, 2021).

Throughout our description of existing perspectives, it is possible to perceive that the discipline does not recognize or agree on a unique definition. There are many possible ways to achieve dynamism in visual identities and many different categorisations and approaches. Yet, authors agree that most Dynamic Visual Identities are still very logo-centred (Jochum, 2013; Lorenz, 2021; Martins et al., 2019; Murdock, 2016). Jochum even argues that “Logo-centrism is still a major issue and unfortunately makes several visual identities very weak in the implementation” (Jochum, 2013, p. 69).

According to Lorenz (2017) we are witnessing not just a trend but a shift in how organisations, institutions and corporations are communicating, defending that the system-based identities are better to solve current communication demands. The aim of this project is to address this approach of developing dynamic visual identities. Therefore, the following section focuses on visual systems and system-based visual identities.

3.3
—
Visual Systems

As already highlighted, not only the studies exhibit a logo-centrism in this domain as the examples shown previously were based on dynamic logos (see section 3.1). Another trend in Dynamic Visual Identities is the creation of visual languages. Language-like visual identities — a term used by Felsing (2010) — are a type of visual identities that base its overall identification on the visual system, meaning the identity’s recognition is achieved through its visual elements without needing a graphic mark.

According to Martin (2021, 2017), these system-based identities answer better to today’s digital demands because they can adapt to different messages, contexts and formats, while dynamic logos are only variations of the same element. Although dynamic logos can bear more messages and meanings than static logos, they also entail some limitations compared to system-based identities, for example, when trying to adapt to different formats. For the author, dynamic logos are a transitional phase between static graphic marks and system-based visual identities (Lorenz, 2021, 2017).

It is important to acknowledge that visual systems in communication existed earlier to digitalisation. For instance, since the beginning of communication there has been a systematisation of written language to enhance the recognition of symbols and facilitate effective communication among humans (Lorenz, 2021). In his research, Martin (2021), identifies five different types of design systems along history. Although they are from the typographic domain, the author argues they can be seen in most graphic design systems. These systems are: The Manual, The Stencil, The Building Blocks, The Program and The Tool (Lorenz, 2021).

The Manual is essentially a guide for individuals who may not have been familiar with the design implementation and need to reproduce or apply it (Lorenz, 2021). In history, the author gives the example of Romain du Roi, a typeface commissioned by King Louis XIV, in 1692, for use by the Royal Print Office. The guidelines to construct this typography would be based on a square grid (Figure 11), which allowed the creation of a consistent typeface. Given that the function of this commission was to design all fonts to be used in French official documents, we can speak of a system-based visual identity (Lorenz, 2021).

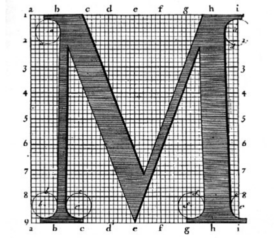


Figure 11.
"M" from Roman du Roi, designed by Jean Truchet.

In the context of corporate visual identities, graphic standards manuals serve as guidebooks that define the rules for displaying logos and visual systems. The goal of these manuals is essentially to transmit to non-creator designers how to correctly apply the logo, use corporate colours, and typographic palettes, and follow the formats without diverging even a fraction from the established guidelines (Heller, 2014). Numerous renowned manuals from the era of the "corporate brand" (Blauvelt, 2011), such as the ones from IBM, Lufthansa, the New York Subway System, and NASA, are widely acknowledged worldwide as benchmarks in design (Heller, 2014).

However, some authors believe this approach is an authoritarian way to institute a visual regime from top down, and to enforce its implementation through explicit guidelines and strict "policing" (Murdock, 2016, p. 8). Moreover, most manuals leave room for misinterpretation, thus failing to achieve their intended purpose (Lorenz, 2021).

The Stencil functions as a template which allows unskilled and inexperienced people to reproduce the system accurately. Lorenz (2021) gives the example of the Plaque Découpée Universelle (Figure 12), from 1876, that allowed the production of a typeface, in lower and upper case. The concept of using stencils is still evident in certain dynamic visual identities, like the previously mentioned MTV logo. In this logo, a template is employed while content is variable. (Lorenz, 2021; Nes, 2012).

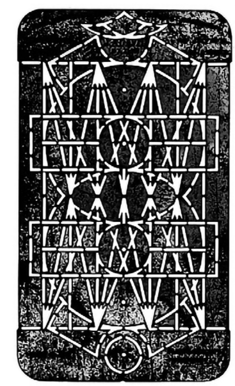


Figure 12.
Plaque Découpée Universelle (PDU), by Joseph A. David.

The Building Blocks can be interpreted as modular systems. There is a selection or creation of graphic components that are combined in different forms, producing varied outcomes (Lupton, 2011). Having limited and recognizable components can make the system identifiable, even when the components are rearranged (Lorenz, 2021).

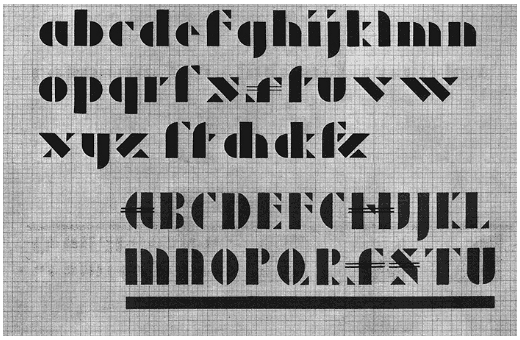


Figure 13.
Kombinationsschrift,
by Josef Albers.

One example is the writing system *Kombinationsschrift* by Josef Albers, from 1931 (Figure 13). This system uses only three geometrical forms (a square, a quarter circle and a full circle), yet it is able to produce twenty-seven different characters, as well as diacritics and punctuation. In addition, the system allows four variations of some glyphs and twelve different weights, showcasing a high degree of flexibility and the potential of the combination of simple forms (Lorenz, 2021).

For system-based visual identities, this approach is particularly popular as it allows variation while maintaining recognition (Lorenz, 2021; Lupton, 2011). For example, the identity for Walker Art Center, designed by Andrew Blauvelt and Chad Kloepfer, in 2005. Its system is based on a "kit of parts" methodology (Lupton, 2011). This design permits produce endless variations of the system by combining existing elements in varied forms; they can also add new patterns as needed, creating a living visual brand (Figure 14) (Blauvelt, 2011; Felsing, 2010; Lupton, 2011). Walker's visual identity is identifiable by its visual system rather than the logo (Felsing, 2010).

Figure 14.
Different applications
of the Walker Art Center
visual identity.

The other form of system is The Program. This concept was introduced by Karl Gerstner, in 1964 (Gerstner, 1964). The author proposed a logical-algorithmic framework to create visual solutions for graphic projects (Conrad, 2023, p. 12). Instead of designing a logo as a solution, he proposed designing a set of rules which would ensure harmony across different applications (Lorenz, 2021; Murdock, 2016). Gertsner seemed to have predicted computer programs by proposing the application of parametric principles for designing systems (Conrad, 2023; Lorenz, 2021).

Karl Gerstner was a pioneer in the domain of "language-like visual identities" as he believed "the design itself must take the place of the signet" (Felsing, 2010). That belief was showcased in some of his visual identity designs. For example, the identity for the record shop Boîte à Musique (Music box) from 1957 (Figure 15). It was based on a rectangular form responsive to different formats, thereby eliminating the need of a replicable logo (Hollington, 2011; Lorenz, 2021; Martins et al., 2019).

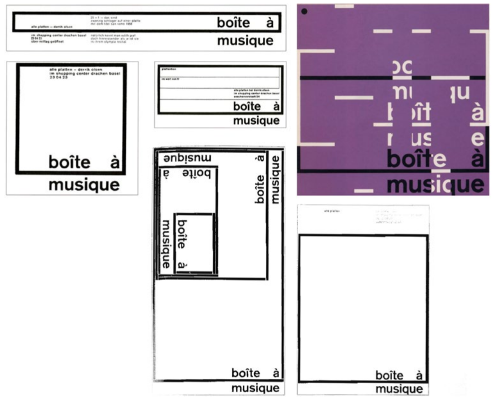


Figure 15.
Boîte à Musique visual identity system.

Lastly, the Tool refers to the instruments through which systems are generated. The outcomes are always conditioned by the tool that is used. These conditions can provide consistency to a system, and the design process itself can be the identifiable element of a visual identity (Lorenz, 2021). Using unconventional tools can not only stimulate new approaches and result in unique visual expressions as they can also shape the brand visual language (Lupton, 2011).

There have been system-based identities from the past, for example, the Olivetti's identity which was maintained by Gioanni Pintori from 1936 until 1967 (Meggs & Purvis, 2012). This identity was not logo-centric; the coherence and recognition was achieved not through a systematic design program but through the general visual appearance of promotional graphics (Figure 16) (Meggs & Purvis, 2012; Murdock, 2016). This approach could be considered as a dynamic and flexible visual system (Murdock, 2016). Other examples such as Paul Rand's identity for Columbus Indiana or Debora Deborah Sussman's identity for LA84 Olympics, based on a modular system (Meggs & Purvis, 2012; Murdock, 2016).

Figure 16.
Advertising posters from Olivetti., designed by Giovanni Pintori.

For this reason, the author Murdock (2016) believes this way of embracing visual identities is not just a trend but another paradigm apart from corporate and modern beliefs. However, it is safe to say that when everything had to be printed, the application of a flexible visual identity would be more difficult and expensive (Lorenz, 2017). Technology not only facilitated a more systematic approach to design as it also expects it (Llop, 2014; Lorenz, 2021). The following section addresses design practices driven by programming tools and how they can ease and enhance our work.

3.4
—
Programming Tools for Dynamic Visual Identities

This new millennium witnessed graphic designers begin to program themselves, or collaborate with programmers, to create tools for crafting their design artefacts (Conrad, 2023). As we have seen (section 3.1), dynamic visual identities have been employing programming tools for the development of their dynamic graphic elements. The use of these tools have been increasing in recent years in all graphic design domains (Leijsen et al., 2023; Reas & McWilliams, 2010).

Given the characteristics of system-based visual identities — rather than being a finite set of static elements with an instruction manual for their use, are flexible systems based on algorithms that establish constants and variables (Felsing, 2010) — using programming tools for their development proves to be highly relevant. The systematisation of the graphic language into parameters and variables provides designers control over the visual language and coherency across mediums (Llop, 2014).

For the development of these visual identities, the designer becomes the one who sets parameters and rules to generate content, rather than the one who creates them (Guida & Voltaggio, 2016; Lorenz, 2021), the same way Karl Gertsner (1964) has proposed in *Designing Programmes* (see section 3.3).

Furthermore, the use of programming tools can go beyond the limitations of proprietary software not only reaching higher degrees of complexity and innovative results (Guida & Voltaggio, 2016), as well as providing the opportunity to spend more time on experimentation (Reas & McWilliams, 2010).

3.4.1

—

Case Studies

In this section, we present examples that illustrate the premises exposed previously. These cases showcase the potential of programming tools for the development of coherent visual languages.

The selected examples are relevant to the scope of this dissertation due to their ability to produce diverse graphic materials for a single visual identity or when they do not answer to a specific identity, they showcase a variety of visual results with a high level of coherence.

To analyse these case studies was important to provide us with insights that contributed to the development of our system.

→ Puddle Builder, Andreas Gysin & Sidi Vanetti, 2010 — 2015

Andreas Gysin and Sidi Vanetti developed various tools in Processing for the design of the visual identity material for the music for The Puddle, a live electronic music and dj sets around Zürich, around 2010 to 2015 (Gysin, n.d.).

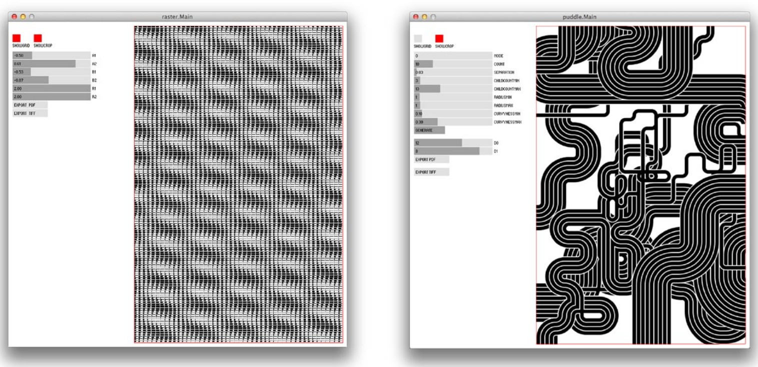


Figure 17. Pudlle Builders 10 & 13 respectively, developed in Processing,

They developed various "puddle builders" (Figure 17) for different editions and for different graphic artefacts. Different builders allowed the creation of different outcomes, yet since the identity system was rooted on a grid, the results maintained overall coherence (Figure 18). All visuals intended to iterate the idea of translating music into patterns (Benyon, 2018). The images produced by the builders would be used to print the graphic artefacts for the festival.

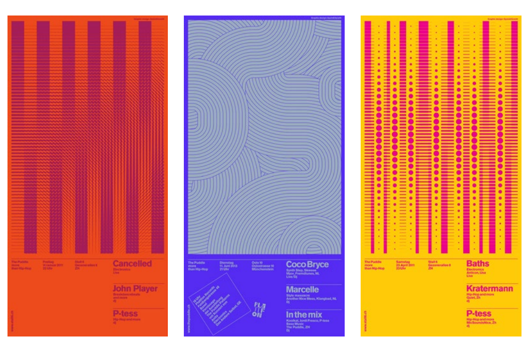


Figure 18. Posters for The Puddle Festival.

→ CIDDIC, André Burnier, 2018

The visual identity system built for CIDDIC (Center for Integration, Documentation and Cultural Diffusion – Unicamp), was developed by the Brazilian graphic designer and creative coder, André Burnier, in 2016 (Burnier, n.d.; Vieira, 2022). This identity was, originally, a non-dynamic system. However, due to the need of producing graphic pieces by non-designers and people from staff, in 2018, Burnier proposed a solution that automated the generation of layouts according to the already existing system's rules (Vieira, 2022).

The tool applies the identity's guidelines for the production of graphic artefacts allowing anyone to develop layouts compatible with the CIDDIC's visual identity (Burnier, n.d.), thus eliminating the need for a manual (Vieira, 2022).

The system (Figure 19) was developed in p5.js and it allows users to insert text, change colours and dimensions. Furthermore, it provides buttons to randomise colours or randomise different layouts for the same information. Files can be exported for digital devices or for printing (Burnier, n.d.). This way, individuals without a background in design can seamlessly produce coherent outcomes, even if they aren't familiar with the visual guidelines.

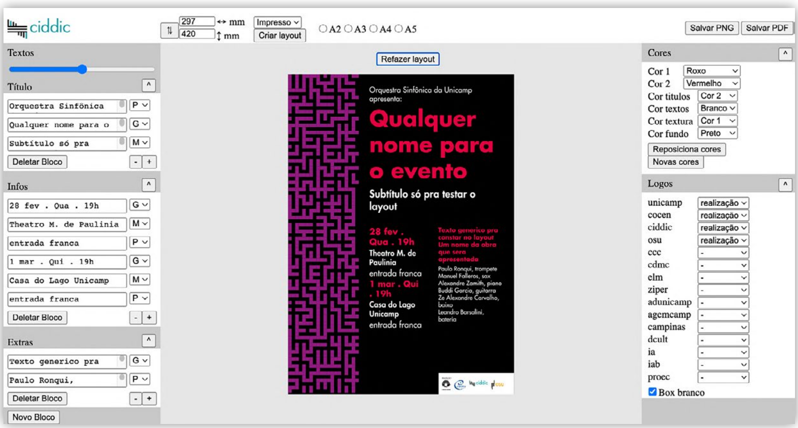


Figure 19. CIDDIC's tool for the development of different graphic materials.

→ Black Light, TwoPoints.Net, 2018

Black Light was an art exhibition presented in Barcelona, in 2018. For its visual identity, TwoPoints.Net developed a full scale visual system with custom lettering, patterns and flexible design elements based on a modular approach. Among the deliverables were signs, posters, banners, flyers, notebooks, tote bags, tickets and buses (*Black Light*, n.d.).

An online tool was developed later, by Tim Rodenbröker, that allowed users to create their own graphics based on the visual identity guidelines. The system (Figure 20) was developed on Processing and allowed the manipulation of a set of different graphic elements and the manipulation of the layout grid. Furthermore, it allowed randomising the design as well as exporting the results in PNG or PDF (*Black Light*, n.d.).

The tool allowed spreading the word about the exhibition as users could share their generated content on social media with the use of hashtags (*Black Light*, n.d.). This tool showcases the potential of programming for the development of coherent visual languages while also exemplifying how these tools can enable the democratisation of design, as it empowers users to manipulate and engage with Black Light's visual identity.

Figure 20.
Black Light's tool developed by Tim Rodenbröker.

→ Analog Algorithm, Christoph Grünberger, 2019

Although this example is not a tool in itself, we found it relevant as a case study as it is an algorithmic-based toolkit to develop new forms, fonts, logos and patterns (*AnalogAlgorithm,* 2019). The visual experiments showcased throughout the book can easily be identified as collections of visual languages (Figure 21) .

This happens because the outcomes are based on a systematic approach — a design process in which individual decisions follow much larger and deeper principles than immediate and spontaneous-intuitive actions (*AnalogAlgorithm*, 2019). The book aims to introduce computational techniques and algorithm methods for the development of graphic content.

This way of developing design artefacts is, as we have seen, very much related to the way how visual identity systems are developed — based on a set of parametric rules.

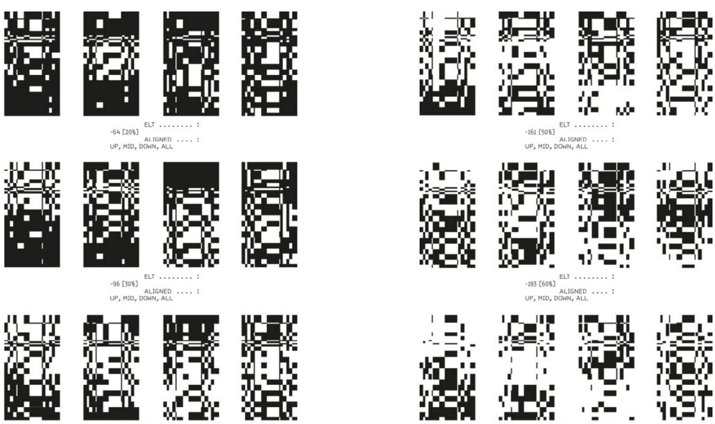


Figure 21.
Grid-based patterns from the book *Analog Algorithm.*

→ SPACE10, DIA Studio, 2020

The brand for SPACE10, a well-known research and design Lab, is based on ever-evolving concepts and flexibility as it expects to allow the brand to constantly represent the diversity of projects they handle (Long, 2020). DIA studio developed a generative tool for SPACE10's visual identity (Figure 22) (*SPACE10*, 2020), which aims to "encourage experimentation and creativity" (Long, 2020).

The system allows the entry of text and its manipulation such as font-size and line height. Moreover, it allows playing with camera position and zoom as well as handling full layouts, images, and real-time adjustments (*SPACE10*, 2020). The system permits various forms of motion and typographic manipulation.

This application allows anyone to operate with kinetic typography and develop adequate content according to the guidelines of SPACE10's visual identity. With this tool, everyone within SPACE10 is able to use the tools, not just co-workers with a design background (Long, 2020; *SPACE10 Identity*, n.d.).

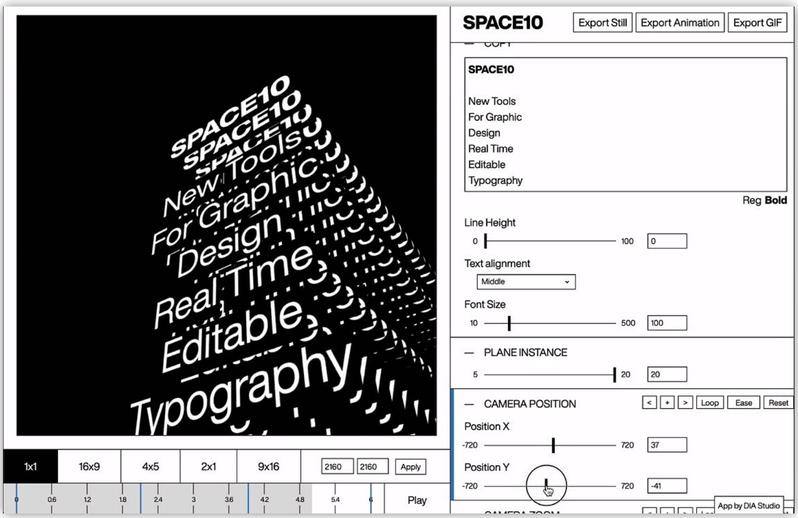


Figure 22. SPACE10 generative app developed by DIA Studio.

→ Munken Creator, Arctic Paper, Juno & Patrik Hübner, 2022

The Munken Creator (Figure 23) is a web-based tool that allows users to create visuals using Munken Sans, a font recently launched by the paper manufacturer Arctic Paper (Gorny, 2022; *Munken Creator*, 2022).

The tool offers various possibilities for crafting typographic explorations such as font size, rotation, varying colour, scale and spacing. Also, it allows uploading background images and providing animation to the typographic elements. These elements are set in an invisible grid that can be arranged in different manners. The final outputs can then be saved as images or videos in WebM and MP4 formats, and used for print or web projects (Gorny, 2022).

While this tool is not designed to address any visual identity requirement, the outcomes can still be perceived as a dynamic

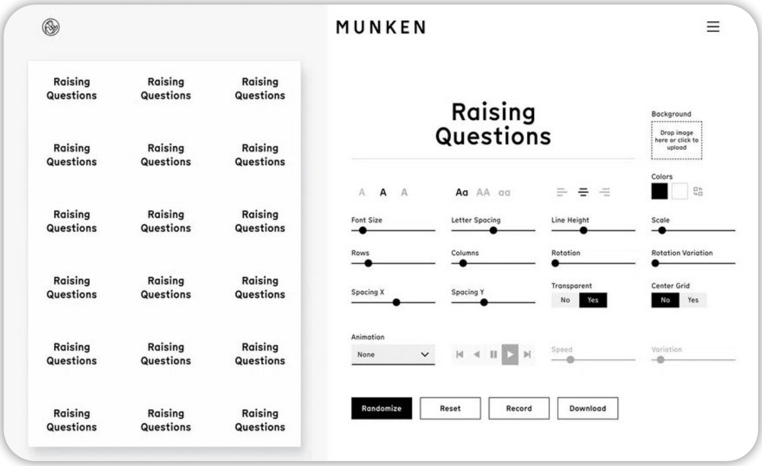


Figure 23.
The Munken Creator tool.

visual language. This is because the tool uses the same typography and employs a grid-based structure, allowing some degree of coherence despite its infinite possibilities.

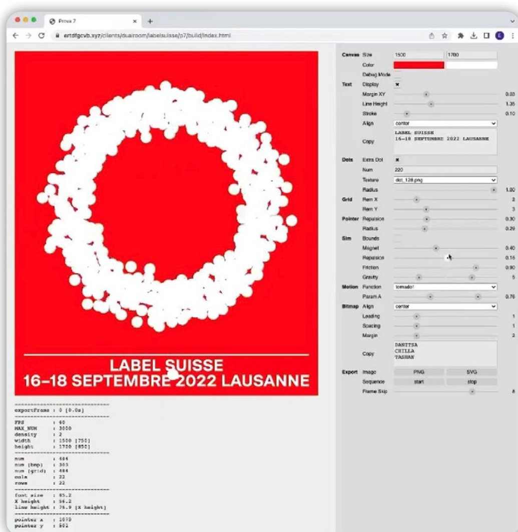
Munken Creator is another example of how these tools allow the democratisation of design, as it is online and accessible to anyone, and hopes to inspire people "to think beyond the boundaries of how we communicate" (Gorny, 2022).

→ Label Suisse Festival, Dual Room & Andreas Gysin, 2022

In 2022, the studio Dual Room developed a visual identity for a music festival in Lausanne, Switzerland. In collaboration with Andreas Gysin they developed a machine (Figure 24) that allowed the production of various graphic content aligned with the visual identity guidelines. The designer justifies the need of a machine due to the amount of images to be produced (Dual Room, 2023).

This tool allowed the manipulation of canvas, providing the possibility to define various formats, the manipulation of text, dots, colours and the grid that define the visual identity (Dual Room, 2023). The dots are the main graphic element of the visual identity that simulate the crowd movements (Label Suisse Festival, 2023) and can work as a pattern or as typography. The machine allows the exportation of still artefacts for printing matters as well as videos and sequences of the animations developed with the tool (Dual Room, 2023).

Figure 24.
Label Suisse Festival tool, developed by Andreas Gysin.



3.5 — Conclusion

The increasing prevalence and relevance of computational approaches in design are undeniable. Regarding the domain of Dynamic Visual Identities, the use of programming tools for developing visual identities is seen as increasingly prominent. Yet, our analysis of the state of the art has revealed that there are opportunities for further exploration and understanding of this field.

