

# CREATIVE CODING EXPANDING VISUAL POSSIBILITIES IN GRAPHIC DESIGN

**Maria Machado e Costa**  
Orientação **José Bártolo**  
Co-Orientação **Rafael Gonçalves**

**Mestrado em Design de Comunicação**  
Escola Superior de Artes e Design

FEVEREIRO 2023



# CREATIVE CODING EXPANDING VISUAL POSSIBILITIES IN GRAPHIC DESIGN

**Maria Machado e Costa**  
Orientação **José Bártolo**  
Co-Orientação **Rafael Gonçalves**

**Mestrado em Design de Comunicação**  
Escola Superior de Artes e Design

FEVEREIRO 2023



# RESUMO

(PT)

O presente projeto foi desenvolvido no âmbito do Mestrado em Design de Comunicação da Escola Superior de Arte e Design e tem como objectivo o desenvolvimento de uma ferramenta que possibilite a criação e exploração de propostas visuais, através da programação como parte integrante e estruturante do processo criativo, no campo do Design Gráfico, apresentando novas formas de representação e diferentes soluções visuais.

O resultado final é apresentado num suporte digital, onde é possível aceder ao mesmo em computador, bem como em telemóvel. A ambição do projeto é instigar e promover uma procura pela integração do código como ferramenta para o design gráfico, bem como possibilitar resultados inesperados e assentes no conceito da aleatoriedade.

Por efeito da evolução tecnológica, a prática do design gráfico expandiu-se e aproximou-se de outras fronteiras disciplinares, das engenharias às artes. A exploração de novos métodos e processos que unem o design e a tecnologia, criam espaço para diferentes soluções visuais e novas formas de representação. Enquanto designers devemos olhar de forma irreverente e crítica para as tecnologias que moldam o nosso dia-a-dia. Não só para o que se encontra visível à superfície mas também perceber a estrutura que molda exatamente essa imagem que é apresentada à nossa frente. Esta metodologia e pensamento estrutural, que aqui iremos explorar, intitula-se de Creative Coding.

## Palavras-chave

Aleatoriedade  
Creative Coding  
Design Gráfico  
Generatividade  
Parametrização  
Serindipidade

# ABSTRACT

(EN)

This project was developed within the scope of the Master in Communication Design at the Escola Superior de Arte e Design and aims to develop a tool that enables the creation and exploration of visual proposals, through coding as an integral and structuring part of the creative process, in the field of Graphic Design, presenting new forms of representation and different visual solutions.

The final result is presented in a digital format, where it is possible to access it on a computer, as well as on a mobile phone. The project's ambition is to instigate and promote a search for code integration as a tool for graphic design, as well as to enable unexpected results based on the concept of randomness.

As a result of technological evolution, the practice of graphic design has expanded and approached other disciplinary frontiers, from engineering to the arts. The exploration of new methods and processes that unite design and technology create space for different visual solutions and new forms of representation. As designers, we must look irreverently and critically at the technologies that shape our everyday lives. Not only for what is visible on the surface but also to understand the structure that shapes this image that is presented in front of us. This methodology and structural thinking, which we