Apart from not existing a general agreement on the domain, it was also noticeable that, despite the evolution of visual identities towards dynamism, the majority of them still exhibit a strong focus on the graphic mark. Furthermore, it was noticed that the designers who are developing computational tools for the development of these visual identities are pioneering design studios or creative coders; whereas there is a visible conventional approach to design being taught in many graphic design schools (Leijsen et al., 2023; Lorenz, 2021).

This system seeks to address this issue by providing a democratic tool for those who want to explore and understand better system-based visual identities. Moreover, it aims to showcase the potential of computational tools in the development of such visual identities.

Overall, the conducted investigation provided us with a great understanding of dynamic visual identities, which has helped us make reasoned decisions throughout the project development and ensure its relevance within the graphic design contemporary scene.

4.

Perceiving Dynamism: An Analysis of DVIs

For a better comprehension of system-based visual identities, we analysed a sample of 40 visual identities of this type. We analysed which variation mechanisms (Martins et al., 2019) were employed and which graphic elements were dynamic in the visual system. The sample of visual identities can be accessed in the appendix .1

This analysis served to better understand different types of variation, their outcomes and characteristics. We aimed to gain knowledge on these mechanisms and their possibilities for implementing them in the future. Additionally, we aimed to identify the graphic elements that can be dynamic within a system, in order to gain insights into the type and amount of elements our system would accommodate.

4.1 — Variation Mechanisms

To analyse the processes that give dynamism to a visual identity, we have grounded the analysis in the research proposed by Martins et al. (2019). As it was seen (see section 3.2), these authors pointed out eight variation mechanisms that can be identifiable in dynamic visual identities. They described each as the following:

Colour variation:
there is a graphic element that changes in colour.

Combination:
there is a combination of different graphic elements that belong to the VI system.

Content variation:
there is an area or space where different imagery is placed, be it in the background or foreground.

Positioning:
there is a graphic element that is positioned in different ways.

Repetition:
there is a repetition of the same graphic element.

Rotation:
there is a graphic element that is rotated.

Scaling:
there is a graphic element that changes in size.

Shape transformation:
there is a graphic element that changes in shape.

(Martins et al., 2019)

Their study had already examined the variation mechanism of 24 visual identities based on the system. For the aim of this project, we believed it was important to analyse a bigger sample. This decision was motivated by several factors: firstly, their study does not focus only on system-based visual identities but all identities that can be considered as dynamic, including dynamic logos; secondly, their study only covers visual identities up to the year 2013.

We wanted a larger sample to make a more extensive study on this matter and also to analyse identities from recent years in order to gain insights into recent developments. We added 16 visual identities to the sample to achieve 40 in total.

We started by analyse the employed variation mechanisms (Table 1) and their combinations.

We analysed the frequency of each variation mechanism. In this sample, it is evident that the most frequently used mechanisms are combination and repetition. Following are colour variation and content variation, respectively. The less used mechanisms are position, rotation, scale and shape transformation.

→ Variation mechanisms frequency

A. Colour Variation: 12;
B. Combination: 18;
C. Content Variation: 10;
D. Position: 8;
E. Repetition: 18;
F. Rotation: 8;
G. Scale: 8;
H. Shape Transformation: 8.

When examining the combination of mechanisms visual, we can notice that these examples only employ up to a combination of four variation mechanisms. There are 11 cases of visual identities that employ a single mechanism (11/40), 15 cases that employ a combination of two (15/40), 7 cases that employ three (7/40) and 7 that use four mechanisms (7/40).

→ Combination of mechanisms

1 mechanism: 11;
2 mechanisms: 15;
3 mechanisms: 7;
4 mechanisms: 7.

Table 1.
Analysis of variation mechanisms.

		A	B	C	D	E	F	G	H	
1.	Boîte à Musique	•	•	•	•	•	•	•	x	1
2.	Columbus, Indiana	x	•	•	x	x	x	•	•	4
3.	Walker Art Center	x	x	•	•	x	•	x	•	4
4.	Lovebytes	•	•	•	•	•	•	•	x	1
5.	IDTV	x	x	•	•	x	•	x	•	4
6.	Mobile Media Lab	•	x	•	•	x	•	•	•	2
7.	Boolab	•	•	•	x	•	x	•	•	2
8.	Paramount	•	•	•	•	x	•	x	•	2
9.	AGI Open Porto	•	x	x	•	x	•	x	•	4
10.	Prima Vina Stiriae Slovenae	•	•	x	•	•	•	•	•	1
11.	EDP	•	x	•	•	•	•	•	•	1
12.	OVG State	•	•	•	•	•	•	•	x	1
13.	São João Porto	x	x	•	•	x	•	•	•	3
14.	Axis of Culture in Katowice	•	x	•	•	x	•	•	•	2
15.	Catalan Wines	x	x	x	•	•	•	•	•	3
16.	Festival Caminhos Cinema Português	•	x	•	•	x	x	x	•	4
17.	Schism	•	•	x	•	•	•	•	•	1
18.	Sofia Breathes	•	x	•	•	x	•	•	•	2
19.	Jewish Museum & Tolerance Center	•	•	x	•	•	•	•	•	1
20.	FLUX	•	x	•	•	x	•	•	x	3
21.	The Floating Eye	x	x	•	•	•	•	•	•	2
22.	KAKAO	•	x	•	•	•	•	•	•	1
23.	WOW	x	•	x	•	•	•	•	•	2
24.	NAA TAA	•	•	•	x	•	•	•	•	1
25.	Dumbar Design College	x	•	•	•	x	•	•	•	2
26.	Whitney Museum of American Art	•	•	•	•	•	•	•	x	1
27.	Porto	•	x	•	•	x	•	•	•	2
28.	Medialab Prado	•	x	•	•	x	x	•	•	3
29.	Black Ligth	•	•	•	•	x	x	•	•	2
30.	Et Cetera Festival	•	•	x	•	•	•	•	•	1
31.	Caracol	x	x	•	•	•	•	•	•	2
32.	6th Guangzhou Triennial	•	•	•	x	x	x	•	•	3
33.	Impira	x	•	•	•	•	•	•	x	2
34.	Manataco	•	•	•	x	•	x	•	•	2
35.	Tribeca Festival	•	x	•	x	•	•	x	•	3
36.	Littlemore	•	•	x	x	•	•	•	x	3
37.	Chaumont Biennale	•	x	•	x	x	•	x	•	4
38.	Baehl	•	•	x	•	x	x	x	•	4
39.	NN North Sea Jazz	x	•	•	•	•	•	•	x	2
40.	Mercat de Música Viva de Vic	x	•	x	•	•	•	•	•	2

A colour variation B combination C content variation D position
E repetition F rotation G scale H shape transformation

In this sample, the application of a single mechanism happens more times than expected. Furthermore, the combination of only two mechanisms (15/40) is more prominent than the combination of three or four mechanisms (14/40). This seemed unlikely, as it was expectable that employing a greater number of variation mechanisms would result in higher levels of dynamism. We analysed, for each of these groups, what were the most commonly used variation mechanisms.

→ 1 mechanism

Shape Transformation: 4;
Content Variation: 4;
Combination: 2;
Position: 1.

Shape transformation can be manifested in various ways, making it a versatile variation mechanism. We believe its use alone can be understood due to this versatility. Content variation can also exhibit versatility, as the content can take on various forms, such as images, shapes or illustrations. When it is employed solely, usually this content is visually coherent in order to be used as a unifying component of all graphic materials, for example the identity for the Jewish Museum & Tolerance Center (19.).

Combination is employed alone when the identity system is based on an extensive set of graphic elements which are then combined in different ways to produce different messages, for example, the identities for EDP (11.) and for KAKAO (22.).

→ Combination of 2 mechanisms

Combination + Repetition: 5;
Position + Rotation: 2;
Colour Variation + Combination: 2;
Colour Variation + Shape Transformation: 2;
Colour Variation + Content Variation: 1;
Colour Variation + Repetition: 1;
Repetition + Scale: 1;
Repetition + Rotation: 1.

When employing only two mechanisms, the most used combination is combination and repetition (5/15). Furthermore, individually, the most used are repetition (8/15), combination (7/15) and colour variation (6/15) indicating their potentiality to produce dynamism. Rotation is used in 3 cases, 2 of them being combined with position. Shape transformation appears twice and combined with colour variation. Scale and content variation are used a single time and are combined with repetition and colour variation.

→ Combination of 3 mechanisms

Colour Variation + Combination + Repetition: 1;
Colour Variation + Combination + Content Variation: 1;
Combination + Repetition + Shape Transformation: 1;
Combination + Repetition + Rotation: 1;
Combination + Position + Scale: 1;
Position + Repetition + Rotation: 1;
Content Variation + Position + Shape Transformation: 1.

Although there are no matching trios in this sample, it is noticeable that repetition and combination are, once again, a prominent combination (3/7). Individually, we count combination (5/7), repetition (4/7), position (3/7), colour variation (2/7), content variation (2/7), rotation (2/7), shape transformation (2/7) and scale (1/7).

→ Combination of 4 mechanisms

Colour Variation + Combination + Repetition + Scale: 2;
Colour Variation + Position + Repetition + Rotation: 1;
Combination + Content Variation + Repetition + Scale: 1;
Combination + Repetition + Rotation + Scale: 1;
Combination + Position + Repetition + Scale: 1;
Content Variation + Repetition + Rotation + Scale: 1.

Interestingly, for all visual identities that employ a combination of four variation mechanisms was found repetition (7/7). Furthermore, it was found again a prominent use of repetition with combination (5/7). The combination Colour Variation + Combination + Repetition + Scale was found twice, in the Walker Art Center and IDTV visual identities. Individually the mechanisms were scale (6/7), combination (5/7), colour variation (3/7), content variation (2/7), position (2/7), rotation (2/7), shape transformation (0/7). Scale has a prominent use here, unlike previous combinations we analysed. This can suggest that the application of scaling may produce more dynamism when combined with other mechanisms.

Furthermore we analysed the combination in pairs, regardless if mechanisms are combined with others (Table 1.2). This has shown, as it had already been noticed by Martins et al. (2019), that

Table 1.2. Pairing of variation mechanisms.

	A	B	C	D	E	F	G	H
A. Colour Variation	•	•	•	•	•	•	•	•
B. Combination	6	•	•	•	•	•	•	•
C. Content Variation	2	2	•	•	•	•	•	•
D. Position	1	2	1	•	•	•	•	•
E. Repetition	5	13	2	3	•	•	•	•
F. Rotation	1	2	1	5	6	•	•	•
G. Scale	2	6	2	2	7	2	•	•
H. Shape Transformation	2	1	1	1	1	0	0	•

repetition and combination are the most used pairing (13/40). Following are the combinations of repetition with scale (7/40), repetition with rotation (6/40), combination with scale (6/40) and combination with colour variation (6/40).

As evident from the table, repetition is the mechanism that appears most frequently in combination with other mechanisms. This observation suggests that repetition may require the presence of other mechanisms to reach its full potential. Furthermore, we have seen previously that even though repetition is one of the most used variation mechanisms, there are no examples of visual identities that use repetition as a single mechanism to achieve dynamism.

The mechanism combination also stands out as being frequently combined with other mechanisms. Unlike repetition, this mechanism is used twice as a single mechanism to achieve dynamism. However, this is a minor instance given its usage of 18 times. These observations suggest that, even though combination and repetition are the most frequently used variation mechanisms they require to be paired with other mechanisms to be fully effective, most times, combined with each other.

Regarding position, rotation, and scale, it seems that these types of variations need to be paired with other mechanisms to achieve dynamism. In this sample, they can be seen as enhancers of dynamism, yet individually, they do not seem to be capable of displaying the potential to make dynamic visual identities.

Shape transformation does not pair with rotation or with scaling in the sample. Moreover, it is the mechanism that is less frequently combined with other mechanisms. This observation indicates, as it was stated before, the versatility of this mechanism, as it does not need pairing for achieving dynamism. The same is possible to point out for content variation.

4.2
—
Dynamic Components

Apart from the variation mechanisms analysis, we used the same sample to examine in which of the visual system components were the mechanisms employed (Table 2). We aimed to understand which graphic elements were most commonly used to convey dynamism and in which ways mechanisms affected these elements.

We changed our approach to define if the dynamism was based on A. a single graphic element, B. various graphic elements, C. typography, D. imagery, E. other. We found it more relevant to focus on understanding the dynamic potential of individual elements or combinations of elements rather than categorising them based on their formal nature. This shift was aligned with our decision to

Table 2.
Analysis of dynamic components.

		A	B	C	D	E	
1.	Boîte à Musique	x	•	•	•	•	1
2.	Columbus, Indiana	x	•	•	•	•	1
3.	Walker Art Center	•	x	x	•	•	2
4.	Lovebytes	•	•	•	x	•	1
5.	IDTV	•	x	•	•	•	1
6.	Mobile Media Lab	•	x	•	•	•	1
7.	Boolab	•	x	•	•	•	1
8.	Paramount	•	x	•	•	•	1
9.	AGI Open Porto	•	x	•	•	•	1
10.	Prima Vina Stiriae Slovenae	•	•	•	x	•	1
11.	EDP	•	x	•	•	•	1
12.	OVG State	•	•	•	x	•	1
13.	São João Porto	•	x	x	•	•	2
14.	Axis of Culture in Katowice	x	•	•	•	•	1
15.	Catalan Wines	•	x	•	•	•	1
16.	Festival Caminhos Cinema Português	•	•	x	•	•	1
17.	Schism	•	•	•	x	•	1
18.	Sofia Breathes	•	•	x	•	•	1
19.	Jewish Museum & Tolerance Center	•	•	•	x	•	1
20.	FLUX	x	•	x	•	•	2
21.	The Floating Eye	•	x	•	•	•	1
22.	KAKAO	•	x	•	•	•	1
23.	WOW	•	•	•	x	•	1
24.	NAA TAA	•	•	x	•	•	1
25.	Dumbar Design College	•	•	x	•	•	1
26.	Whitney Museum of American Art	x	•	•	•	•	1
27.	Porto	•	x	•	•	•	1
28.	Medialab Prado	•	x	•	•	•	1
29.	Black Ligth	x	•	•	•	•	1
30.	Et Cetera Festival	•	•	•	x	•	1
31.	Caracol	•	x	•	•	•	1
32.	6th Guangzhou Triennial	x	•	•	•	•	1
33.	Impira	x	•	•	•	•	1
34.	Manataco	•	•	x	•	•	1
35.	Tribeca Festival	•	•	x	x	•	2
36.	Littlemore	x	•	x	x	•	3
37.	Chaumont Biennale	•	•	x	x	•	2
38.	Baehl	x	•	•	x	•	2
39.	NN North Sea Jazz	•	•	x	•	•	1
40.	Mercat de Música Viva de Vic	•	•	•	x	•	1

A single graphic element B various graphic elements
C typography D imagery E other

change the path of the system development from giving a limited set of geometrical forms to allowing users to upload their own elements (see section 5.2 & appendix 2). Analysing visual identities in this manner was important to gain insights on the possibilities we wanted to give our system. This is how we defined each component:

A single graphic element:
one single element makes the dynamism of the system

Various graphic elements:
variation mechanisms are applied to various graphic elements that usually share the same visual forms, ensuring their coherence.

Typography:
dynamism is applied in typography.

Imagery:
imagery is used as a dynamic component of the system.

Other:
variation is applied to other component of the system.

We identify various graphic elements as being one component of the system as they usually belong to the same visual form and maintain coherence. This is aligned with how Nes (2012) has defined graphic elements as a part of an identity system (see section 3.2). For this analysis we distinguish one graphic element from various because we wanted to analyse which was the most common and if having more graphic elements can achieve more dynamism.

In this sample it is noticeable that most visual identities rely their dynamism on a single component of the system (this includes the B. various graphic elements) (33/40). Only 6 of them apply variation to more components of the system, 2 of them employ variation on various graphic elements and typography, 1 on a single graphic element and typography, 2 on typography and imagery. Only one visual identity was found to apply dynamism in 3 components, the visual identity for Littlemore (36.). This identity applies variation to one single graphic element, a shape of an L, to typography, that changes its tracking according to the L transformations, and to imagery because it employs content variation.

→ Dynamic components frequency

A. A single graphic element: 9;
B. Various graphic elements: 15;
C. Typography: 12;
D. Imagery: 12;
E. Other: ⌀.

As it is seen, most employ variation to a set of various graphic elements (15/40), following is most seen variation on typography (12/40) and imagery (12/40), in last is variation applied to one single element (9/40). We did not find any other possible element to apply variation. However, this analysis is only referent to the visual domain. Imagery being dynamic usually has to do with the usage of content variation (10/40). However, for example, the visual identity for the Chaumont Biennale (37.) uses images as a component of the system and not as a means for variation.

The analysis of dynamic components gave us insights on which possibilities we wanted to implement in our system, which ended up being the possibility of applying variation to diverse graphic elements, uploaded by users .

4.3 — Conclusion

We acknowledge that this analysis remains open to different interpretations. Furthermore, there is space for further investigation regarding the relationship between used variation mechanisms and dynamic components. Nevertheless, we found the developed analysis to be adequate to perceive the performance of variation mechanisms and the possibilities of their combinations, as well as which kind of components can employ dynamism within a system and how they achieve it.

Some other remarks were noticed. For instance, we found, during our analyses, that most visual identities have a graphic mark or an element that serves a similar function, unlike some authors have argued (see section 3). We acknowledge that this sample is not representative of the entire domain; nonetheless it was worth noting this aspect as one might argue that system-based visual identities renounce graphic marks.

Furthermore, we have encountered different relationships between the identity's visual language and the element that is used to represent it. We have noticed that this relationship can be: (i) the visual language defines the graphic mark, (ii) the graphic mark defines the visual language or (iii) the visual language and the graphic mark are distinct. However, we did not extend this analysis as the main goal of this project was to focus on visual systems.

Overall, this analysis was important to perceive the complexity and characteristics of such visual identities. The knowledge acquired was fundamental to guide the decisions and development of the project.

5.

—

Project

↘

The project intention was to work within the scope of system-based visual identities as perceived in the previous section (see section 4). Moreover, its conceptualisation was built upon the study of Martins et al. (2019).

As seen (see section 1.3) , the project goal is to develop an online tool for users to freely interact with, aiming that by doing it, they can further understand system-based visual identities and their characteristics. Our intention is that the tool can provide users with insights to later develop their own visual languages, as well as notions on the potential of computational tools for the development of such visual systems.

The tool aims to provide two possibilities: one of free experimentation, which can contribute to novel possibilities for the development of new designs and visual identities and another to develop materials that can answer the needs of a specific design request. It is important to acknowledge that the project does not intend to generate complete visual identities, but to provide knowledge on how to develop coherent visual languages and, if possible, to produce novel visual outputs.

Although it might be possible to apply the results in a real-world context, that is not the tool's purpose given the complexity of visual identities. By enhancing their scope of creative resources, users can then develop based on the drafts produced by the tool. Above all, it is expected that users can apply the knowledge obtained to their design projects.

Prior to the development of the tool developed in the context of the present dissertation, a preliminary version was designed to explore these concepts. This initial iteration allowed us to gain pertinent insights into the potential of a tool for this matter. Subsequently, based on the insights we gained, we further developed a more robust and solid tool.

5.1 — Preliminary Project

In the context of the course of *Computational Design Laboratory*, from the Mater's Degree, we have developed a computational tool that aligns with the research and proposal of this dissertation. The project was named *Understanding Language-like Identities: Towards Visual Languages* in reference to the expectations we hoped to achieve.

The development of this tool allowed us to conduct initial experiments on variation mechanisms (Martins et al., 2019), which are the variation processes employed in the project developed within the context of this dissertation. This preliminary tool provided us with beneficial conclusions on the use of a parametric tool for the development of coherent visual results.

5.1.1
—
The Tool

The tool allows application of some variation mechanisms — colour variation, positioning, repetition and shape transformation — to an initial shape — a square. The interface (Figure 25) is divided into 4 modules: canvas, variation mechanisms, canvas, options, and final options.

Figure 25. Interface of the preliminary tool.

→ Canvas
The canvas is where visual results are displayed. To select elements on the canvas, users click above the desired shape. If it is selected it gets a green stroke. The selected element is the one that will suffer variations. To unselect an element, the users click again above it or above the canvas.

→ Variation Mechanisms
Variation mechanisms are the processes that provide dynamism to a design system (Martins et al., 2019), as already seen (see sections 3.2 & 4.1). This tool enables users to play with: (i) colour variation — changes the colour of a selected element; (ii) positioning — changes the position of a selected element; (iii) repetition — repeats elements; and (iv) shape transformation — changes the width and the height individually of a selected element. Users apply these mechanisms through the use of sliders, which enables a faster interaction and change of elements. To change colour, the system provides a colour picker (Figure 26).

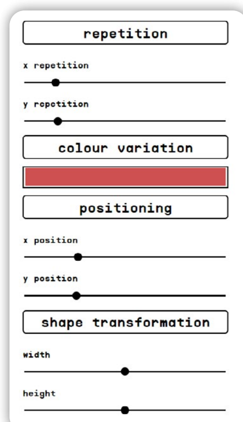


Figure 26. Variation mechanisms interaction with sliders.

→ Canvas Options
Canvas options are "background colour" and "format". These buttons allow changing the colour of the background through a colour picker and changing the width and height of the canvas. The "format" feature allows users to export different visuals for different graphic artefacts. For example, users can define a format of a social network profile cover or a format of a poster. Any rectangular format is achievable by interacting with sliders to change the size of the canvas.

→ Final Options
Under the canvas are the final options. When the drawing is done, users can export or clear the canvas.

5.1.2
—
Results

This preliminary tool enabled the production of coherent visual outcomes (Figure 27 & Figure 28). These results can be perceived as belonging to the same visual family. They were applied in mockups to better understand their potential in the development of language-like identities (Figure 29 & Figure 30).

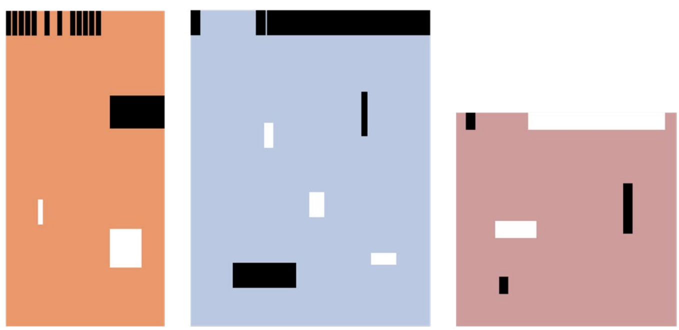


Figure 27. Three visual outputs that maintain coherence.

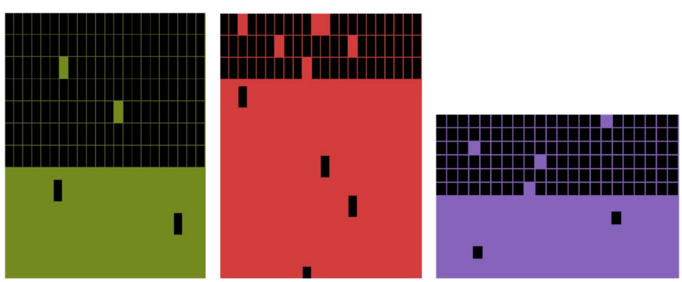


Figure 28. Three visual outputs that maintain coherence.

Figure 29.
Application of the visual outputs of Figure 27.

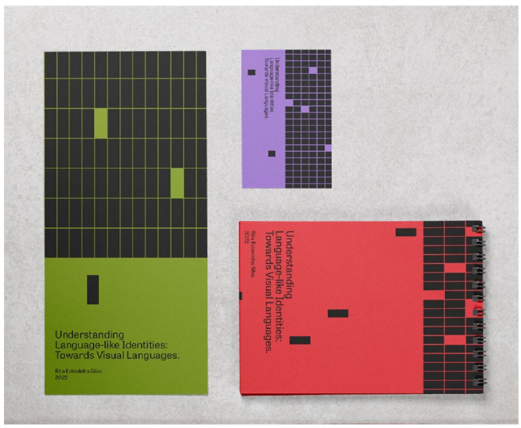


Figure 30.
Application of the visual outputs of Figure 28.

5.1.3
—
Conclusions

The examples show that the combination of the same set of mechanisms can achieve diverse results. Therefore, we acknowledge that the more possibilities the system allows, the more varied and novel results will be.

Although the system did not commit to generate complete visual identities, the possibility of creating coherent visual outputs enables the development of language-like systems. However, to use the tool for that purpose, users need to understand the fundamentals of such identities or they will not be able to get appropriate results. This happens due to the exploratory character of the system, which allows users to create anything without cohesion. We expect that this issue is tackled by the implementation of rules when developing outcomes, which is something we address in the project developed in the context of this dissertation (see section 5.2.2 — 3.2).

Besides its formal mistakes and implementation limitations, the tool shows great potential for the creation of coherent visual results. It is perceptible how the application of only four variation mechanisms already offers a wide range of outputs. Considering this project as an initial exploration for a more exhaustive research — the present research — and by evaluating the results provided by the tool, we can prospect that a more complete system can be an advantage for designers and users that want to learn and develop language-like identities.

5.2 — DynamicIDioms: Towards Visual Languages

DynamicIDioms: Towards Visual Languages was developed for the manipulation of graphic elements that can provide dynamism to a system-based visual identity. Following the same principles proposed by Martin Lorenz (2021) in his book *Flexible Visual Systems* (see section 3.2), we aim that this project can work as a tool kit for the development of visual systems. We intend that, by playing with the tool, users become familiar with processes and mechanisms to develop their own systems. Moreover, we hope to showcase the potential of programming tools for easing design processes while maintaining coherent outcomes.

For this matter, we decided to implement an online tool to allow easy access by the general public. Given the increasing relevance of these concepts in the contemporary world (Lorenz, 2021), we perceive our tool as a valuable resource for designers. Moreover, as stated by some authors (Leijsen et al., 2023; Lorenz, 2021; Murdock, 2016) many educational programs around the world are still based on traditional methods of developing design. By making this tool easily accessible and by democratising design, we hope it can unlock new creative possibilities.

5.2.1 — Conceptualisation

As seen in the previous analysis (see section 4), dynamic visual identities can be composed of one or more dynamic components, not including the graphic mark. The dynamism can be applied to one single element, a group of graphic elements, typography or images.

Therefore, we wanted our system to allow the input of multiple and varied graphic elements, according to users' needs.

To achieve the dynamism in these elements we decided to build our system upon different variation mechanisms proposed by Martins et al. (2019), as seen before these are: (i) Colour variation, (ii) Combination, (iii) Content variation, (iv) Positioning, (v) Repetition, (vi) Rotation, (vii) Scaling and (viii) Shape transformation (see section 3.2 & 4.1).

To produce a higher range of possibilities, we thought users would be able to change and interact with the values of these mechanisms. For example, if applying the rotation mechanism, users might change the rotation degrees of a visual component, or even, the centre from which the rotation is applied. Following the same approach as many of the generative tools showcased previously (see section 3.4.1), we considered the implementation of sliders for users to easily play and change parameters values.

These mechanisms solely do not guarantee the coherence that visual identities expect. In language-like visual identities, new combinations of elements repeatedly occur on the basis of a "grammar" (Felsing, 2010). This grammar is what ensures that the overall design system is set based on the same rules.

From the beginning, our intention was to implement a "grammar" in our system that could chain the actions that the user is making. For example, if a user chooses to apply rotation to a shape, and then repetition is applied, the system will memorise these actions. Next, if the user decides to play with the value of rotation degrees, the system will rotate the shape and "remember" to repeat it. We aimed that our tool could chain variation mechanisms into a sequence that would work as the regulator of the visual results that are being outputted. Different outputs and variations are then achieved by playing with the values of variation mechanisms, ordered inside the system grammar.

In regard to the interface, we conceived a canvas that displayed the visual results, a section for selecting variation mechanisms, a section of the system grammar, and another for changing the canvas format — given that we are working within the branding domain, we intend to allow the system to adapt and generate different formats. The grammar displays the variation mechanisms used; the values of variation mechanisms can be accessed when they are already displayed inside the grammar. After a user applies a mechanism, it will immediately be chained into the grammar and it will apply the process according to the previously chosen mechanisms.

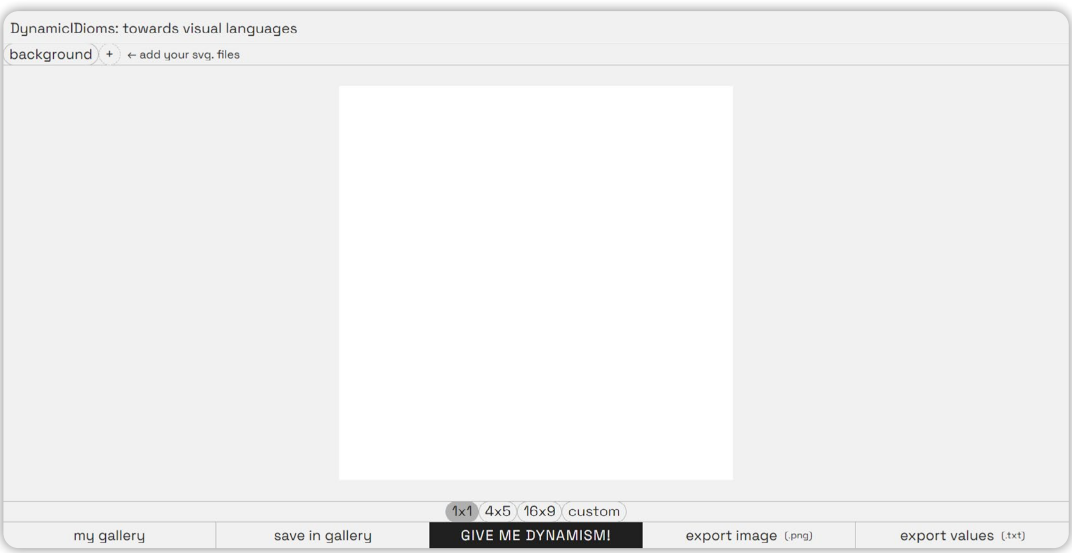
These identified functionalities — allowing different dynamic components; exploring variation mechanisms; and addressing the concepts of a grammar — were used as requirements to be considered throughout the entire development process. Even though the main idea for the tool remained constant during the development of the project, we encountered some technical challenges and difficulties that required that alternatives be found to implement a working solution. Yet, we remained firm to the development of these initial concepts and functionalities. In the following section we showcase the tool, its development and potentialities.

5.2.2
—
The Tool

DynamicIDioms was developed in web development languages: HTML, CSS and JavaScript. When beginning this project I had no familiarity with these web languages. This was translated in the encounter of some limitations and dificulties throughout the development process that delayed our project. However, the decision to develop a tool for the web was to allow the most users to freely experiment and play with it. Additionally, it provides a greater project dissemination as it can be made available online and accessible on any computer, unlike other programming tools. Unfortunately, given the nature of the tool and personal programming limitations, it does not operate in smaller devices which is a limitation that we aim to address in the future.

The tool interface (Figure 31) is divided into different units that correspond to different possibilities and actions. These units are: (1.) Graphic Elements — includes the background and allows the upload of graphic elements to be inline within this section; (2.) Style and Compose — allows setting the styling and composition of graphic elements — within each graphic element section; (3.) Dynamism — allows setting variation mechanisms to each graphic ele-

Figure 31.
DynamicIDioms tool interface.



ment; (4.) Formats — allows setting the format of the canvas; (5.) Give me Dynamism — a button that applies the dynamism to all components at the same, enabling the output of diverse visual results; (6.) Gallery — allows saving different outputs and displaying them together; and (7.) Export — allows exporting the results as images and the values of the chosen variation mechanisms for each graphic element. The places in the interface for each of these sections can be seen in Figure 32, according to this sequencing.

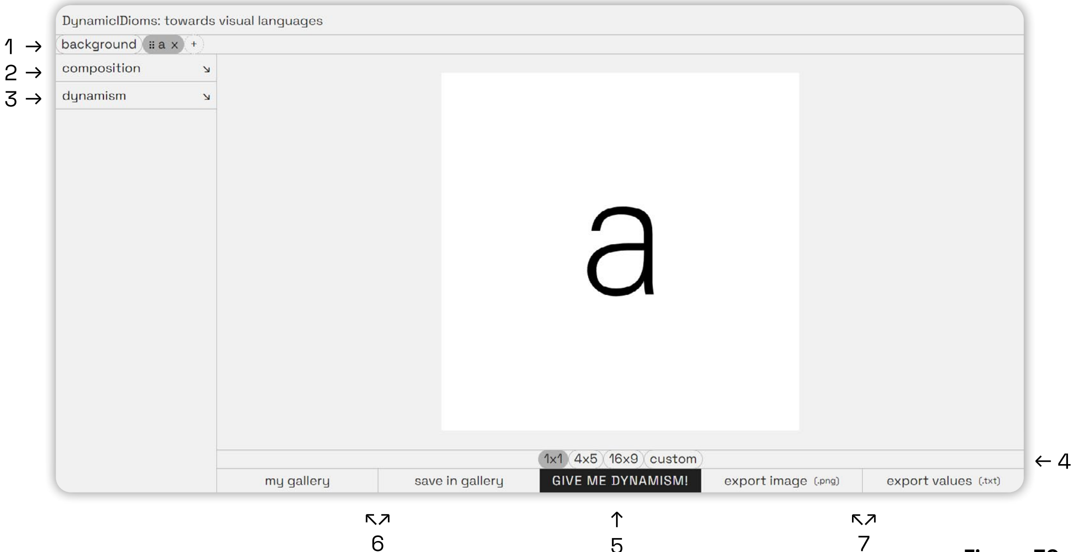


Figure 32. Interaface sections numbered.

In the next paragraphs, we provide detailed explanations of the tool's functionalities and its development process, following these interaction sections. Regarding the development process, it is possible to consult our appendix 2 to get a better perception on the evolution of the present system.

1. Graphic Elements

→ Programming

We have seen that dynamic identities can base their dynamism on different visual components of the identity system (see section 4.2). Therefore, it was relevant that our tool would permit this possibility, translating the reality of contemporary identities and enhancing the creative possibilities of our users. With this said, the first phase of interaction corresponds to the upload of graphic elements from the user's computer. Uploading personal graphic elements gives users the possibility to play with their own visuals and develop outcomes according to their own aesthetics and style. These files are the visual elements chosen by users to develop their visual identities. These are the elements that will compose the visual language and where users can apply the dynamism; if desired these elements can also be static.

The initial plan was to provide users tools to develop their own forms (rectangles, ellipses, triangles, polygons) or their own text (providing default typography) (see appendix 2.). These elements would be the starting point of their visual identity, in which dynamism could be applied if desired. Later in the process, we concluded that this approach would constrain user's creative potential resulting in quite similar visual languages, so we decided that users would be able to upload their own graphic elements in the form of SVG (Scalable Vector Graphics). This change allowed a greater potential to the tool because it gives users opportunity to explore the tool with their own graphic elements.

SVGs are vector graphics written in Extensible Markup Language (XML), that are rendered by most web browsers. One advantage of using SVGs is that they are accessible within the DOM (Document Object Model), and we were already using DOM. Therefore, it is possible to customise their appearance using CSS (Cascading Style Sheets) and manipulate them using JavaScript.

We started programming in p5.js (see appendix 2.), a library for JavaScript that has great potential for creative coding. However, we could not use the uploaded SVGs as vector images, it only worked when SVGs were rendered into pixels. Even using a specific library for loading SVG (p5.js-svg), the library showed many limitations in regard to using SVG as vectors, namely, a simple function such as rotate seemed to cause issues. This problem was pointed out by other users in GitHub and StackOverflow (Figure 33). As our goal was to take advantage of vector properties, we have decided to change the path and program with JavaScript as this language showed way more capacity to manipulate SVG files without losing their vector properties.

Figure 33.
p5.js SVG limitations pointed out by users on GitHub and StackOverflow.

The SVG accepts different types of data such as forms, raster or bitmap images and text. However, at this preliminary stage, the tool only accepts forms, which are vectors — lines and curves described mathematically. In SVG, <path> is the element that defines a shape, the d attribute inside this element holds the path data that provides the mathematical information required to create the shape's outline (Figure 34).

```
<path d="M0.75 0.75v30.5h30.5v-30.5zM28.75 28"></path>
```

Figure 34.
SVG <path> representation.

In future work, we hope to allow the upload of SVG images and text. In this stage, if users want to upload these graphic elements, they need to create outlines in Adobe Illustrator, or on other graphic design software. To guarantee the SVG is being exported in the form of <path>, it is possible to check the code on Adobe Illustrator when exporting as SVG (Exporting As > SVG > Show Code).

In HTML, the element <svg> is a container for SVG graphics, which can receive various <g> — a g is a SVG element used to group other SVG elements, namely path — and various <path>. When uploading, the system creates a single <svg> that receives the <path> of each uploaded graphic element and embeds them in a <g>. The <g> is a group that it is named according to the order of upload (Figure 35). Basically this <svg> is the element that will comprise all uploaded SVGs into one single container. This <svg> has the size of the canvas, allowing all paths to be visible while bounded by the limits of this canvas.

```
<div id= “canvas”>
        <svg>
        <g id=”svgElement-1”>
                <path> [...] </path>
        <g  id=”svgElement-2”>
                <path> [...] </path>
        </svg>
</div>
```

Figure 35.
HTML representation of the <svg> container from our system.

The reason why we embed all paths inside the same <svg> container was in order to avoid the DOM from treating multiple SVGs as separate elements. Before deciding to use this approach we were creating an <svg> element for each SVG uploaded. As they are DOM elements they were placed after each other rather than being located above each other. To tackle this problem we initially set all <svg> to position absolute in CSS, which was removing them from the normal document flow and visually stacking them on top of each other (Figure 36).

```
<div id= “canvas”>
        <svg> [...] </svg>
        <svg> [...] </svg>
        <svg> [...] </svg>
</div>
```

Figure 36.
Previous system structure.

Later, we realised this was generating problems when exporting. Because the browser was treating them as separate elements rather than a whole, the export was resulting in incorrect representations of the canvas. In these exports, the SVGs positioning was inaccurate to what the user was seeing on screen. Therefore, we changed our method and grouped all SVG in one single SVG document. This way, any CSS or positioning applied to the container

<svg> affects all the embedded SVGs collectively. Furthermore, this approach not only avoids the issues mentioned before as it also allows proper control over the SVGs, by manipulating the <g> or the <path> directly.

As we have seen, when uploading the system gets the <path> of each upload and inserts in the main svg inside a <g>. Furthermore, it deletes any information inside the <svg> that is not relevant such as empty <g> elements. If the SVG has more than one path, the system merges all paths into one single path. Having many paths was resulting in too much complexity when applying transformations during the development process, so we decided to overcome these difficulties by merging all paths into a single one. Unfortunately, this can sometimes lead to distortions, if the SVG is too complex. It is possible to overcome this issue using Adobe Illustrator, going to Object > Compound Path > Make (Figure 37). By doing this, the SVG will have a single <path> which will prevent distortion problems.

When this happens, it means that transformations will be applied to the unique <path>, even if it has many forms, the user cannot apply, for example, different colours to different forms within the same uploaded SVG. When the system joins the paths it removes any style information from individual paths which means that if the user uploads an object like this, the system will set it like that (Figure 38).

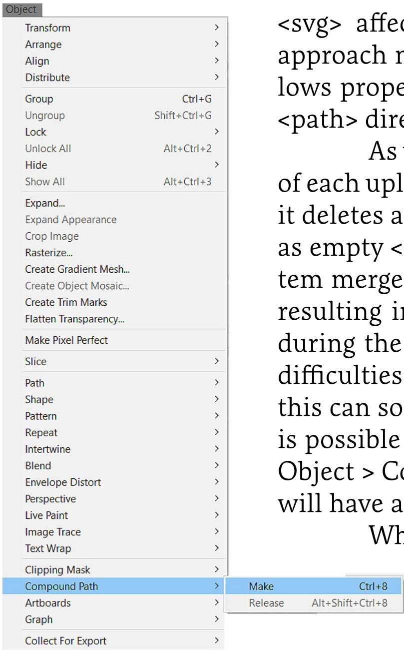


Figure 37.
Make Compound Path in Adobe Illustrator to avoid SVG distortions.

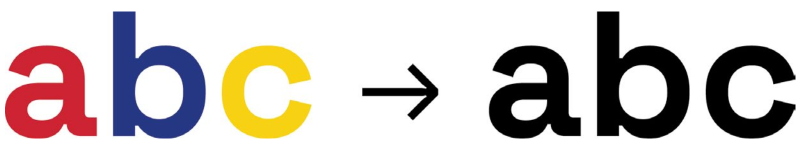


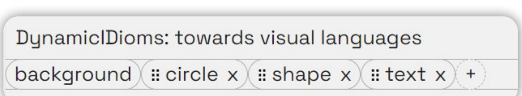
Figure 38.
The system ignores SVGs style information such as colour.

Our decision to convert all SVGs to their most basic form was due to the project's purpose that lies more in exploring dynamic processes for visual identities rather than manipulating highly complex graphic elements. Later, the user can change the colour or even insert images inside these forms. We aim that future work can address this issue.

→ Interface

The interface begins with an indication to upload the SVG files. A button with a plus sign allows multiple uploads at once. After the upload is done, the initial indication disappears and buttons with the name of each SVG appear inline, resembling different web tabs. Displaying the file name facilitates a relation between each SVG and each button. Each of these buttons will then allow individualised interaction with each graphic element (Figure 39).

Figure 39.
Graphic Elements Buttons.



All elements appear in the canvas by order of upload. If users need to change order, these buttons can be grabbed and placed before or after any other. An icon for drag-and-drop actions — six squares in a grid of 2x3 — is displayed in each button so users understand this behaviour. If there is a replacement of buttons, the corresponding SVGs will change their appearance within the canvas according to the new buttons order. This was developed with jQuery.js UI library, using a function sortable — enables a group of DOM elements to be sortable by clicking on and dragging an element to a new spot within the list while the other items will adjust to fit. By default, sortable items share draggable properties. Moreover, each button has a cross for deleting, which gives users the possibility to eliminate graphic elements if these are not desirable anymore.

Apart from having the file name, when clicking on each button its corresponding SVG gets highlighted. All SVGs that are not selected get a degree of opacity. This gives users a visual perception of which graphic element they are playing with. The selected button gets darker, to indicate its active state and opens a new working space on the left side of the canvas that allows composing the element and / or applying dynamism. Clicking above the selected button or in the "canvas area" closes the new working space.

Regarding the background, it can be considered as a visual component of any graphic artefact. Therefore, the background is a default asset of the system that is previously defined as an element possible to manipulate. It is defined as a button that makes part of the same area of any other uploaded graphic elements, and has the same visual properties. Although its working space will differ a bit from other graphic elements due to its different nature, it also allows to style it and to apply dynamism.

2. Composition and Style

→ Background

The background is pre-defined in the system as a button. Clicking on this button opens a working area with a style section. The section to style the background allows users to change its colour or to upload an image that will fill the format of the canvas. The user changes colour by using a colour-picker, which the user can always manipulate.

It is possible to add an image by clicking above the container. When the user uploads the container gets the image of the upload and its stroke gets filled indicating that it has a selected image while the image fills the background according to the chosen format. When there is an upload, a cross appears so the user can always change or delete the background image.

→ Elements

The composition area for each SVG allows the manipulation of its position, size, rotation and colour. Each option can be manipulated through the use of sliders, except, colour that is applied through a colour-picker. In this section, users can set the layout and appearance of static elements or the starting point of dynamic elements.

All transformations are applied through CSS directly in the element <path> of the selected SVG (Figure 40). To apply these transformations we opt for the use of sliders. The slider interaction updates each property: position sliders update translation values in x (horizontal) and y (vertical); the rotation slider updates the element rotation degree (from 0 to 360 degrees); scale slider updates the scaling value and the colour-picker updates the element fill attribute, changing the colour of the <path>. All transformations are always applied in the centre of the element.

Figure 40. CSS transformations applied to the <path> element.

```
<path [...]  fill="#000000" style="transform: translate(10px, 10px) rotate(20deg) scale(1)"></path>
```

CSS transformations are about how elements are rendered on the screen, so it does not change the actual values of SVGs, only how they appear in the canvas. Because SVGs can have different internal coordinates and dimensions, applying the same attributes to different SVGs can result in different visual outcomes. For example, if the real size of an SVG is 20x20 pixels and another is 800x800 pixels, even though they can appear the same size on the screen, they will present quite different values for the same CSS transformation.

This added some complexity to the program development. However, we wanted to make it possible for all SVGs to be uploaded and displayed accurately in the canvas. We did not want to restrict the possibility of upload due to internal information that most times is hidden from users. The solution was to have sliders with different ranges adapting to the internal values of each uploaded SVG. For example, in position, the slider range is set from the negative value corresponding to the width of the path to the width of the canvas. This means that a SVG with 20px size translated in (10px, 10px) can be visually in the same place as an SVG with 800px translated in (-100px, -100px). This was a challenge that accompanied the system development all the time. For size, it is also noticeable that some will have a bigger range to grow than others, this is according to its real value. Size uses scale, and in the same way, it affects different SVGs in different ways according to their real values.

We chose to keep sliders from showing values, not only because of the issues mentioned above but also, given the intent of developing an exploratory tool we found to be enough to have merely visual responses to the user interaction.

This working area in HTML is only one for all SVGs, thus we had to create a way to update sliders for each SVG. This function gets the information of SVG and updates sliders accordingly. Everytime a new SVG is selected the sliders update.

3. Dynamism

Dynamism is, as well as composition and style, a section within the working space of individual SVGs. This allows the creation of diverse dynamic components within the same visual identity as each SVG can detain its own form of dynamism. Both sections are interconnected, meaning that modifications made in the previous section will have an impact on this section.

For the creation of dynamism, we applied the variation mechanisms established by Martins et al. (2019) — Colour Variation, Content Variation, Positioning, Repetition, Rotation, Scaling, Shape Transformation. It is by the selection of these mechanisms that the user starts to apply different forms of dynamism to each graphic element. As this workspace is specific for each graphic element it will only apply variation to the selected SVG. Users can apply variation mechanisms to all elements if desired. However, they need to set them individually.

The background can also display some form of dynamism. However, given its nature, it is only possible to apply colour variation or content variation. To apply it, users open the menu for background, similar to the menu for each SVG.

Users can always combine more than one variation mechanism. These mechanisms will be enchained by order, and the outputs vary accordingly. For example, applying rotation before repetition is different from applying repetition before rotation as we are going to explain later (see section 5.2.2 — 3.2). This enchainment allows higher possibilities to the tool and showcases the potential of variation mechanisms as integrated units. We call this a "grammar" which we will explain further.

Figure 41. Selection of variation mechanisms.

3.1 Variation Mechanisms

When opening the section dynamism, the user sees the following message: "select variation mechanisms to apply dynamism to your svg". A button with a plus sign permits adding variation mechanisms (Figure 41). Clicking on one will establish the respective variation mechanism to the SVG element. Following, a button will be displayed in the working area.

Each variation mechanism button opens a small menu that allows users to define the dynamic range for that specific variation. For example, when choosing rotation users can define the range from minimum to maximum degrees in which the element can rotate. Establishing these ranges can yield dynamism while providing the needed constraints to build a cohesive visual identity. Every mechanism starts with default values relative to the values setted on the previous section. This enables a rapid perception of the mechanism effects without having to manipulate anything.

To define these ranges we used the noUiSlider library, a JavaScript range slider that allows sliders with two handlers. The manipulation of these handlers specifies the minimum and maximum values for variation mechanisms. Users can define these limits as well as change them whenever needed. For colour variation or content variation users need to select colours or upload images instead of applying ranges, therefore these do not start with default values as they require the mentioned actions.

The system allows the selection of one of each variation mechanism. After getting displayed, the selected variation mechanism disappears from the selection box because it cannot be chosen again. The possibility of having more than of the same variation mechanism could be considered for future work, for example repeating the repetition. However, for the present version of the project, we decided to implement only one of each, which we believe to already display great potential.

When the user selects any variation mechanism, a button "apply" appears. This button has more contrast than others due to its importance in the system. This is the button that permits users to check the variations being applied to their elements. The button gets the defined range values and makes a random between the minimum and maximum. It applies the variation mechanism with the random value. Every time it is clicked, the user can see the results varying in the canvas.

The system does not allow specific manipulation inside a variation mechanism, it only allows the selection of the range in which the variation will happen. To check the possible outputs for the selected variation range users click on "apply".

As we've seen, when users upload an SVG it creates a <g> with the respective path. When applying variation mechanisms the program creates a new <g> which will be a clone from the first <g>. Clone is a JavaScript method that creates a clone of an object, identical in every way, but stored in a different variable. This new <g> is called <clone-svgElement> as it gets the name of the main element (Figure 42).

Figure 42.
Clone of the SVG element.

```
<g id= svgElement-1>  [...] </g>
<g id=clone-svgElement-1> [...]  </g>
```

This is the <g> that will receive variation, while the initial one remains untouched. This allows saving the initial information if users want to delete or remove variations. Also, it facilitates the interaction between this working section and the previous one: if the user changes the element in “compose”, the “apply dynamism” section will clone the first <g> and receive the new transformations.

Moreover, each variation mechanism can be easily deleted by simply clicking on the cross displayed in each button. All these functionalities allow users to very easily test new forms of dynamism and play with different visual outputs.

Following, we explain individually each variation mechanism and its features.

→ Colour Variation

Colour variation allows varying the colour of the SVG element (Figure 46) or the background (Figure 47). In our system, the function allows the selection of colours in which the graphic element will vary. The menu starts with six empty colour pickers. Users do not need to necessarily select six colours. However, to observe results they need to pick at least two.

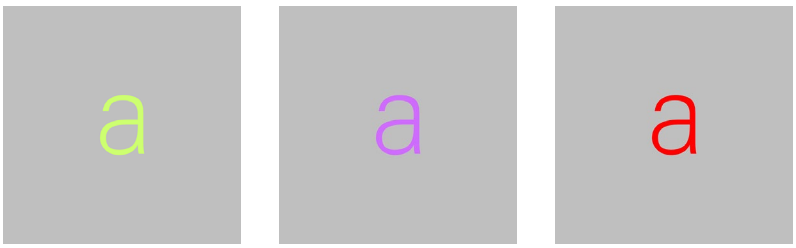


Figure 46. Varying colour of a SVG element.

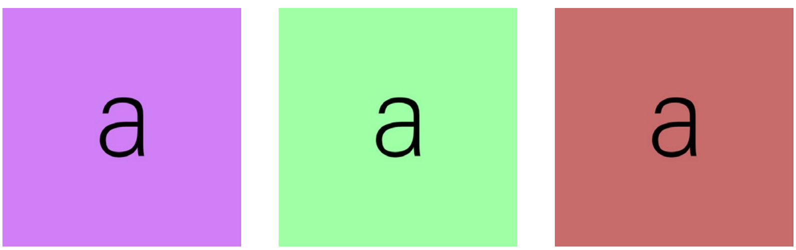


Figure 47. Varying colour of the background.

The empty colour-pickers start with a dashed stroke, when colour is selected the stroke gets solid, indicating to users that the colour is chosen. Furthermore, when selected, the colour-picker gets a cross down below to eliminate the respective colour if desired. Users can always change the colour of colour-pickers.

In programming, an array of colours is being created when colour variation is selected. Each array gets the colours chosen for the respective SVG element. Clicking on "apply" will make a random in this array and fill the SVG element with the picked colour.

The same logic is employed in the working area of the background by applying the CSS function background-colour.

→ Content Variation

Content variation is referent to changing imagery either inside a graphic element (Figure 48) or in the background (Figure 49). In content variation, the menu is similar to colour variation. Six empty containers with a plus symbol indicate the possibility to upload files. Each container allows the upload of one image. The uploaded images will be inserted inside the <g> element as <image> elements. <image> is an element from the SVG document that can display raster image files (PNG, JPEG) and other SVG files.

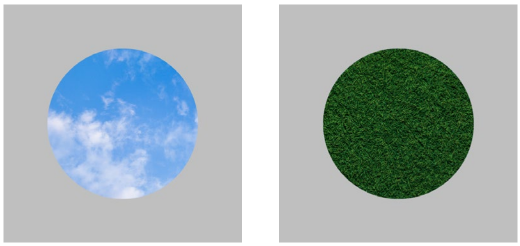


Figure 48. Varying content within a SVG element.

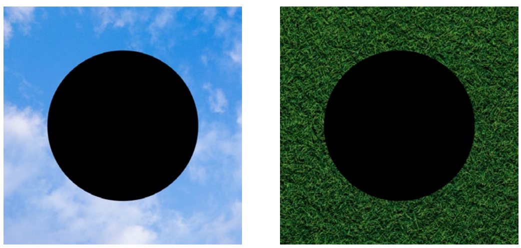


Figure 49. Varying content in the background.

To accomplish this, we used an element called <clipPath>. This element sets the boundaries of the visible image and functions like a mask: parts of the image that are inside these edges are shown, while those outside are hidden. The <path> is the element that will define the clipping region: inserted inside a <clipPath> (Figure 50). The uploaded images, rendered into SVG <image> are put inside the <g> with the CSS property clip-path that will then make a connection between both the image and the path. the <clipPath>. This property allows images to be masked into our SVG element.

Figure 50. HTML clipPath representation.

```
<g id=clone-svgElement-1>
        <clipPath id=myClipPath >
                <path> [...] </path>
        </clipPath>
<image clip-path=”url(#myClipPath)” > [...] </image>
</g>
```

All the uploaded images will be inside the <g>. Clicking on “apply” will display them randomly. The system selects one image randomly and displays that one, while the rest get set to display:

none. Similarly to colour variation, an array gets the data of the images and creates them when needed. For example, if users go back to composition and change the original <g >, there is the need to create a new clone with the new composition settings; if there is content variation, the new clone needs to receive the defined images inside and make a clip-path again. To do this, the system calls the array and creates new images and a new clipping-path in the new clone.

For background, content-variation is applied with the CSS function background-image, always filling the space of the chosen format. Again, it gets a random image from the array of images chosen to the background and sets the picked one to fill the background.

This mechanism was the hardest to implement. Given the concerns we already mentioned regarding the internal dimensions and coordinates of SVGs, it was not trivial to insert images that would fit the <path> or that would match its position on the canvas. To accomplish this, every time an image is uploaded, the program gets the bounding box of the <path> element and the x and y position of the top left corner and sets these properties into the <image>. Still, this wouldn't always match the <image> and the <path>.

Due to different orientation and aspect ratios of images and SVGs, sometimes the images would not fill the space of the SVG element. To tackle this problem, a function compares the aspect ratio of images with the aspect ratio of the SVG's <path>. If the aspect ratio of the <path> is greater than or equal to the aspect ratio of the image, it means that the path is wider or has the same proportions as the image. In this case, the code adjusts the width of the image to match the width of the path element's bounding rectangle, making sure that the image fits within the path. If the aspect ratio of the path is smaller than the aspect ratio of the image, it means that the path is narrower compared to the image. In this case, the code adjusts the height of the image to match the height of the path. Images are never distorted inside, they are setted according to path.

Moreover, when images are inserted into the <g> these <image> start to define the boundaries and size of the <g> which means that applying transformations to the <g> would get the centre of the image rather than the centre of <path>, which in our program is the main element. We solved this by applying transformations directly to the <path> or to the <image>.

→ Position

Position implies the change of position of the graphic element within the given format (Figure 51). Position menu has two sliders, one referent to horizontal position and another one to vertical position. This allows SVGs to move only in one direction only if desired. For instance, setting the minimum and maximum of vertical position to the same value and only applying a range on horizontal position will make the SVG move horizontally only. To achieve this variation, the function changes the translation values of the <path> inside, using the CSS property transform.

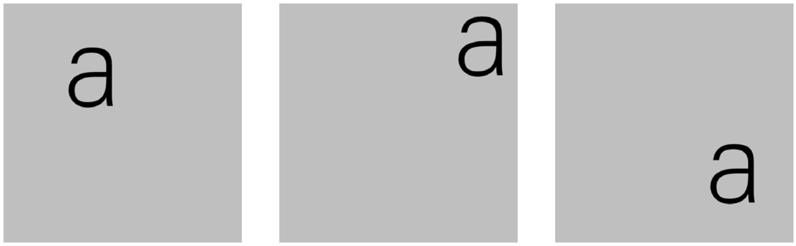


Figure 51.
Varying position.

The default values start with a small range relative to the position set on composition, both horizontally and vertically. Similarly to the position sliders mentioned in the composition section, these had to receive the maximum and minimums according to the SVG real size due to discrepancies in SVGs internal values (see section 5.2.2 — 2).

→ Repetition

Applying Repetition allows repeating the graphic element vertically and horizontally (Figure 52). In this menu there users can define the minimum number and the maximum number of repetitions, through an input box. For this range we opted not to use sliders, so the user knows exactly the number of repetitions. The system will repeat randomly within the defined range. Minimum is one and maximum is fifty. If users want to always have the same number of repeated elements they can set the minimum and maximum to the same value.

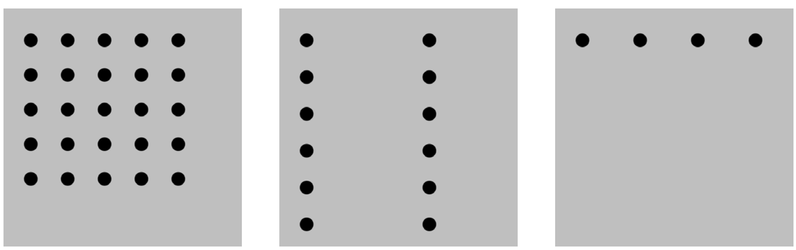


Figure 52.
Varying repetition.

A slider permits defining a range of varying space between the repeated elements. If users do not desire to vary this space they can set the minimum and maximum to the same value. This logic happens both for horizontal repetition and vertical repetition. Users can define to only repeat elements one way.

Repetition has a particularity from other variation mechanisms. To achieve the chain of variation mechanisms (the grammar), we needed to guarantee that variation mechanisms would show different results when applied in different orders. This means that after applying repetition, the remaining variation mechanisms would be applied to the group of repeated elements rather than on the first element.

When repetition is chosen the system gets the cloned <g> and creates a new <g> called repetition with the group repeated elements inside (Figure 53). This will permit applying the transformations to the <g id=repetition> instead of applying to the <g id=-clone>.

```
<g  id=clone-svgElement-1> [...] </g>
<g id= repetition-svgElement-1 >[...] </g>
```

Figure 53.
Repeating clones.

Everytime "apply" is clicked a new <g id=repetition> clones the cloned <g id=clone> according to the number randomly picked from the ranges defined previously, making a translation from left to right, up to down. If there are variations being applied to the clone, the <g id=repetition> will get it because it is cloning that element. If transformations are after repetition, they will be applied to the <g-repetition>. Any transformation before repetition will be applied to the path of the clone. From the point that the user chooses repetition, any variation will be applied to <g id=repetition>. We address this point further in grammar (see section 5.2.2 — 3.2).

→ Rotation

In Rotation, the graphic element is rotated (Figure 54). This mechanism allows defining a varying rotation from 0 to 360 degrees through a slider. It uses the CSS property "transform" to rotate the element. Clicking on "apply" will get a random value from the range defined by the slider and set the rotate function.

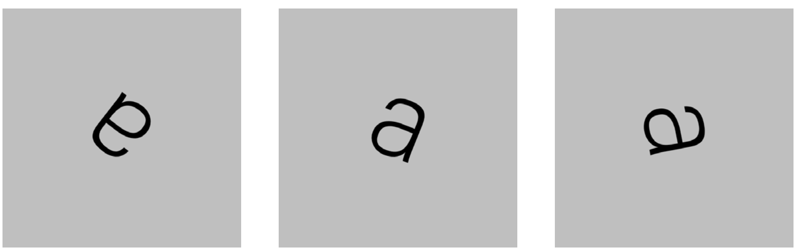


Figure 54.
Varying rotation.

→ Scale

Scale makes graphic elements change in size (Figure 55). A slider allows users to define the minimum and maximum range. This mechanism uses the CSS property to scale elements. It always scales

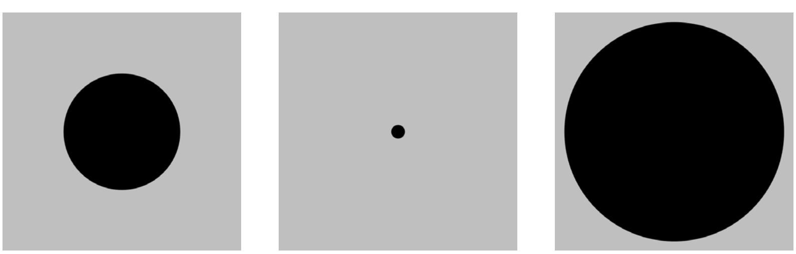


Figure 55.
Varying scale.

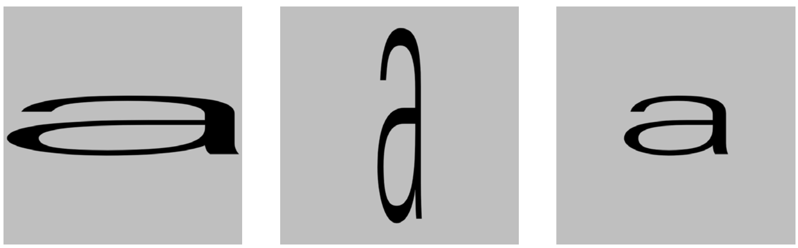
proportionally to the dimensions of the element. Clicking on "apply" will set a random value from the range defined in the slider and change the scale values.

The scaling permits a range from 0.8 to 20. The scaling function multiplies these values with the size of the element. For example, if an object with 100px gets a scale value of 5 it will appear 500px. As we have seen earlier, the real properties of SVG can affect scaling, meaning the same value can have apparent different results in different SVGs.

→ Shape Transformation

In shape transformation there is a graphic element that changes in shape (Martins et al., 2019). Ideally our program would provide a variety of different shape transformations as shape transformation can be many things. For our prototype we only apply scale horizontally and/or vertically, allowing some degree of shape distortion (Figure 56). Although we acknowledge the could be inside scale and not shape transformation we decided to use it to showcase some distortion.

Figure 56.
Varying transformation on with and on height.



Shape transformation allows applying scale only on width or on height, or both. Clicking on "apply" will get a random value from the range defined in the sliders and set the value to <path> element.

3.2 Grammar

When selecting more than one, these buttons are displayed from top to bottom, translating their sequencing (Figure 57). This sequence is reflected in the visual results, as the order of application affects the visual result. Buttons have colours to give users a better perception of this order. Moreover, it highlights the significance of variation mechanisms in our system.

This sequencing of variation mechanisms can be seen as a visual grammar as it provides rules to the dynamic content. This possibility not only shows the potential of variation mechanisms combined, it also evidences the versatility of our system, as same variation mechanisms can result in different visual outputs.

According to the Cambridge Dictionary a grammar is "the rules about how words change their form and combine with other words to make sentences". We call this functionality a grammar because, in our system, the order of mechanisms defines the rules about how dynamism changes its form and how the combination of different variation mechanisms produces different visual languages and meanings.

In order to demonstrate how order affects results, users can easily change the variation mechanisms order by simply drag and drop the desired button. An icon for drag and drop is displayed in each button, indicating this functionality. When this happens, the visual output will be displayed according to the new order rules. This functionality allows users to test very easily the best visual outputs for their chosen variation mechanism.

To accomplish this chain, each SVG that has dynamism entails an array that gets the order of the chosen variation mechanisms. For each mechanism, the array receives the values defined by the respective sliders. This array also changes its order when buttons positions are re-ordered. Mechanisms are then applied to the visual outputs according to the order of variation mechanisms inside the array.

Each variation mechanism is prepared to act in any position within this array. However, the architecture of code had to be thought of to allow these differences in the output. These requirements had to be considered along the programming development. We explain them in the following paragraphs.

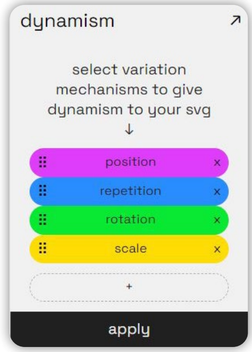


Figure 57.
Sequencing of selected variation mechanism.

→ Colour Variation & Content Variation

Both these mechanisms can be applied at the same time. We have decided to develop it this way, because, otherwise, the selection of one after another would override the effect of the previous. For example, selecting content variation after colour variation would cancel the effect of colour variation. In our analysis we verified that there are visual identities that apply a combination of these both (see section 4.1), thus we chose to allow this opportunity as well.

If both of these mechanisms are selected for the background, the program will make a probability between both and either set content variation or colour variation for different outputs (Figure 58). Thus, it is possible to set dynamic backgrounds varying between different images and colours.

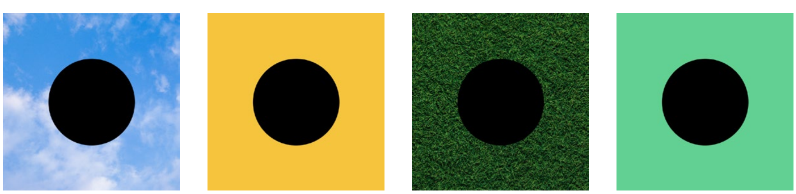
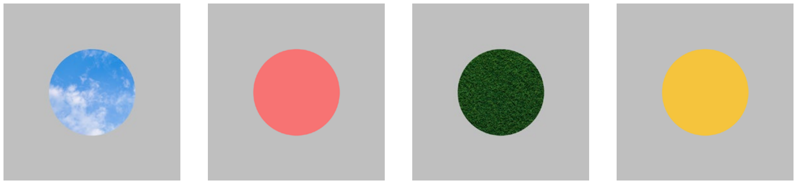


Figure 58.
Combining colour variation and content variation in the background.

The same is applicable for SVG elements (Figure 59). If users select colour variation and content variation the elements will vary between different images and colours. When combining these mechanisms, order does not influence the output. For this reason, the background does not offer the possibility to change mechanisms' order as it would be ineffectual.

Figure 59. Combining colour variation and content variation in an element.

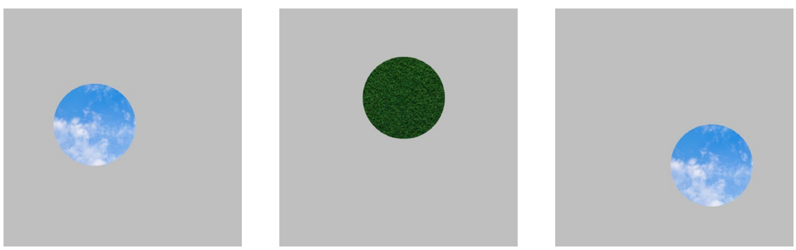


→ Content Variation & Position / Scale / Shape transformation

These mechanisms always consider if content variation exists in the array. As they are performed in the element <path>, when applied, the images that were selected from content variation, were not getting any transformation which was causing displacements. To tackle this, each of these variation mechanisms applies transformations to images if content variation is selected.

If position changes and there is content variation,the function gets the new position and changes the position of x and y of the image according to the new position of the element (Figure 60). The visual result is the same whether position is placed before or after content variation.

Figure 60. Content variation & Position.



If scale changes, the size of the image changes according to the new size of the element (Figure 61). Scale achieves the same results whether is placed before or after content variation.

Figure 61. Content variation & Scale.

The images in shape transformation adapt its width or height according to the new aspect ratio produced by the distortion (Figure 62).

Figure 62.
Content variation & Shape Transformation.

These mechanisms combined have higher chances of leading to misplacements. In some iterations, the image may not completely fill the newly transformed element. However, given that this occurrence only happens in a few instances, we have decided to move forward while acknowledging this issue. Shape transformation achieves equal results whether it is placed before or after content variation.

→ Content Variation & Rotation

Content Variation with rotation have different behaviours according to the order. If rotation is before content variation the path element is rotated while the image stays in place (Figure 63). Here the rotate function is applied to the path element. This case has a higher probability that the path will not fill the image completely on the edges. However, it is a minimal inaccuracy, so we have decided to proceed and leave this to be solved in future work.

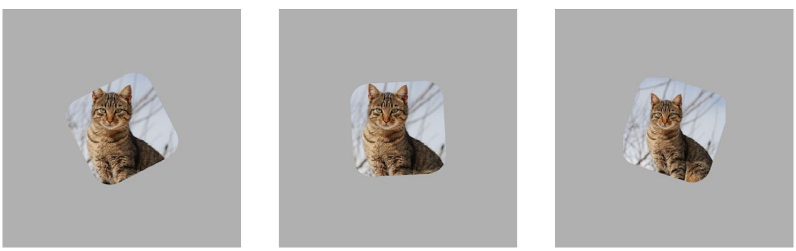


Figure 63.
Rotation befor content variation.

If content variation is before rotation, the rotate function is applied to the image. When transformations are applied to the image, the clip-path also suffers the transformations. In this condition, both elements are rotated (Figure 64). To do this we had to get the centre of the path in order to define the rotation centre of the image. As most time images are bigger than the path, rotating the image from its centre was making displacements of the SVG element.

Figure 64.
Content variation
before rotation.

Figure 65.
Content variation
before repetition.

→ Content Variation & Repetition

In both these functions are conditions that consider if the other exists and in which position of the grammar. If content variation is before repetition, the images are inserted in the <g-clone> and then the repetition function repeats this clone with the images inside. Thus, visually, each element has an image inside (Figure 65).

If content variation is after repetition the behaviour is quite different. Instead of repeating the <g-clone> (Figure 66) the repetition gets the <path> of this clone and applies the repetition to it (Figure 67).

Figure 66.
HTML representation
of content variation
before repetition.

```
<g id=repetition-svgElement-1>
        <g id=clone-1>  [...]
                <image> [...] </image>  [...]
        </g>
<g id=clone-2>  [...]
                <image> [...] </image> [...]
        </g>
<g id=clone-3>  [...]
                <image> [...] </image> [...]
        </g>
</g>
```

Figure 67.
HTML representation
of repetition before
content variation.

```
<g id=repetition-svgElement-1>
        <clipPath>
                <path> [...] </path>
                <path> [...] </path>
                <path> [...] </path>
        </clipPath>
<image> [...] </image>
</g>
```

This will make a clipping mask as a whole. The image will be masked by the group of repeated elements (Figure 68). The reason why it repeats differently was because <clipPath> does not function if there are <g> inside it, so the solution was to repeat the path. Com-ing to this conclusion led us to decide to join all paths into a single one when uploading SVG elements (See section Upload SVGs). We concluded that having this possibility would be more relevant for our system than allowing complex and colourful reproductions of the uploaded SVGs.

As we've seen, when applying content variation to a single element, the <clipPath> is made inside the clone (see section 5.2.2 — 3.1). When applying content variation after repetition, the <clipPath> is made to <g-repetition> and the uploaded <image> is inserted inside this <g>. The CSS function clip-path is then applied to the images inside the <g-repetition>.

The logic to fill the space of this group with the images is the same as to fill a single one. The code makes a comparison between aspect ratios of images and the aspect ratio of the group bounding box and sets the width or height of the image accordingly. Furthermore, it gets the x and y position of the first path and sets the image to that point. This is made every time the user uses "apply" so it always gives an accurate filling of the clip path.

Figure 68. Repetition before content variation.

→ Repetition & Position / Rotation / Scale / Shape transformation

As we have seen, the repeating function creates a new <g> with repeated elements. Variation mechanisms position, rotation, scale and shape transformation, have conditions to apply to different <g> according to where they are positioned in the array, if after or before repetition. If these mechanisms are applied after it applies the variation in the <g>. If they are before they apply to the clone. This allows achieving the following results.

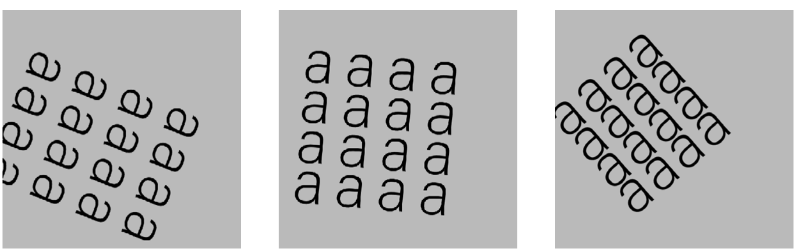


Figure 69. Repetition before rotation.

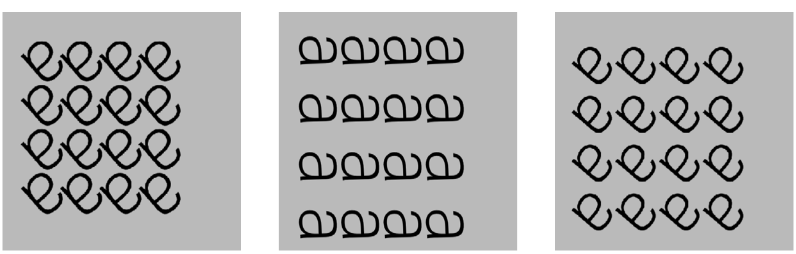


Figure 70. Rotation before repetition.

→ Position & Scale & Rotation & Shape Transformation

The impact of the combination of some variation mechanisms, for example position, scale, rotation and shape transformation, is relatively minimal, as the outcomes are quite similar regardless of order. Although the system is applying the CSS transformations by order, the results do not showcase variations due to the nature of these mechanisms. These mechanisms have higher impacts when combined with, for example, repetition, which hasalso been assessed in our analysis (see section 4) . Yet, they are relevant mechanisms to set dynamism for visual identities, even if used individually.

4. Formats

Having different formats allows to experiment different needs,testing the possibilities of the visual identity. The tool offers three formats: a square (1x1), a poster (4x5) and a banner (16x9). Moreover, users can customise their own format, using sliders to change the width and height of the canvas. This allows users to define different formats for developing different graphic artefact, depending on their wishes.

This functionality can be found right below the canvas. Changing the format not only adapts the canvas as it also adapts the position of the existing elements.

To adapt the position of the SVGs the system gets the position of the element, compares it to the size of the canvas and then when the format is changed it makes a rule of three: multiplying the current position with the new canvas size and dividing this value by the old canvas size will provide the new position value. This is made for all elements displayed in the canvas. Also, if there is content variation applied to any SVG, the system also changes the position of the images according to the new position and to the new format.

5. Give me dynamism

This button is the one to be most used as it is the one that will produce different visual results for the variation mechanisms selected by the user, thus enabling an easy way to assess the possibilities of the grammar as a visual identity. As it is a button with high importance in the process of experimentation, we decided to place it in a central position within the interface and with a contrasting appearance from other buttons..

This functionality gets the order of variation mechanisms set to each dynamic element — the ones where dynamism was employed — and applies the defined variations to all, at the same time. While the apply button inside each SVG working space only allows one to see the dynamism applied to that specific element, this button permits to check dynamism employed to all elements.This makes users perceive how these dynamic elements coexist within the visual identity.

Clicking on "give me dynamism" will randomise the values defined previously within the grammar of each dynamic element, always displaying different outputs based on the rules defined for each. This function allows users to rapidly check the results as a whole and perceive the potentialities of different dynamic components to achieve a dynamic visual language.

6. Gallery

The tool allows saving the visual results achieved. Given that we are working within the context of visual identities, having a gallery to keep all results together becomes highly relevant to perceive their coherence. The results are displayed in a gallery, allowing the user to perceive the visual identity as a whole. Although it is possible to perceive the visual dynamism using the button "give me dynamism", it is by seeing the results side by side that the user gets the notion of the dynamic possibilities of the generated visual language (Figure 71).

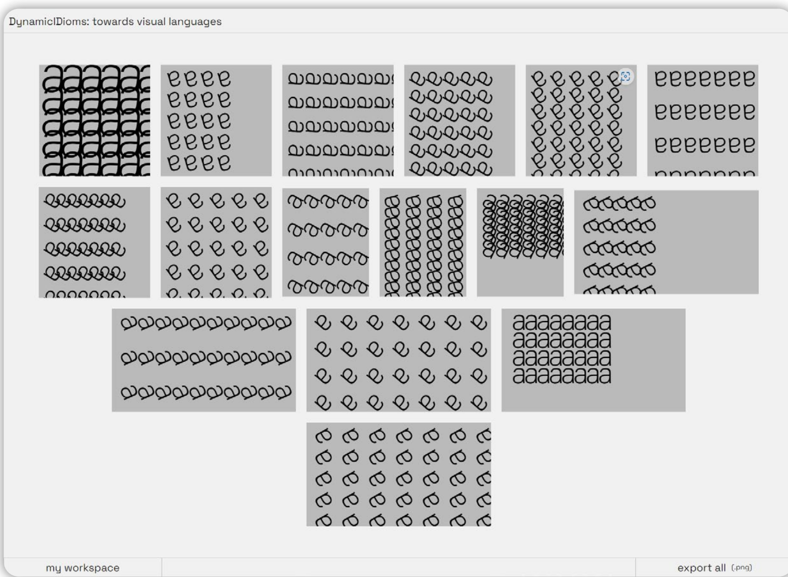


Figure 71. Gallery with saved results.

Right on the side of the "give me dynamism" button there is a button "save to gallery". Their togetherness was set on purpose, so users can easily save results they like while continuously testing new visual results. While we were developing the system we noticed this should be a requirement so their interaction is easier and faster.

Clicking on this button will save the current result displayed in the canvas on the gallery.

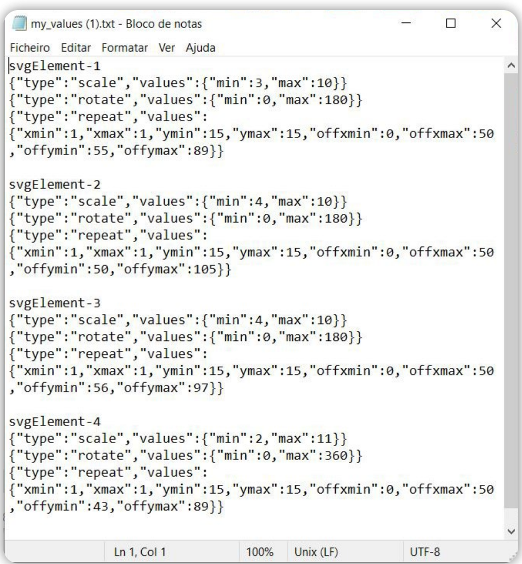
Right beside the button to save there is the button "my gallery". This button takes users to another interface section where the visual results are displayed in a grid and the working area disappears. When entering this area the button "my gallery" transforms itself to "my workspace" allowing a rapid interplay between both. This gallery is exclusive to each user and each interaction, so if the user did not start playing and saving results the gallery will be empty.

7. Export

The system allows exporting PNG images. A button to export the output displayed in the canvas is located on the right side of the "give me dynamism" button so users can easily export any outcome they like. Exporting images gives users the opportunity to use them for mockups and other digital purposes. For exporting, we employed the library html2canvas — takes screenshots of the user's browser.

We initially had the goal to export content as PDF so users could take advantage of the vectorial properties of SVGs. We were able to accomplish this but only by exporting an image as a PDF, which was not desirable due to the loss of the vectorial properties. We experimented with multiple libraries such as svg2pdf, pdfmake, html2pdf.js, but we were not able to make the system export a vectorial output that matched the canvas. We decided in this stage, to export only PNG images, with the intention to later offer the possibility to export in PDF so users can easily manipulate the outcomes in other software if desired.

Figure 72.
Grammar values in TXT file.



svgElement-1
{"type":"scale","values":{"min":3,"max":10}}
{"type":"rotate","values":{"min":0,"max":180}}
{"type":"repeat","values":
{"xmin":1,"xmax":1,"ymin":15,"ymax":15,"offxmin":0,"offxmax":50
,"offymin":55,"offymax":89}}

svgElement-2
{"type":"scale","values":{"min":4,"max":10}}
{"type":"rotate","values":{"min":0,"max":180}}
{"type":"repeat","values":
{"xmin":1,"xmax":1,"ymin":15,"ymax":15,"offxmin":0,"offxmax":50
,"offymin":50,"offymax":105}}

svgElement-3
{"type":"scale","values":{"min":4,"max":10}}
{"type":"rotate","values":{"min":0,"max":180}}
{"type":"repeat","values":
{"xmin":1,"xmax":1,"ymin":15,"ymax":15,"offxmin":0,"offxmax":50
,"offymin":56,"offymax":97}}

svgElement-4
{"type":"scale","values":{"min":2,"max":11}}
{"type":"rotate","values":{"min":0,"max":360}}
{"type":"repeat","values":
{"xmin":1,"xmax":1,"ymin":15,"ymax":15,"offxmin":0,"offxmax":50
,"offymin":43,"offymax":89}}

In the gallery, there is a button to export all that exports all saved artefacts into a ZIP to the user's computer. To do this, we employed the javaScript library jszip. This permits the easy export of a series of varied graphic materials for the same visual identity in a single click.

The tool also provides a function to export values into a txt file (Figure 72). These values are referent to the variation mechanism and respective range values of each dynamic element that form the rules of the whole visual system. It exports the dynamism applied for each SVG element — the grammar — and if there are elements that do not have dynamism, the file will display the text "static element". This function allows users to save the rules of the identity system they developed. It was important to export these values so users get the understanding that these systems have coherence because they are based on predefined rules.

However, right now this is only a TXT file that provides the formula for the developed visual identity. We hope in the future these files can have other functions such as the opportunity to upload these rules and the system adapts to the rules within the file — allowing users to continuously work on their visual identities.

5.2.3
—
Results

For developing the results we have decided to explore two different approaches: one of free experimentation and other that answered to specific briefings. We chose to conduct the results this way because we believe our tool can inherently offer both possibilities.

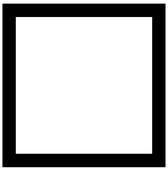
For the experimentation approach we tested the system possibilities with different graphic elements and then applied them to mockups, showcasing possible applications and visual identities. For the other approach we asked a colleague to answer specific briefings provided by chatGPT. We have decided to conduct experiments with other users in order to exceed personal constraints and gain a broader understanding of the system's capabilities in the most varied contexts.

Free Experimentation

In this part, we experimented applying variation mechanisms into different graphic elements without rules or predefined briefings. This allowed us to perceive many different potentialities of the system for the generation of varied visual languages. The next pages showcase the developed results.

↘

Graphic Element



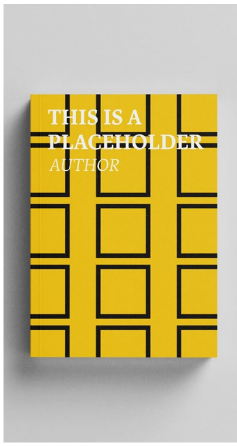
→ 1 graphic element
Colour Variation / Rotation / Repetition

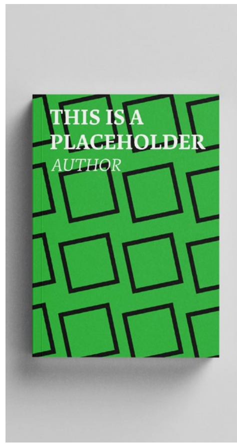
This example explored a pattern produced by applying of rotation, repetition into a square. The square is rotated and then the repetition is applied. Furthermore, the background is varying its colour. Then we applied the generated patterns to mockups.

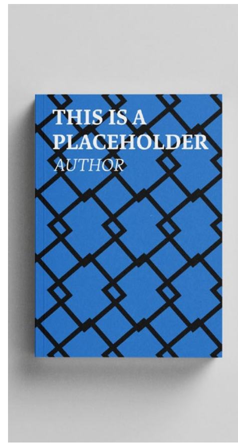
Grammar

→ Background
{"type":"colourvariation"}}

→ Square
{"type":"rotate","values":{"min":0,"max":360}}
{"type":"repeat","values":{"xmin":4,"xmax":10,"ymin":4,"ymax":10,"offxmin":0,"offxmax":50,"offymin":0,"offymax":44}}



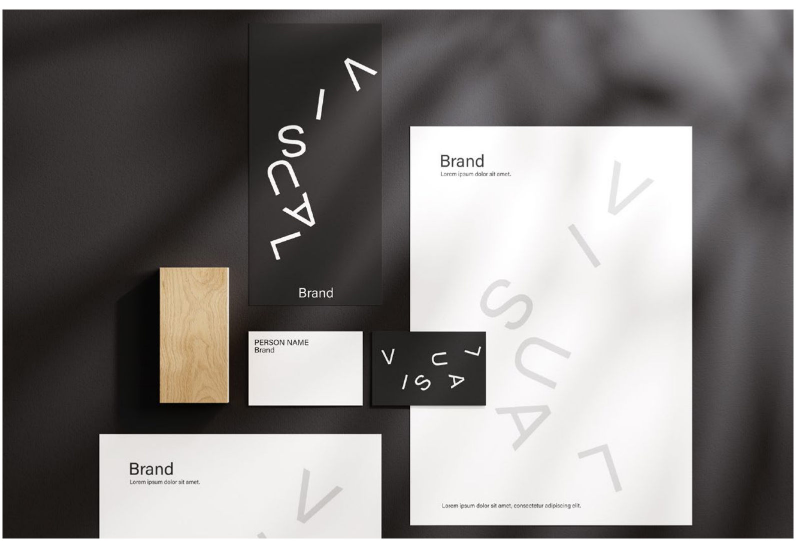




→ Typography
Position / Rotation

Here we explored dynamism applied to typographic elements. For this there is a need to upload each letter individually to apply different forms of variation to each. All of them have rotation and position, however their position changes vertically.

In a case of a visual identity applying this kind of dynamism these variation mechanisms could be applied to different messages and not only to single word.



Graphic Elements

Grammar

→ V
{"type":"position","values":{"minx":38,"maxx":318,"miny":26,"maxy":26}}
{"type":"rotate","values":{"min":0,"max":180}}

→ I
{"type":"position","values":{"minx":67,"maxx":284,"miny":111,"maxy":111}}
{"type":"rotate","values":{"min":0,"max":180}}

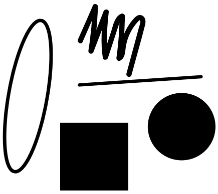
→ S
{"type":"position","values":{"minx":44,"maxx":325,"miny":203,"maxy":203}}
{"type":"rotate","values":{"min":0,"max":180}}

→ U
{"type":"position","values":{"minx":49,"maxx":305,"miny":292,"maxy":292}}
{"type":"rotate","values":{"min":0,"max":180}}

→ A
{"type":"position","values":{"minx":41,"maxx":283,"miny":386,"maxy":386}}
{"type":"rotate","values":{"min":0,"max":180}}

→ L
{"type":"position","values":{"minx":20,"maxx":274,"miny":469,"maxy":469}}
{"type":"rotate","values":{"min":0,"max":180}}

Graphic Elements



Content Variation

Grammar

→ Background
{"type":"colourvariation"}}

→ Ellipse
{"type":"rotate","values":{"min":0,"max":360}}
{"type":"colourvariation"}
{"type":"transform","values":{"wmin":1,"wmax":23,"hmin":1,"hmax":30}}

→ Line
{"type":"rotate","values":{"min":284,"max":360}}
{"type":"repeat","values":{"xmin":1,"xmax":1,"ymin":1,"ymax":4,"offxmin":156,"offxmax":228,"offymin":42,"offymax":62}}

→ Square
{"type":"position","values":{"minx":28,"maxx":373,"miny":-38,"maxy":331}}
{"type":"scale","values":{"min":2,"max":6}}
{"type":"rotate","values":{"min":0,"max":180}}

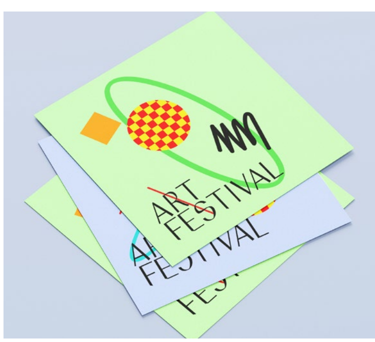
→ Circle
{"type":"contentvariation"}
{"type":"position","values":{"minx":30,"maxx":379,"miny":-9,"maxy":312}}

→ Doodle
{"type":"rotate","values":{"min":0,"max":180}}
{"type":"position","values":{"minx":21,"maxx":345,"miny":-34,"maxy":351}}

→ Various graphic elements
Colour Variation / Content Variation / Position, Repetition / Rotation / Scale / Shape Transformation

This visual language was inspired by the work of the portuguese tattoo artist Leonor Cunha; then it was applied to mockups in the form of a possible visual identity for an Art Festival.

We aimed to explore the creative and artisitc possibilities of our tool too. Even though this applies all mechanisms, it applies different to different graphic elements as it is possible to see in the grammar, thus it maintains aesthetical coherence.



→ Various graphic elements Scale / Rotation / Repetition

This identity was influenced by the visual identity for walker art center. However, with a limited set of graphic elements. We idealised the results could belong to a beauty brand. the generated patterns were used for labeling different products of the same brand.

Graphic Elements

Grammar

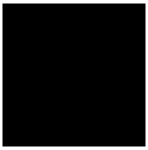
→ Rect
{"type":"scale","val-
ues":{"min":3,"max":10}}
{"type":"rotate","val-
ues":{"min":0,"max":180}}
{"type":"repeat","val-
ues":{"xmin":1,"xmax-
":1,"ymin":15,"ymax-
":15,"offxmin":0,"offxmax-
":50,"offymin":55,"offy-
max":89}}

→ Rect Slim
{"type":"scale","val-
ues":{"min":4,"max":10}}
{"type":"rotate","val-
ues":{"min":0,"max":180}}
{"type":"repeat","val-
ues":{"xmin":1,"xmax-
":1,"ymin":15,"ymax-
":15,"offxmin":0,"offxmax-
":50,"offymin":50,"offy-
max":105}}

→ Triangle
{"type":"scale","val-
ues":{"min":4,"max":10}}
{"type":"rotate","val-
ues":{"min":0,"max":180}}
{"type":"repeat","val-
ues":{"xmin":1,"xmax-
":1,"ymin":15,"ymax-
":15,"offxmin":0,"offxmax-
":50,"offymin":56,"offy-
max":97}}

→ Ellipse
{"type":"scale","val-
ues":{"min":2,"max":11}}
{"type":"rotate","val-
ues":{"min":0,"max":360}}
{"type":"repeat","val-
ues":{"xmin":1,"xmax-
":1,"ymin":15,"ymax-
":15,"offxmin":0,"offxmax-
":50,"offymin":43,"offy-
max":89}}

Graphic Element



→ 1 graphic element
Colour Variation / Shape Transformation

For this visual identity we explored colour variation and transforming the rectangle. This provided a high sample of different outcomes that could be used for social media posts.

Grammar

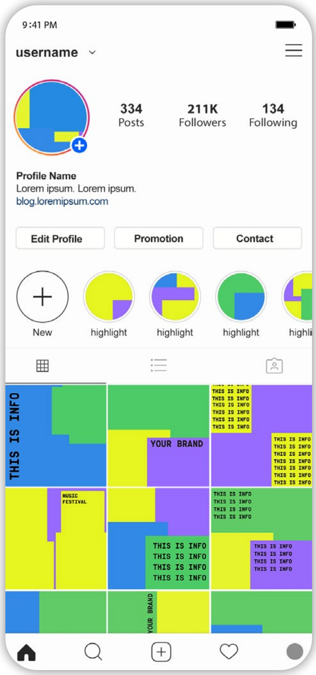
→ Background
{"type":"colourvariation"}

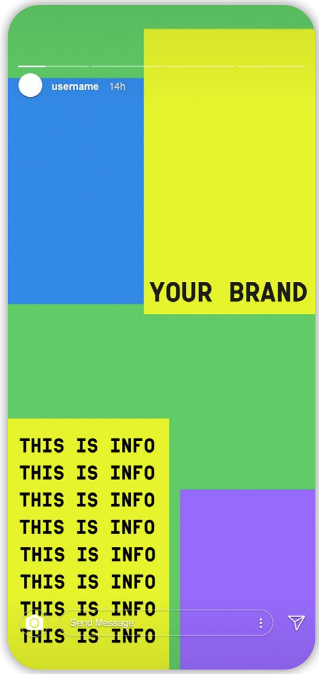
→ Square
{"type":"transform","values":{"wmin":1,"wmax":23,"hmin":1,"hmax":23}}
{"type":"colourvariation"}

→ Square
{"type":"transform","values":{"wmin":1,"wmax":17,"hmin":1,"hmax":17}}
{"type":"colourvariation"}

→ Square
{"type":"transform","values":{"wmin":1,"wmax":20,"hmin":1,"hmax":17}}
{"type":"colourvariation"}

→ Square
{"type":"transform","values":{"wmin":1,"wmax":15,"hmin":1,"hmax":14}}
{"type":"colourvariation"}

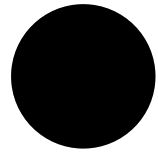




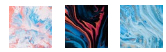
→ 1 graphic element & imagery
Colour Variation, Content Variation

For this, we have experimented the combination of colour combination and content variation. The results could belong to a visual identity for an health care center, for example.

Graphic Element



Content Variation



Grammar

→ Background
{"type":"contentvariation"}{"type":"colourvariation"}

→ Circle
{"type":"contentvariation"}
{"type":"colourvariation"}

During the development of these materials, we were not concerned with meeting briefings but rather with exploring different possibilities. Because these examples do not answer specific contexts, they could be applied for diverse applications or for diverse visual identities, thus, they are not so realistic. Yet, it became evident while developing these results that the tool has the potential for generating endless creative possibilities. Mainly, due to the possibility of uploading any type of graphic element. Nevertheless, even with a single visual element it is possible to achieve interesting and varied explorations through the combination of different variation mechanisms.

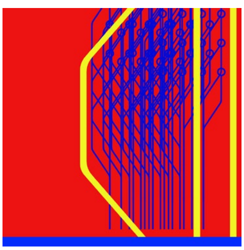
Answering Briefings

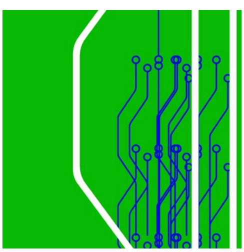
Being previously familiarised with the system and its possibilities conditions our expectations for possible outcomes. In order to test the system to its full potential and exceeding personal expectations, we had the desire to provide the tool to someone else. The next section showcases examples developed by other colleagues.

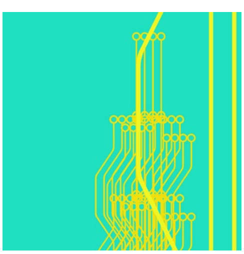
Visual Results

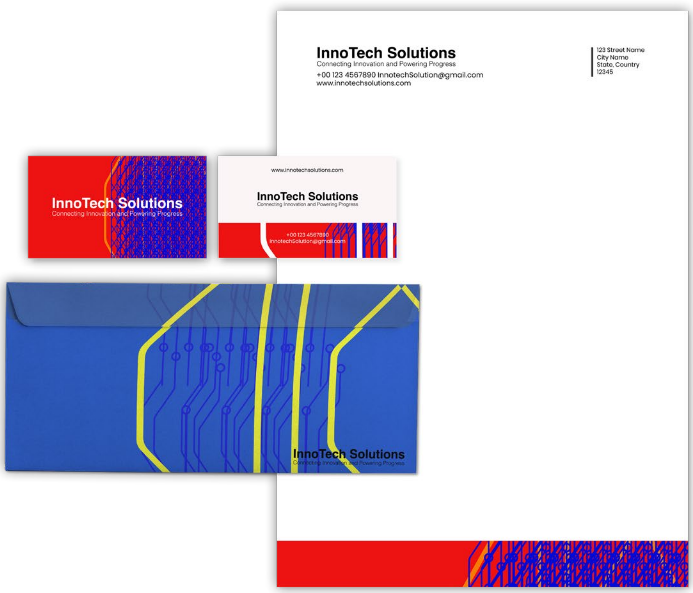
→ Briefing 1: Tech Startup

InnoTech Solutions is a forward-thinking tech startup specializing in innovative software solutions. They are seeking a visual identity that reflects their modern, innovative, and professional approach to technology.





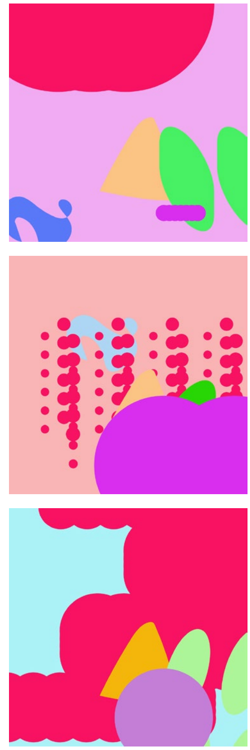




→ Briefing 2: Children’s Book Store

Imagination Kingdom Books is a brand for a Children's Book Store-that should be playful, containeing whimsical shapes like stars, clouds, and speech bubbles; the visual identity should evoke imagination and creativity.

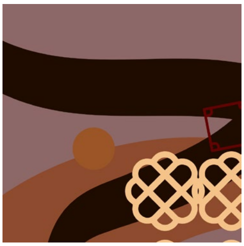
Visual Results

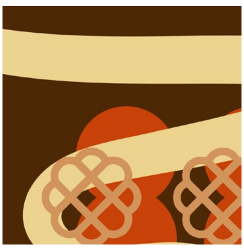


Visual Results

→ Briefing 3: Artisanal Coffee Shop

The branding for Le Café should exude a sense of warmth and comfort, inviting customers to relax and savor artisanal coffee in a cozy environment. Mosaic tile patterns and textures Coffee cup silhouette with steam swirls Curved, cozy shapes reminiscent of coffee stains.





5.3 — Conclusion

We acknowledge the small sample of results presented in this section, which happened as a consequence of having to extend the system development. Unfortunately, this need did not provide us with sufficient time to showcase the quantity of results we had initially predicted.

Nevertheless, we believe the shown results are a good demonstration of the system possibilities, as they are diverse and employ different dynamic components as well as different combinations of variation mechanisms. Furthermore, the results showcase dynamism while maintaining high degrees of coherence.

The results revealed that the tool is more useful for developing visual languages, such as patterns, than answering all visual identity requirements and communication requirements. For example, while the system allows static elements, most of them, from the shown examples, were subsequently placed in proprietary software such as Adobe Illustrator and Adobe Photoshop. Furthermore, given the complexity of visual identities and their necessity to answer to many different contexts, adapting all these messages appeared to be a more challenging and slow task. We conclude that the tool is more valuable for crafting the core visual language of a visual identity than to produce specific visual applications.

It was evident that the randomisation of results added significant value to the tool due to its surprise effect and easiness of the process, an impression that was also mentioned by colleagues while interacting with the tool. The ability to see different results so quickly was quite engaging. Moreover, colleagues understand the potential of combining different variation mechanisms for achieving different forms of dynamism.

Colleagues found the interface easy to navigate and understand. The gallery seemed to be undoubtedly an useful asset for viewing results collectively and exporting all at once in an easier way. Furthermore, the fact that the Give Me Dynamism button and the Save to Gallery are together proved to be efficient for optimising the interaction and use of the gallery.

While experimenting, it was also evident that it is possible to generate results that can be used as dynamic logos. However, this was not so much explored as it is not its purpose. The grammar and the exported values were beneficial for keeping the information of which mechanisms were applied and in which graphic element. Furthermore, having this knowledge provides the possibility to, if desired, apply again the same grammar to the same or to different visual elements.

Overall, we are satisfied with the results, even though we aim to continue exploring the tool possibilities. The results align with the goals we have set initially and the tool has proven to be valuable for not only engaging users in the development of visual languages as well as provide them with insights for their design.

5.4
—
Future work

Although we consider to have reached a good stage of the project, we have taken into account some aspects that could be developed in the future to add even more value to our system. More than anything, we aim to continue exploring the tool's capabilities and potentialities by generating more results and testing its limits. Moreover, we would like to continue working on the system, in order to enhance even more its functionalities.

Regarding the system implementation, we considered for future work to implement the variation mechanism Combination, which we did not implement at this stage. Adding to this, we believe it would be interesting to experiment more variation mechanisms, for instance the interaction mechanisms pointed out by Lorenz (2021) (see section 3.2). Moreover, we hope to introduce more variations of shape transformation.

We also considered the possibility of having more than one of the same variation mechanisms, for example repeating the repetition. However, for the present version of the project, we implemented the possibility to select only one of each, which we believe to already display great potential.

Regarding applying dynamism to graphic elements, we would like to permit a more detailed manipulation of variation mechanisms. For instance, users would have the capability to select precise values within a variation mechanism, instead of relying on randomisation for each iteration. We also thought that for each mechanism it could be possible to select specific values instead of defining a range. For example, inside the variation mechanism position, the user could define 3 specific positions in which the element could vary. For this prototype we found range manipulation enough to showcase the potential of variation mechanisms.

We would like our system to allow the upload of other graphic elements to enhance even more its capabilities. Additionally, it could be a possibility to make it responsive to smaller devices, so it could be accessible to an even broader public.

In the future, we aim to disseminate the project through the writing of scientific articles and through various other means.

6. — Conclusion

Branding is a significant area of communication design that continues to expand and evolve in parallel with technological advancements and the emergence of new media. Designers have the responsibility to gather the knowledge and tools to effectively respond to new contemporary demands. The past years have seen a rise in a new approach to visual identities, characterised by dynamic and flexible features. It is within this context that this dissertation emerges.

As seen in the document, we began by conducting an extensive literature review on this topic, which provided us with valuable insights for the development of our system. Following that, we analysed existing identities to gain a deeper understanding of their main characteristics and features. This analysis was extremely important for us to start the project development with a strong understanding of these visual identities and their possibilities.

During the system development, we have encountered challenges, mostly due to programming limitations, which have delayed our process. Nevertheless, we believe that the project exceeded our initial expectations, and we were able to implement features that were not originally proposed, such as the gallery, the possibility to export the grammar, and the possibility to drag-and-drop variation mechanisms to easily check the outcomes of different dynamic sequences.

Due to this commitment to the system development, we were not able to produce as many results as originally anticipated. Yet, we consider that the showcased results can demonstrate the tool's potential for both free experimentation and for answering specific branding briefings.

In our conclusions, we found that the system may not be the most suitable for designing realistic identities with all their intrinsic requirements. However, the system showed great potential in developing coherent visual languages, which was the primary goal of our project. Overall, we believe that we successfully achieved the goals we have defined at the beginning.

Despite there is room to improve the system, we believe this tool is a valuable contribution to the field of Dynamic Visual Identities. We believe that the tool meets the purpose for which it was developed: providing its users with a comprehensive tool kit for the development of dynamic visual systems and showcasing the advantage of computational tools for easing design experimentations and processes.

Bibliography

AnalogAlgorithm. (2019). https://www.analog-algorithm.com/

Bauman, Z. (2000). *Liquid modernity*. Polity Press ; Blackwell.

Benyon, L. (2018, March 19). *A Snapshot of Switzerland's Rich Poster-craft History*. Eye on Design. https://eyeondesign.aiga.org/a-snapshot-of-switzerlands-rich-poster-craft-history/

Black Light. (n.d.). TPN—BCN, HH. Retrieved 27 August 2023, from https://new.twopoints.net/black-light/

Blauvelt, A. (Ed.). (2011). Brand New Worlds. In *Graphic design: Now in production* (1st ed, p. 190 — 209). Walker Art Center.

Burnier, A. (n.d.). *Ciddic*—André Burnier. Retrieved 27 August 2023, from https://www.andreburnier.com/project/ciddic

Coelho, R. (2022). Flexible ID: to adapt is to resist. *Design, Visual Communication and Branding*, 82–107.

Conrad, D. (2023). Introduction I. *In Graphic Design in the Post-Digital Age A Survey of Practices Fueled by Creative Coding* (pp. 7–23). Set Margins' publications.

Costa, J. (2004). La imagen de marca: *Un fenómeno social* (1a ed., 5a imp). Paidós.

Costa, J. (2009). *Imagen corporativa en el siglo XXI* (4a ed). La Crujía.

Dual Room. (2023, May 8). *Label Suisse Festival Identity*. Instagram. https://www.instagram.com/p/Cr_RCQStzYc/

Dubberly, H. (2004). *How do you design? A Compendium of Models.* Dubberly Design Office.

Felsing, U. (2010). *Dynamic identities in cultural and public contexts*. Lars Müller Publishers.

Gerstner, K. (1964). *Designing Programmes* (1st Edition). Alec Tiranti Ltd. London.

Gorny, L. (2022). *New web tool Munken Creator lets you create downloadable, animated font experiments.* It's Nice That. https://www.itsnicethat.com/news/arctic-paper-munken-creator-graphic-design-191022

Grant, J., & Vodeb, O. (2023). *What is post-branding? How to counter fundamentalist marketplace semiotics* (First edition). Set Margins' publications.

Guida, F. E. (2014). *Generative Visual Identities: New Scenarios in Corporate Identity*. 121–132.

Guida, F. E., & Voltaggio, E. (2016). *Programming Visual Representations. Evolutions of Visual Identities between Tangible and Intangible.* 419–430.

Gysin, A. (n.d.). *Ertdfgcvb* (2009-2014). Retrieved 27 August 2023, from https://ertdfgcvb.xyz/p2/index.html

Heller, S. (2014, April 17). *The Lost, Surprisingly Soulful Art of Corporate Identity.* The Atlantic. https://www.theatlantic.com/entertainment/archive/2014/04/the-lost-strikingly-soulful-art-of-corporate-identity/360818/

Hollington, S. (2011). Flexible visual identity systems. *MA in Communication Design, Kingston University.*

Jochum, E. (2013). Dynamic Branding.

Kreutz, E. de A. (2007). *Identidade Visual Corporativa Mutante: Uma estratégia comunicacional contemporânea*. 15.

Kreutz, E. de A. (2012). *Construindo marcas mutantes. Chasqui*. Revista Latinoamericana de Comunicación, 119, 61–65.

Label Suisse festival. (2023, June 8). Fonts in Use. https://fontsinuse.com/uses/54047/label-suisse-festival

Landa, R. (2005). *Designing brand experience: Creating powerful integrated brand solutions.* Cengage Learning.

Leijsen, R. van, Conrad, D., & Lorusso, S. (2023). *Graphic Design in the Post-Digital Age A Survey of Practices Fueled by Creative Coding.* Set Margins' publications.

Leitão, S., Lélis, C., & Mealha, Ó. (2014). *Marcas que se querem mutantes: Princípios estruturantes e orientadores.* II International Congress Ibero-American Communication.

Lelis, C. (2021). Smart logos: A user's dashboard for the visualisation of meaningful brand experience data. *InfoDesign - Revista Brasileira de Design Da Informação, 18*(3). https://doi.org/10.51358/id.v18i3.892

Llop, R. (2014). *Un sistema gráfico para las cubiertas de libros: Hacia un lenguaje de parámetros.* GG, Editorial Gustavo Gili.

Long, M. (2020, September 23). *Space10 unveils "ever-evolving, democratic" visual identity and microsite*. Design Week. https://www.designweek.co.uk/issues/21-28-september-2020/space10-visual-identity/

Lorenz, M. (2021). *Flexible Visual Systems: The design manual for contemporary visual identities* (1st edition). Slanted.

Lorenz, M. (2017). From Static to Flexible Visual Identities. *GRAPHISME EN FRANCE 2017 Logos & Visual Identities,* 25–35.

Lupton, E. (Ed.). (2011). *Graphic design thinking: Beyond brainstorming* (1st ed). Princeton Architectural Press ; Maryland Institute College of Art.

Martins, T., Cunha, J., Bicker, J., & Machado, P. (2019). Dynamic Visual Identities: From a Survey of the State-Of-The-Art to a Model of Features and Mechanisms. *Visible Language*, 53, 4–35. https://doi.org/10.34314/vl.v53i2.4649

Meggs, P. B., & Purvis, A. W. (2012). *Meggs' history of graphic design* (5. ed). Wiley.

MIT Media Lab Identity. (n.d.). E Roon Kang. Retrieved 28 August 2023, from https://eroonkang.com/feature/mit-media-lab-identity

Munken Creator. (2022). https://colab.munken.com/munkencreator

Murdock, J. E. (2016). *Fluid identity: History & Practice of Dynamic Visual Identity Design* [Master's Thesis, Kent State University]. http://rave.ohiolink.edu/etdc/view?acc_num=kent1480096180818074

Nes, I. van. (2012). *Dynamic identities: How to create a living brand* (I. van Nes, Ed.). BIS Publishers.

Pearson, L. (2013). Fluid marks 2.0: Protecting a dynamic brand. *Managing Intell. Prop., 229*, 26.

Raposo, D. (2008). Design de Identidade e Imagem Corporativa. In *I+Diseño: Revista internacional de investigación, innovación y desarrollo en diseño, ISSN 1889-433X, Vol. 1, No. 1, 2009, pag. 194* (Vol. 1).

Reas, C., & McWilliams, C. (2010). *Form+code in design, art, and architecture* (1st ed). Princeton Architectural Press.

SPACE10. (2020). DIA Studio. https://dia.tv/project/space10/

SPACE10 Identity. (n.d.). Retrieved 28 August 2023, from https://identity.space10.com/

Vieira, B. L. (2022). *Sistemas de identidade visual generativos: Projeto, processo e potencialidades.*

Visit Nordkyn. (n.d.). Neue Design Studio. Retrieved 28 August 2023, from https://neue.no/work/visit-nordkyn/

Wheeler, A. (2013). *Designing brand identity: An essential guide for the whole branding team* (Fourth edition). John Wiley & Sons, Inc.

1 2 9 0 UNIVERSIDADE D
COIMBRA

Appendix 1.

—

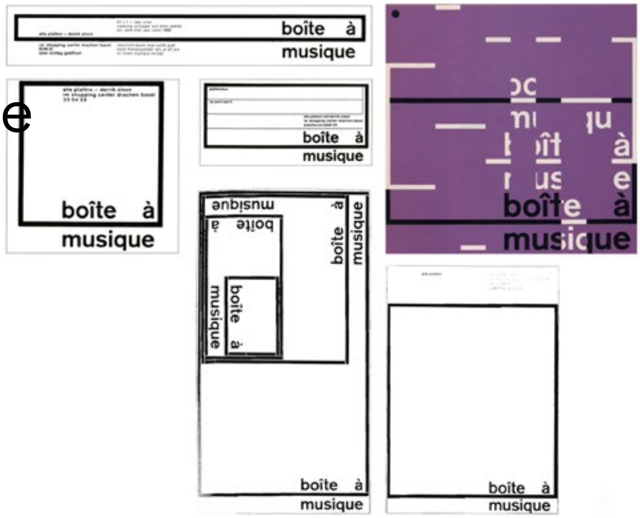
Dynamic Visual Identities

↘

This appendix contains all the visual identities analyzed in the dissertation Computational Tools for the Development of Dynamic Visual Identities in the Section 3.

The identities are numbered in the order of their appearance within the analysed lists.

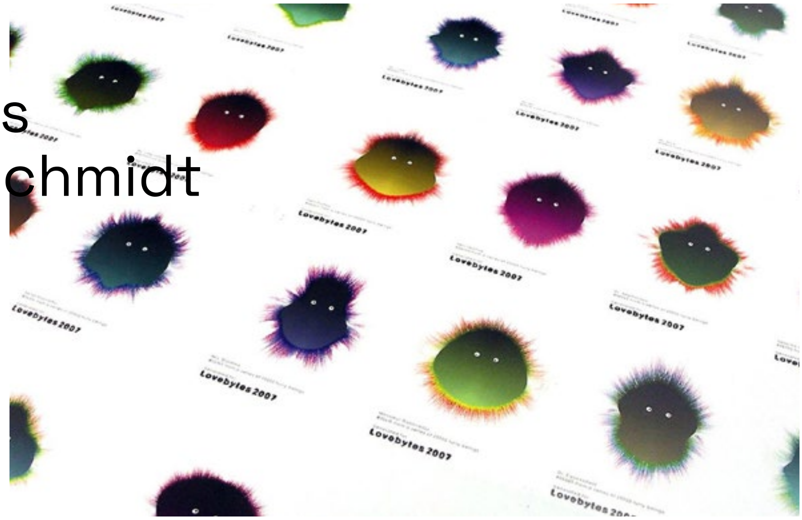
01.
Boîte à Musique
Karl Gerstner
1957



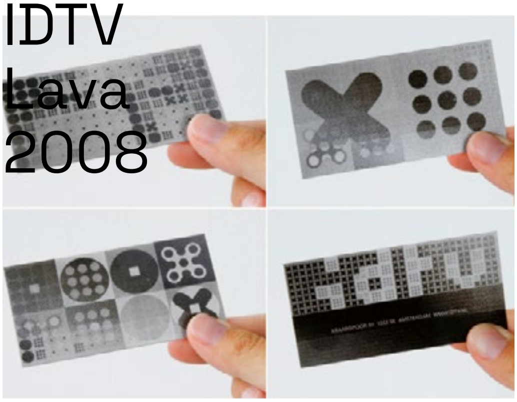
02.
Columbus Indiana
Paul Rand
1983

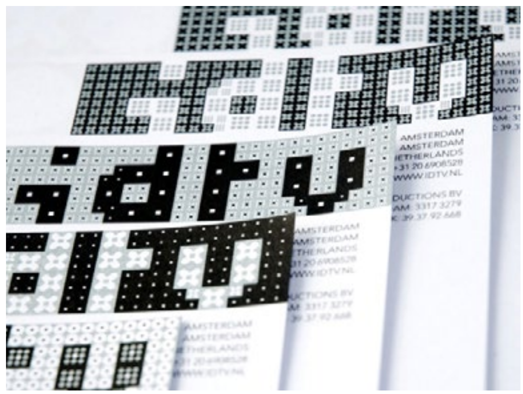
03.
Walker Art Center
Chad Kloepfer
& Andrew Blauvelt
2005

04.
Lovebytes
Karsten Schmidt
2007



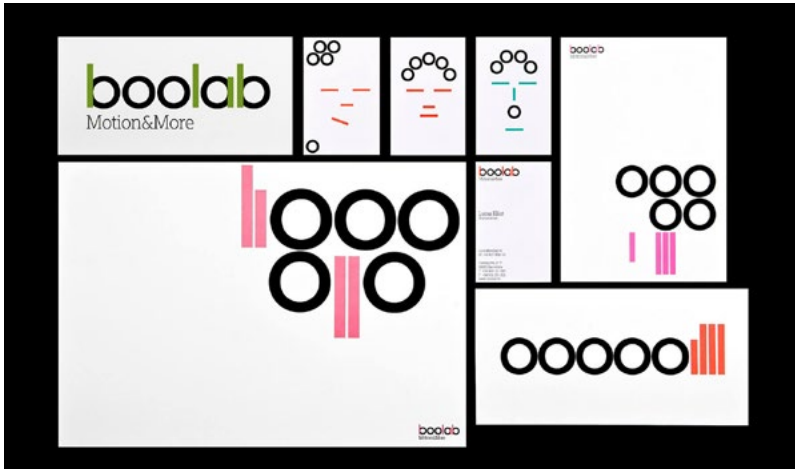
05.
IDTV
Lava
2008





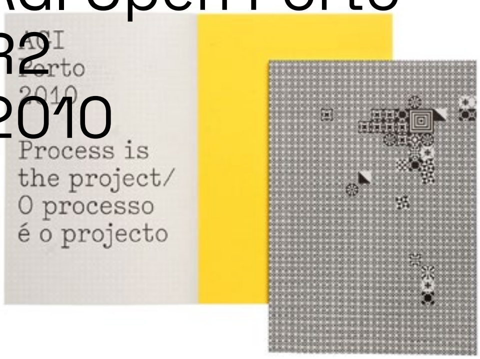
06.
Mobile Media Lab
Feed
2008

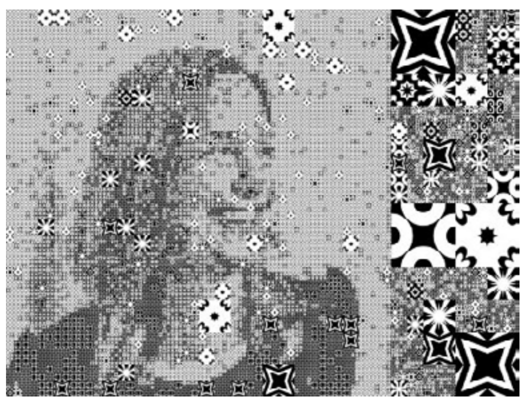
07.
Boolab
Mucho
2009



08.
Paramount
Mind Design
2009

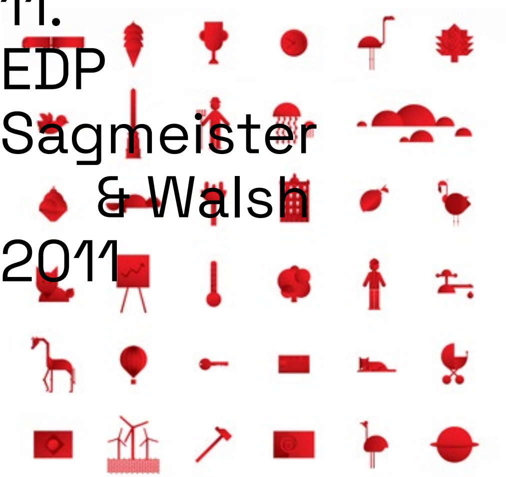
09.
AGI Open Porto
R2
2010





10.
Prima Vina Stiriae Slovenae
IlovarStritar
2010

11.
EDP
Sagmeister
& Walsh
2011



12.
OVG State
Studio Dumbar
2011

13.
São João Porto
R2
& Tiago Martins
2011

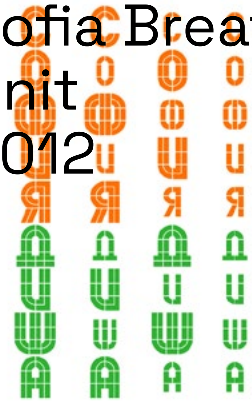
14.
Axis of Culture in Katowice
Aleksandra Krupa
2012

15.
Catalan Wines
Toormix
2012

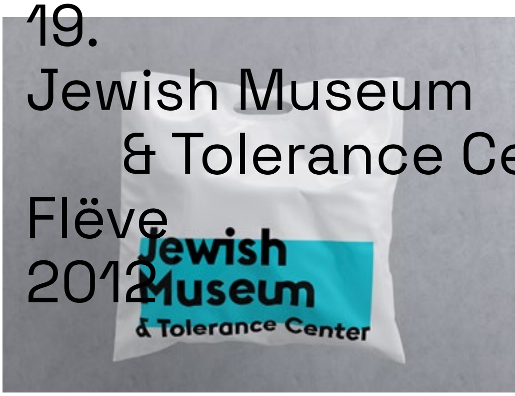
16.
Caminhos Film Festival
José M. Cunha & João Cunha
2012

17.
Schism
b2s6
2012

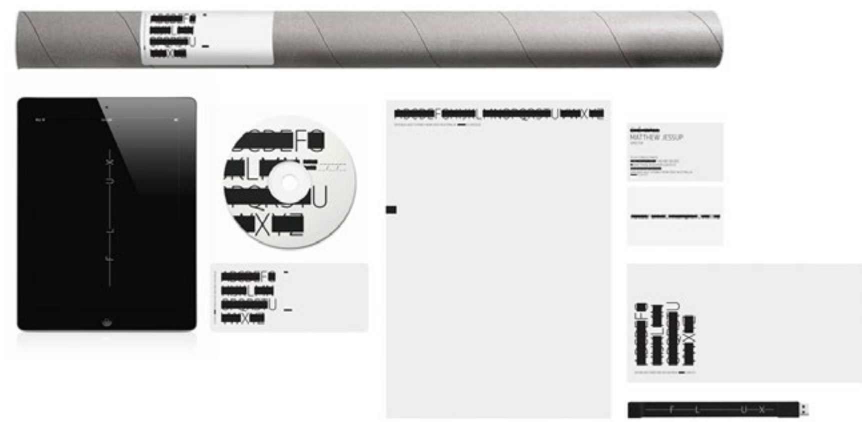
18.
Sofia Breathes
Unit
2012



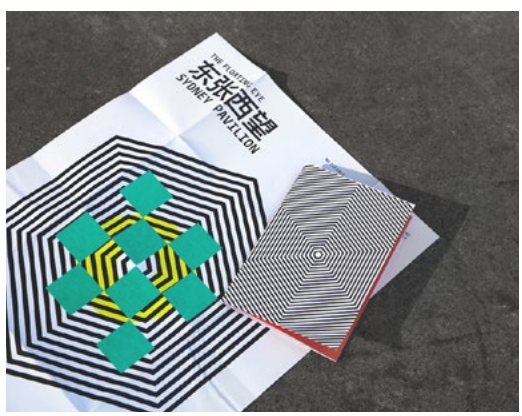
19.
Jewish Museum
& Tolerance Center
Flëve
2012



20.
Flux
Re
2012

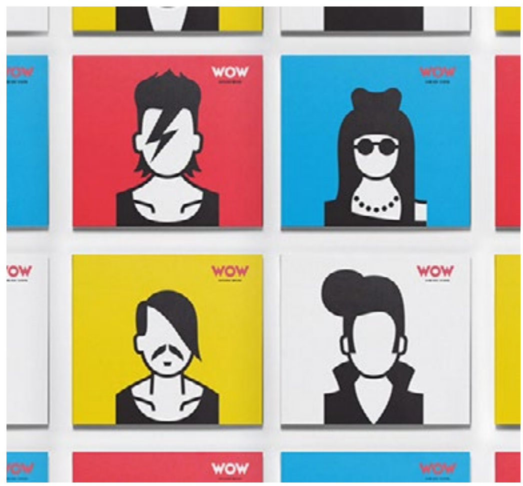


21.
The Floating Eye
Re
2012



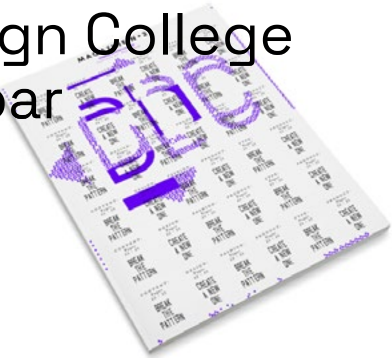
22.
KAKAO
Plus X
2013

23.
WOW
Vlad Likh
2013



24.
NAA TAA
Evan Dorlot
2013

25.
Dumbar Design College
Studio Dumbar
2013

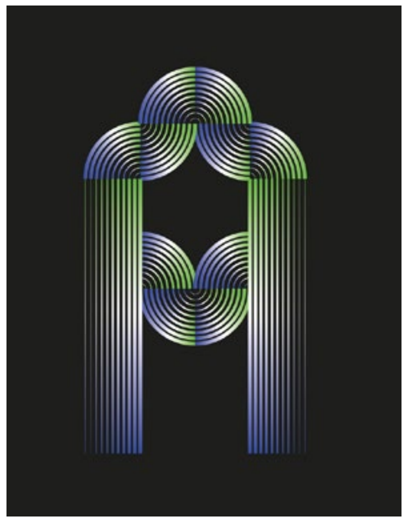



26.
Whitney Museum
of American Art
Experimental Jetset
2013

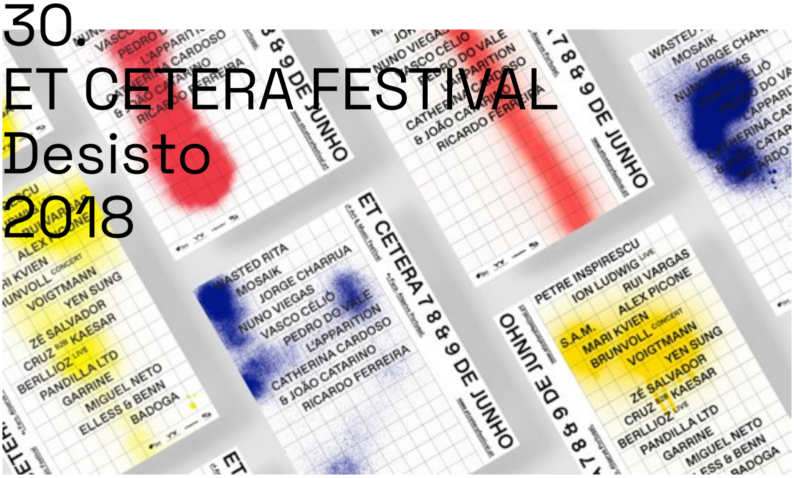
27.
Porto.
Studio Eduardo Aires
2014

28.
Medialab Prado
Tata & Friends
2017

29.
Black Ligth
TwoPoints.net
2018

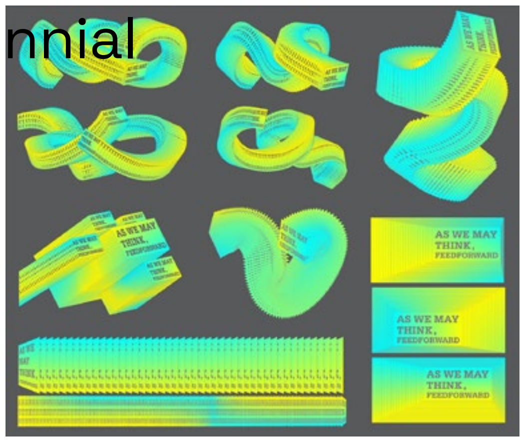


30.
ET CETERA FESTIVAL
Desisto
2018



31.
Caracol
Sometimes Always
2018

32.
6th Guangzhou Triennial
Another Design
2019

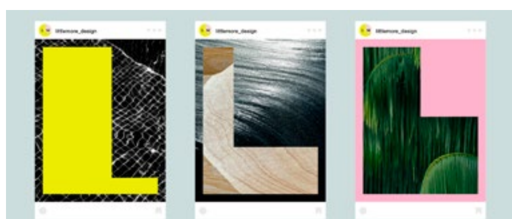


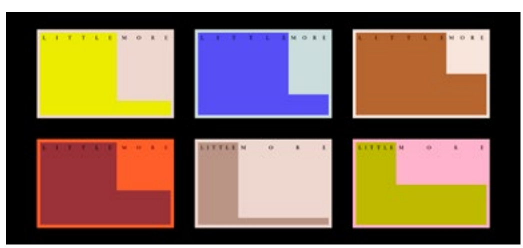
33.
Impira
DesignStudio
2019

34.
Manataco
Pentagram
2019

35.
Tribeca Festival
Pentagram
2021

36.
Littlemore
DutchScot
2021

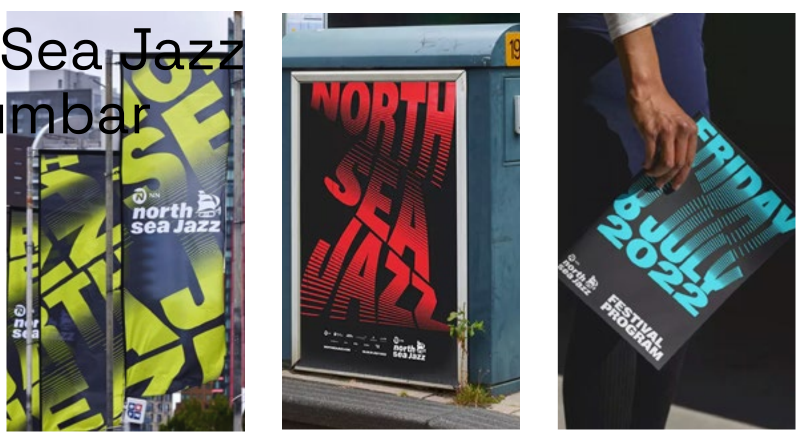




37.
Chaumont Biennal
DIA Studio
2021

38.
Baehl
Brand Brothers
2021

39.
NN North Sea Jazz
Studio Dumbar
2022



40. 35th Mercat de Música Viva de Vic
Quim Marin Studio
2023

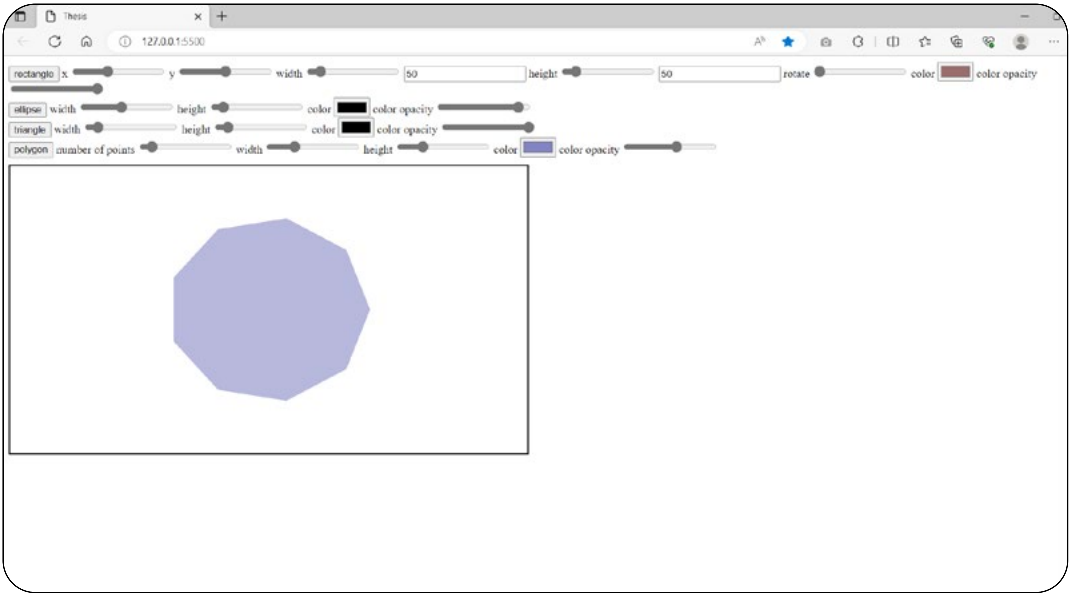
Appendix 2.

—

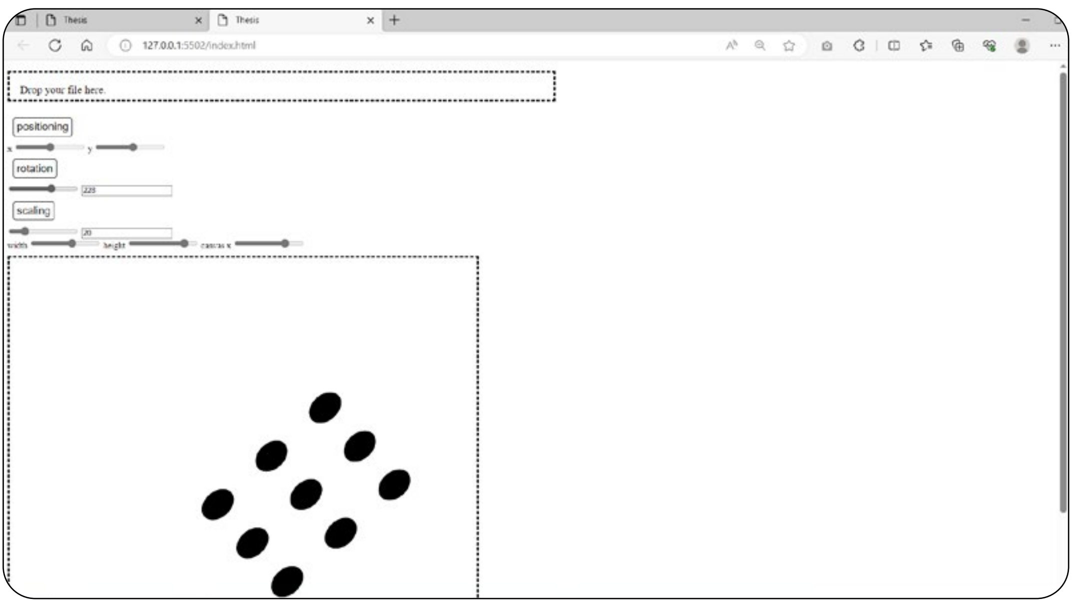
Tool Development

↘

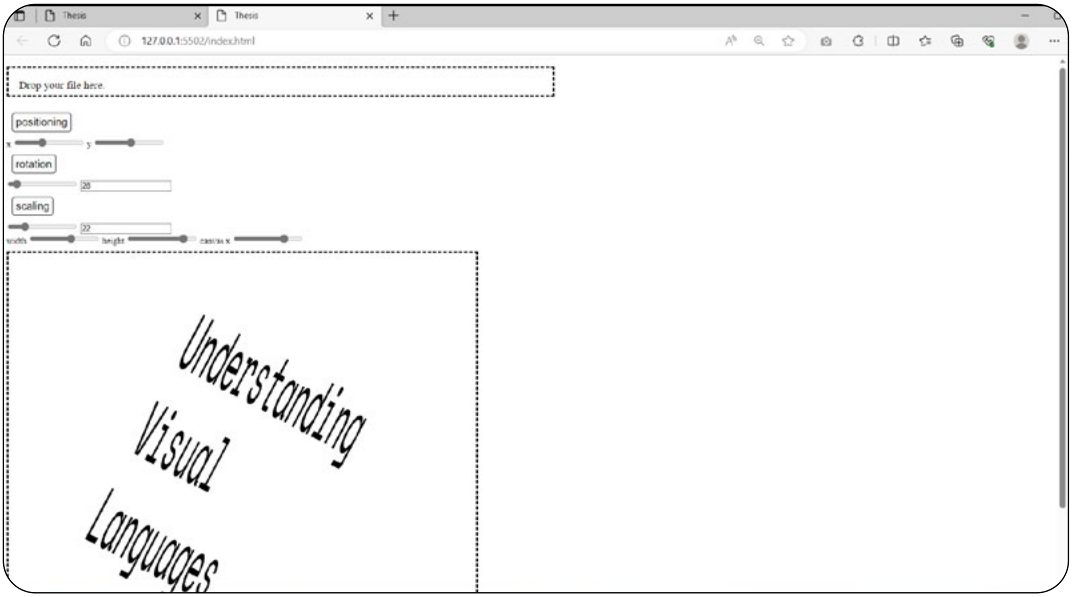
This appendix contains the development of the tool with images of the process



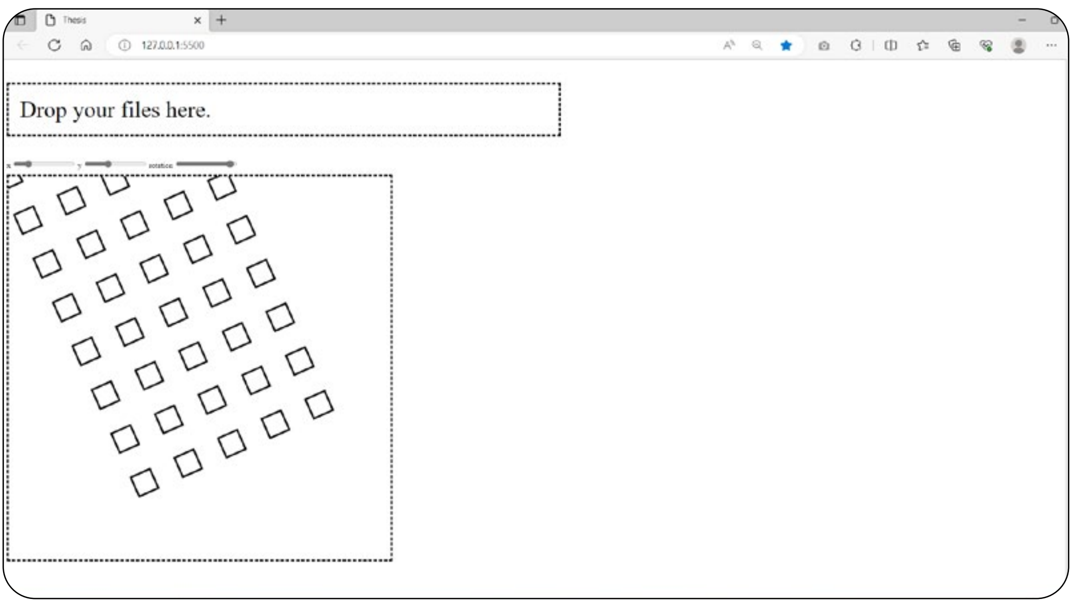
first experiments creating shapes



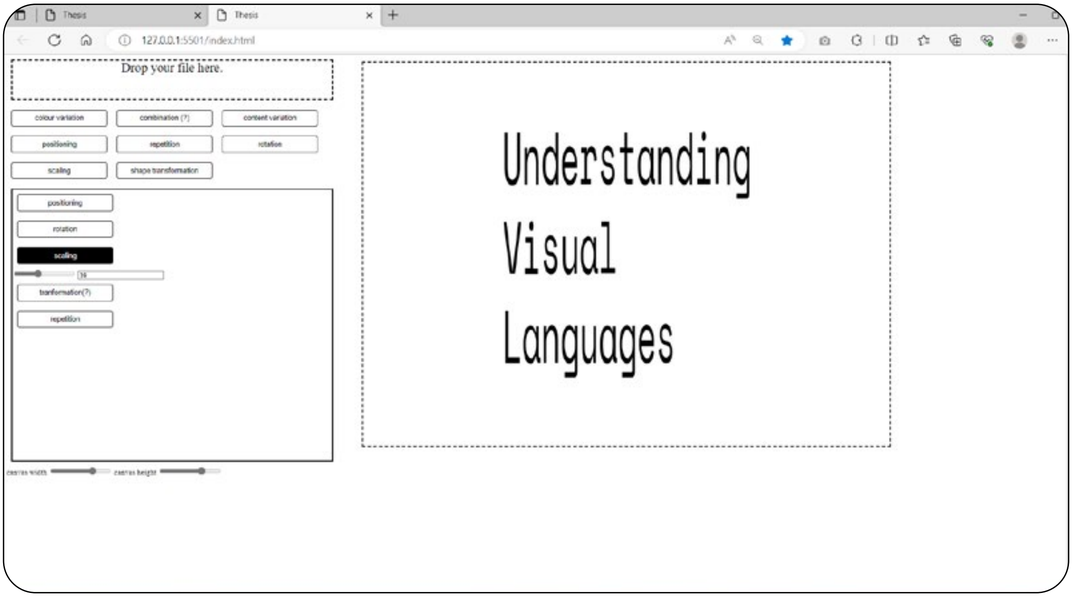
uploading SVGs as images and manipulating them using p5.js

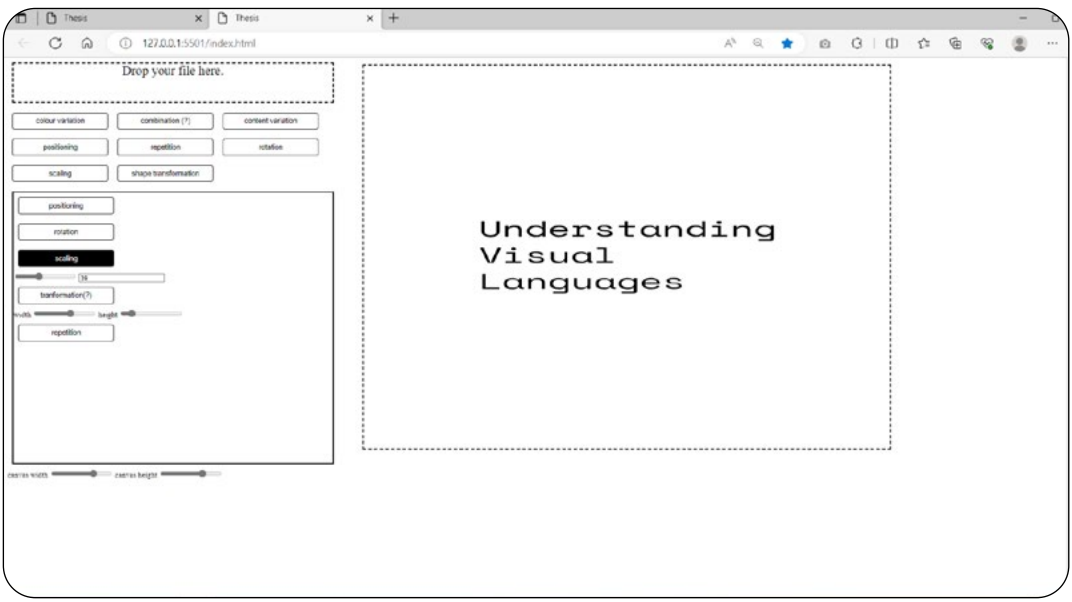


allowing any SVG to be uploaded using drop files

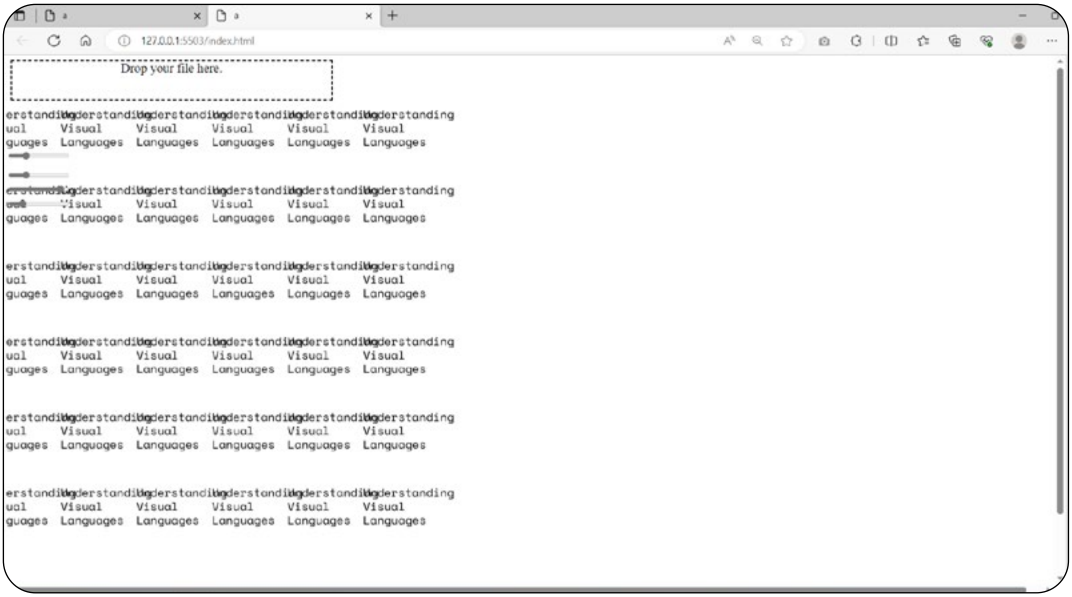


repetition experiments

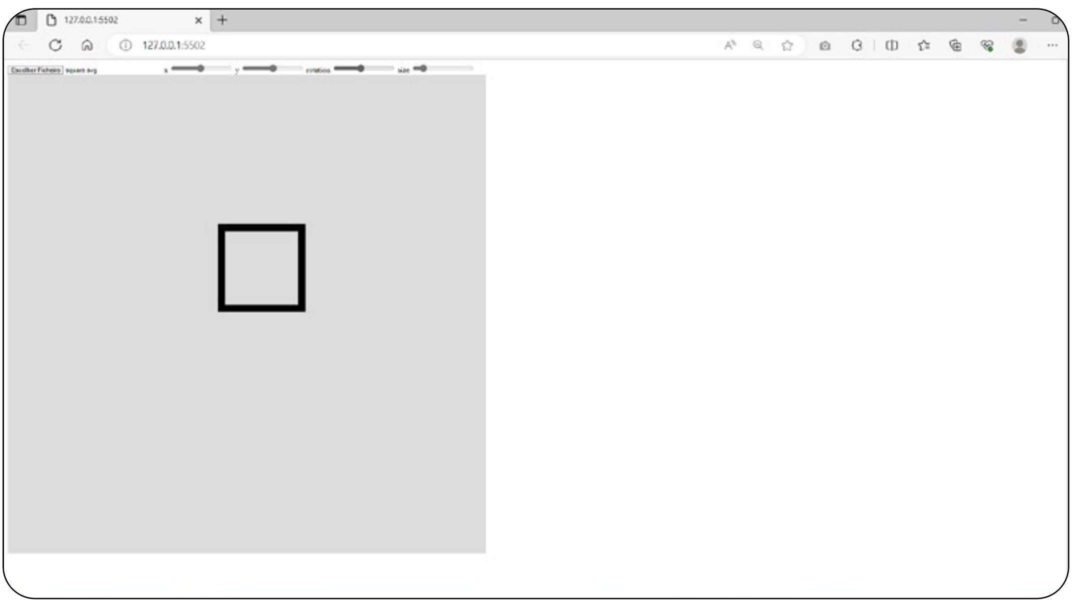




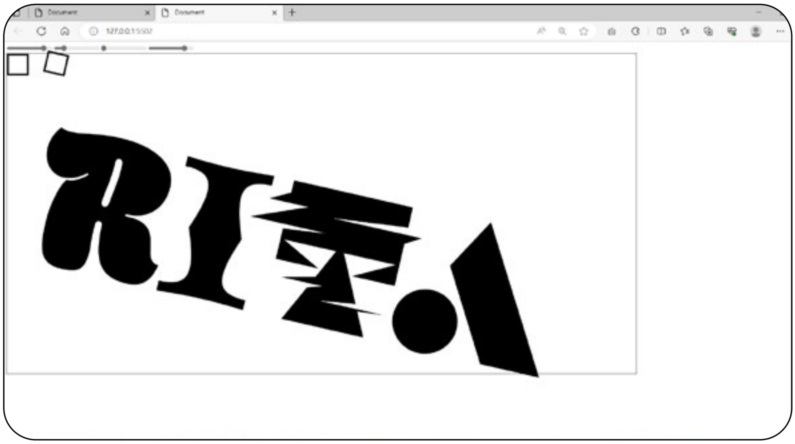
manipulating a SVG with variation mechanisms

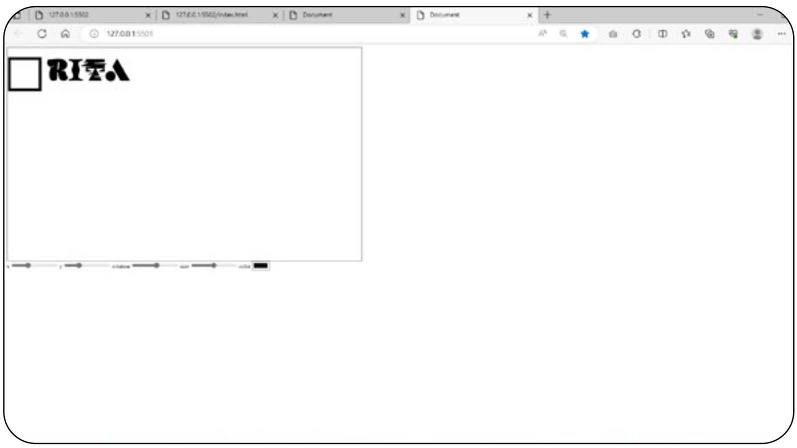


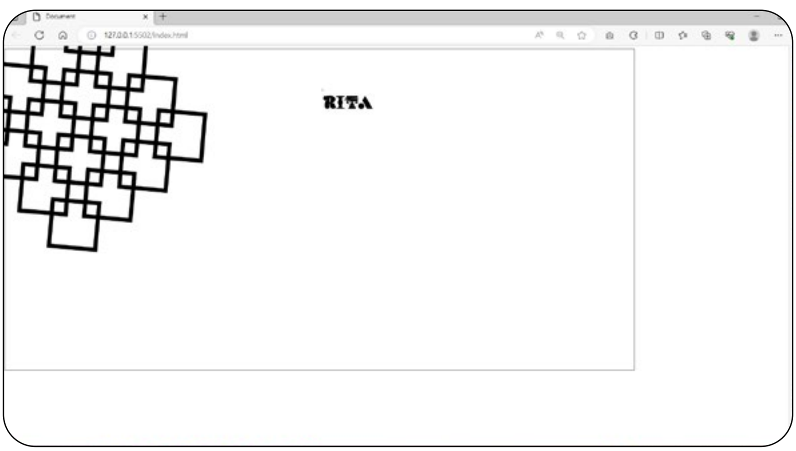
drop elements, repeating them and manipulating definitions
(number of repetitions and offset)



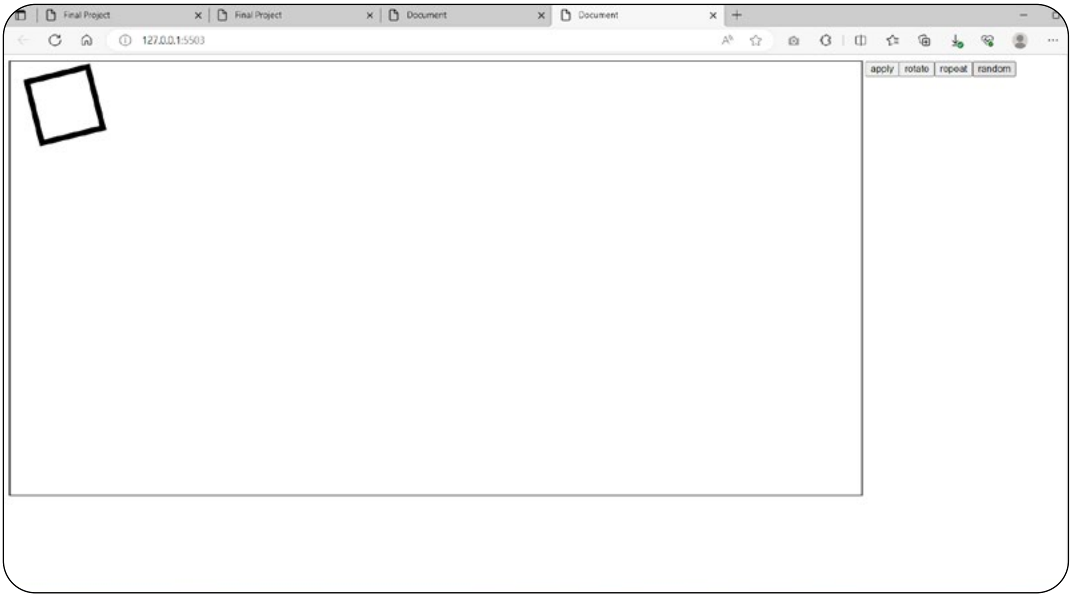
rendering SVG as vectorial elements in p5.js gives errors in some functions



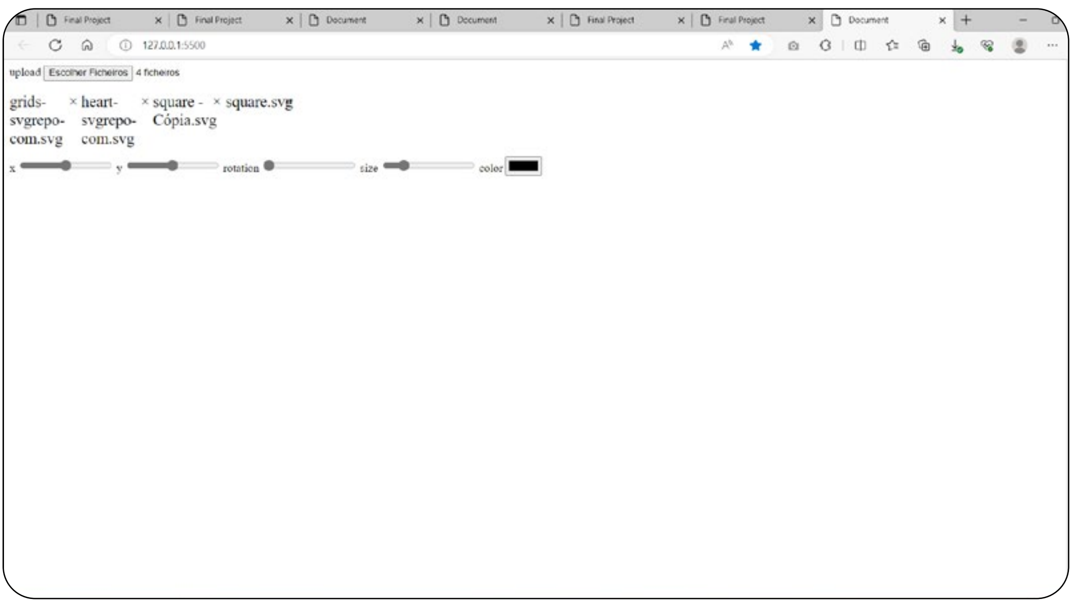




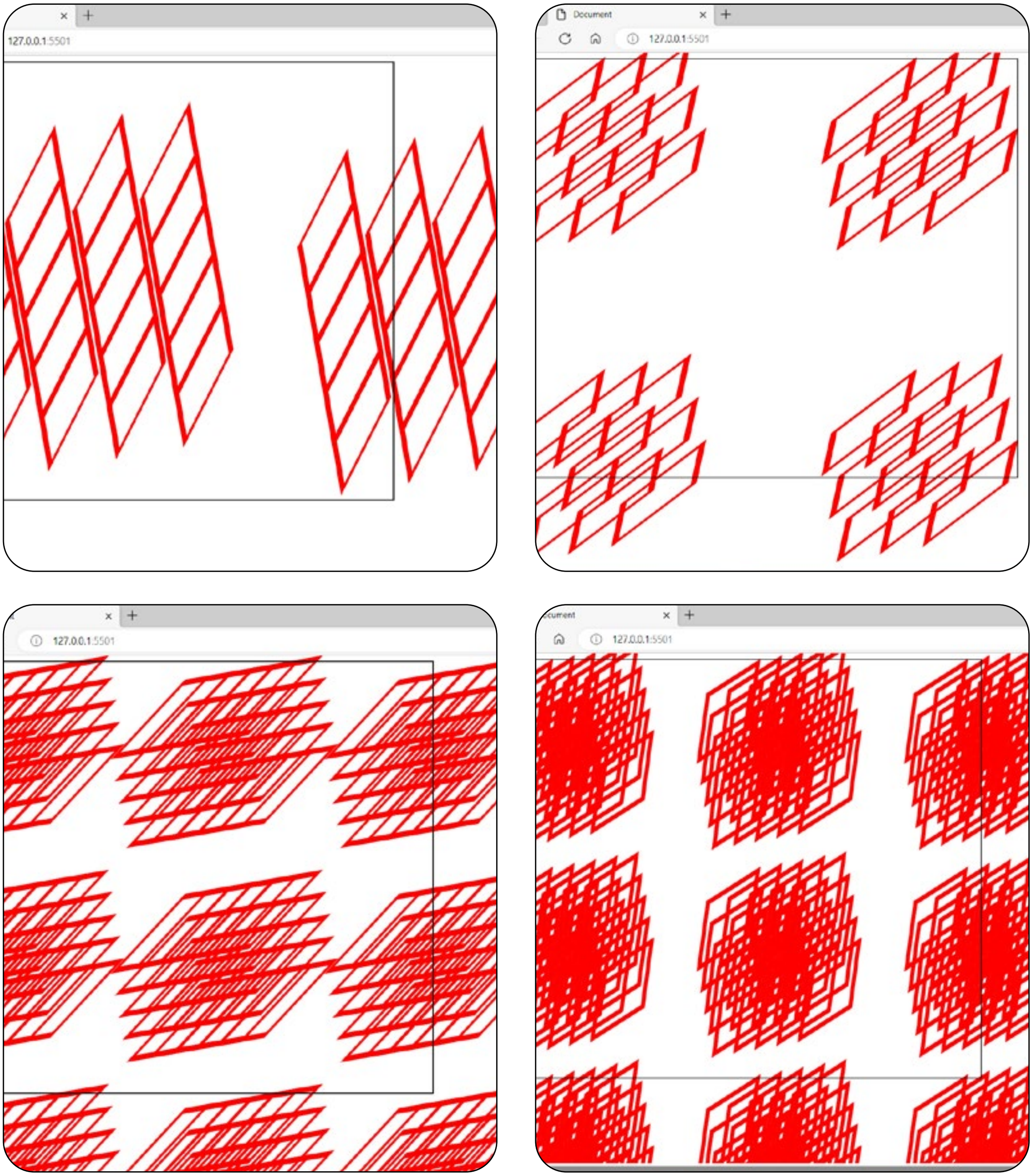
first experiments with SVGs and JavaScript



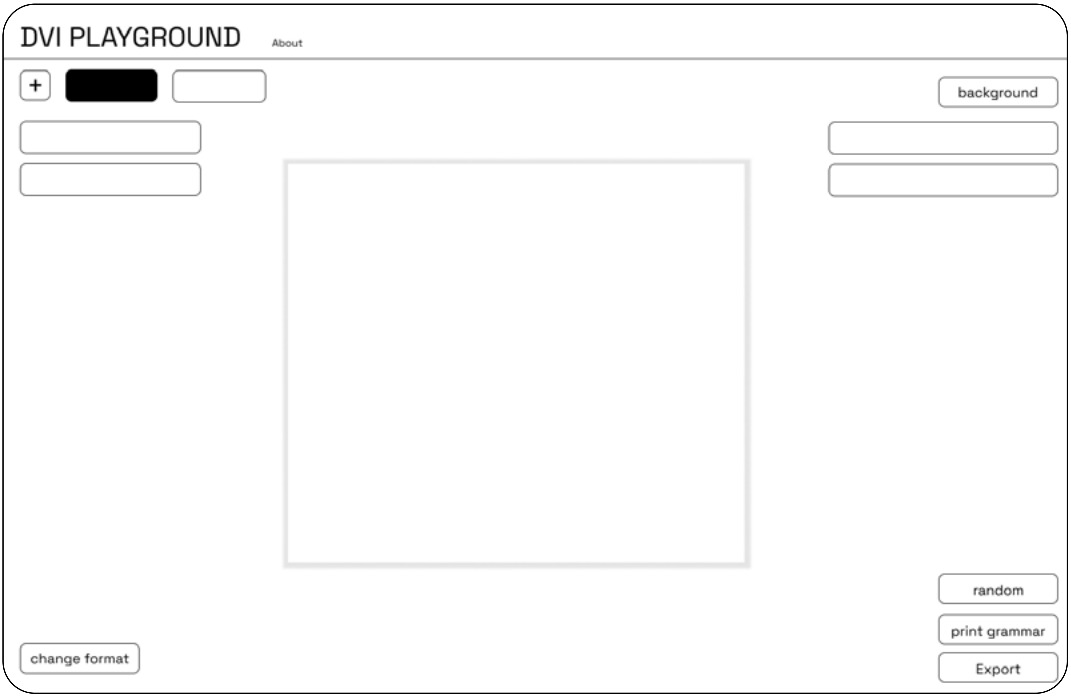
randomising variation mechanisms with buttons

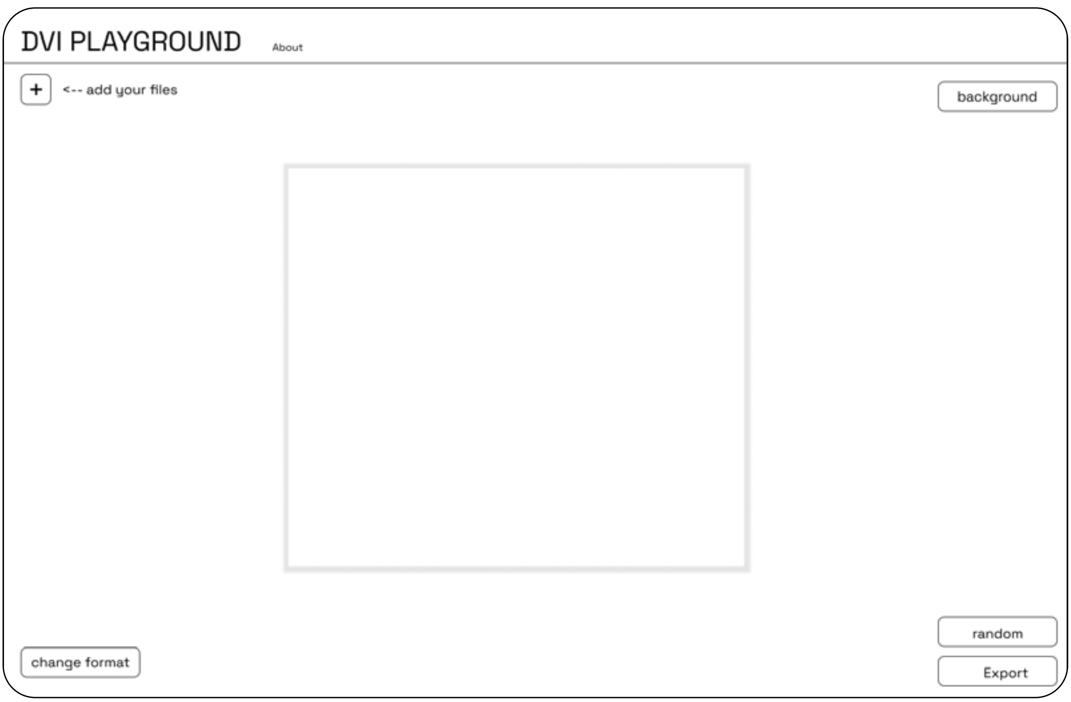


displaying multiple SVG files

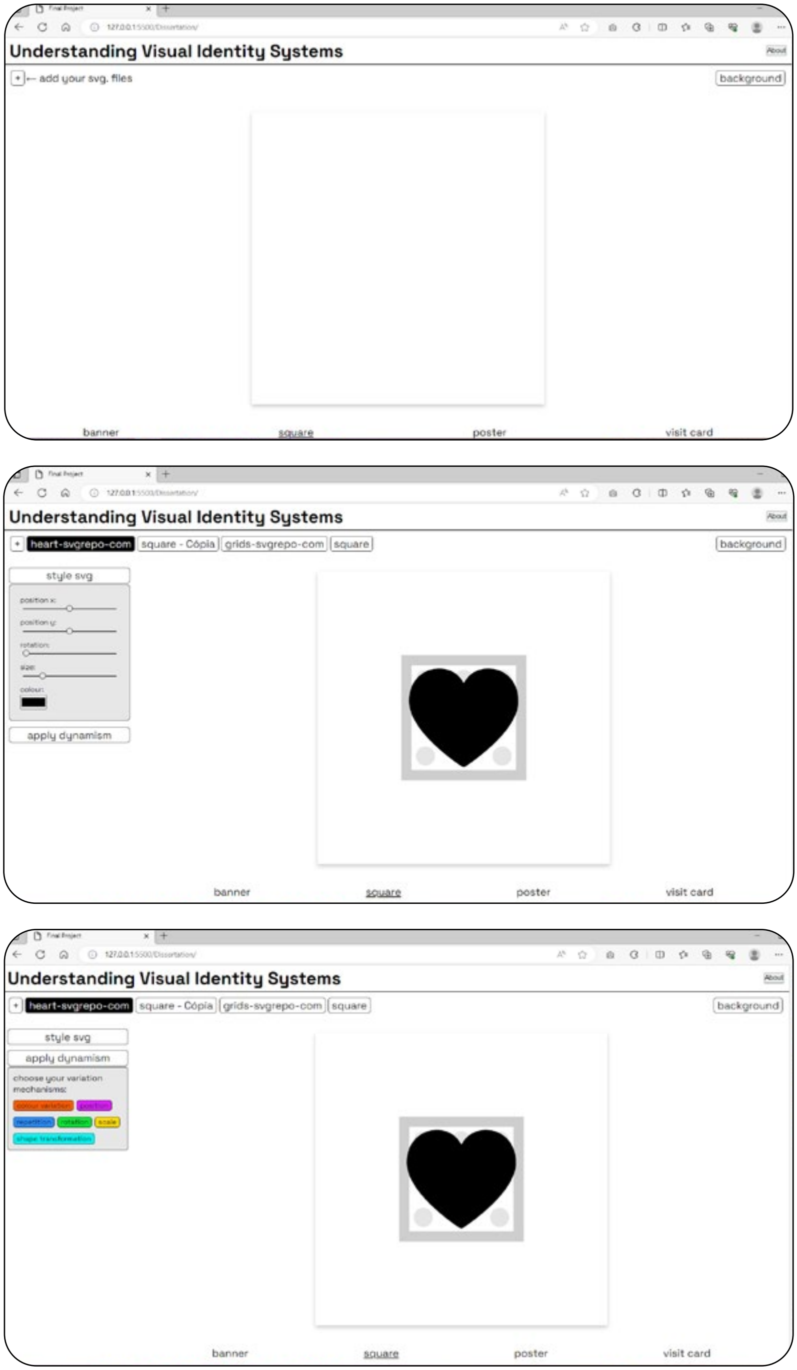


testing CSS transformations and variation mechanisms potential;
in this phase we were applying more than once the same mechanism (repeating the repetition)

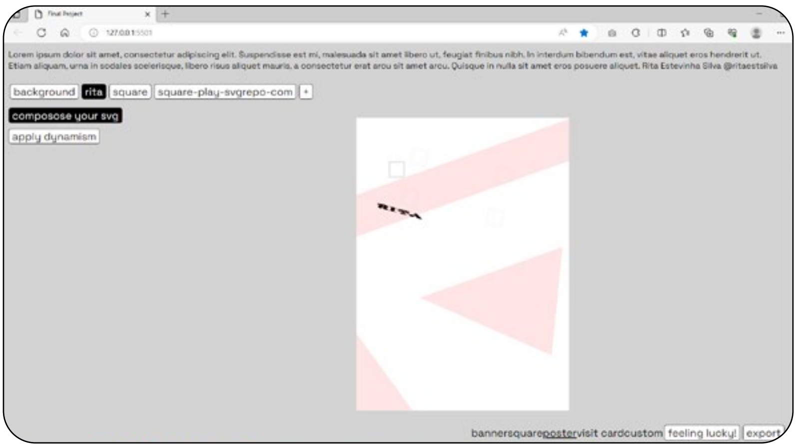


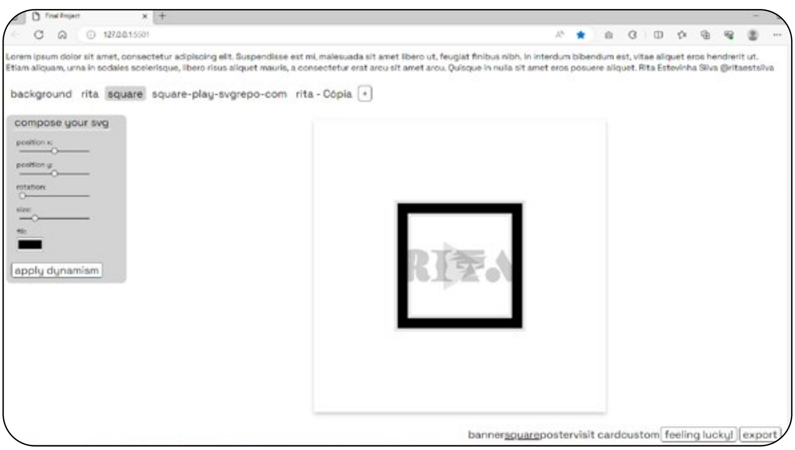


wireframes in Figma

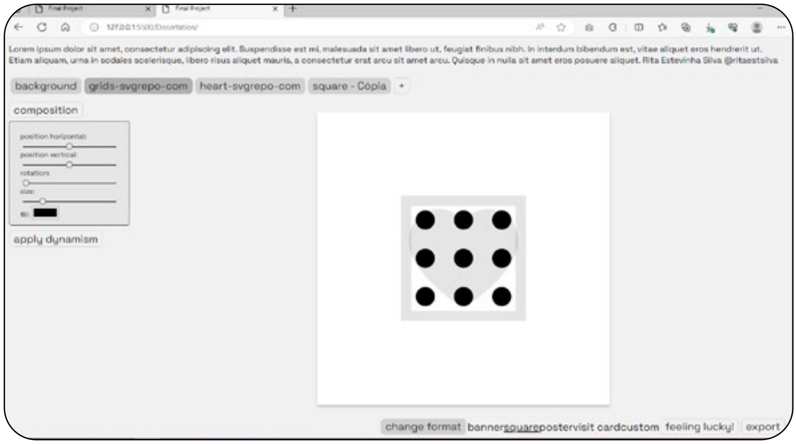


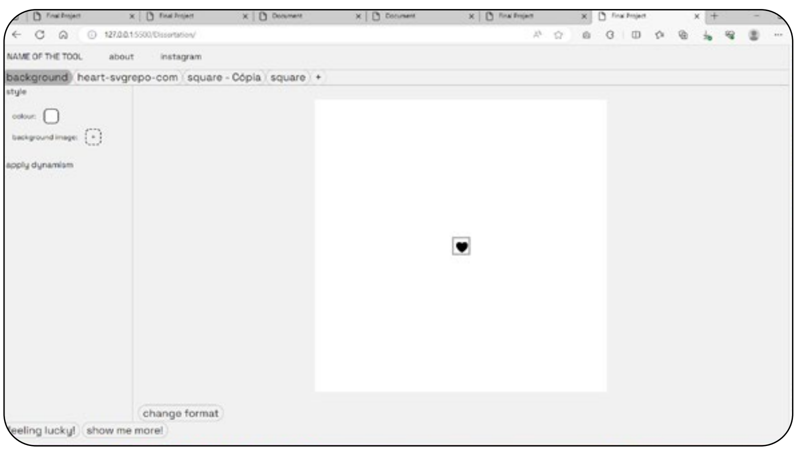
initial interface



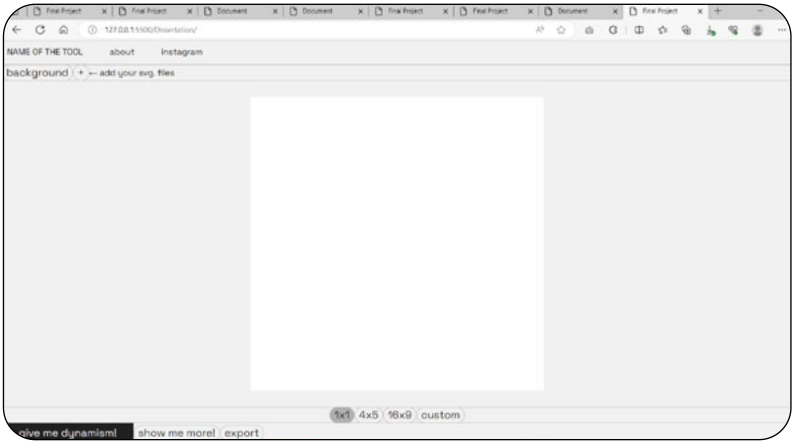


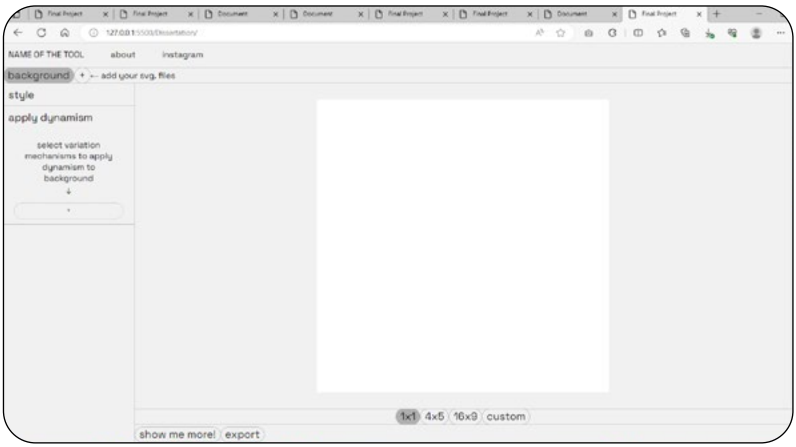
interface experiments





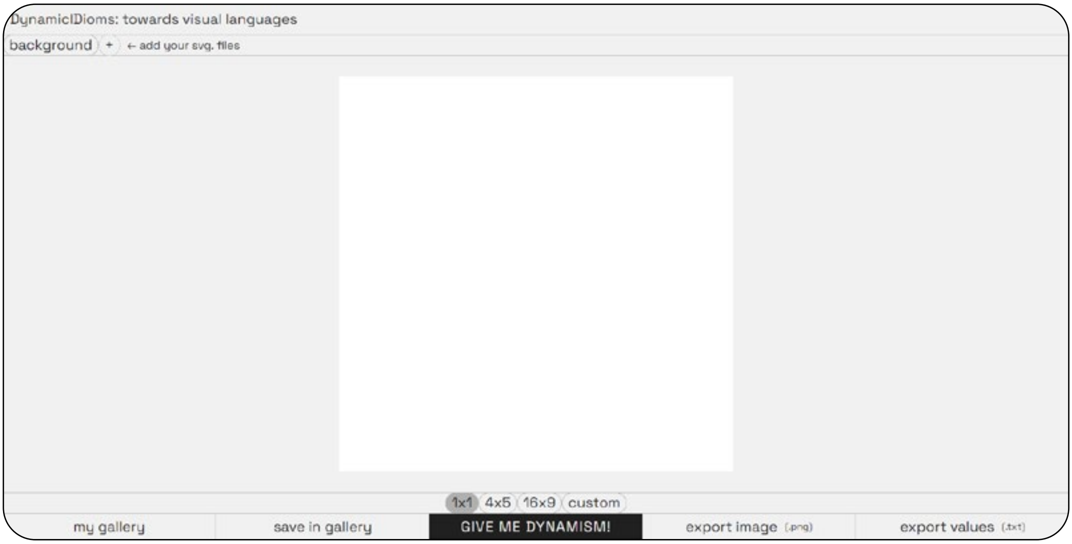
interface experiments

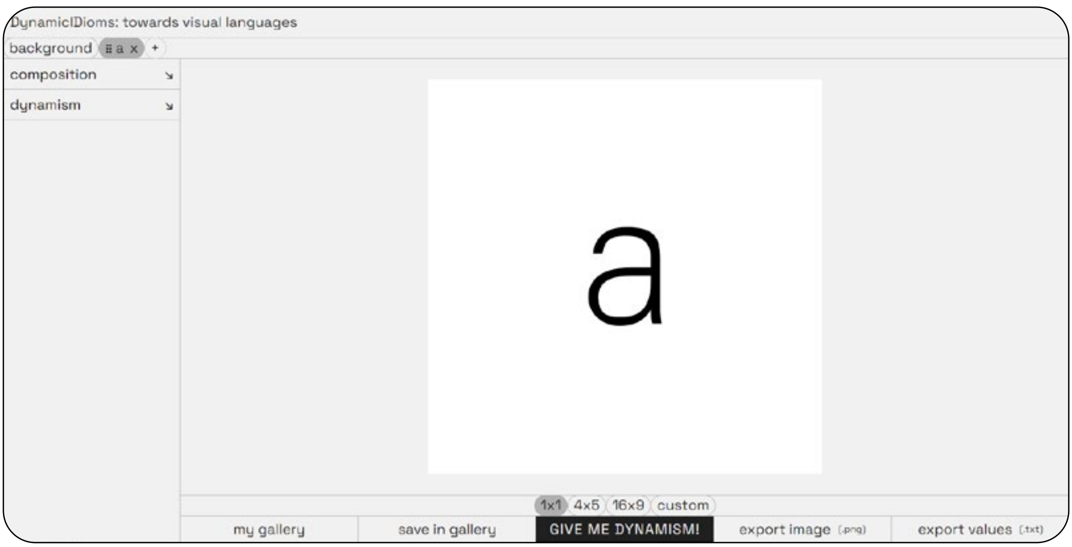




interface final experiments

gallery


DynamicIDioms: towards visual languages
background
+
← add your svg. files
1x1
4x5
16x9
custom
my gallery
save in gallery
GIVE ME DYNAMISM!
export image (.png)
export values (.txt)


DynamicIDioms: towards visual languages
background
a ×
+
composition
dynamism
a
1x1
4x5
16x9
custom
my gallery
save in gallery
GIVE ME DYNAMISM!
export image (.png)
export values (.txt)

Final Tool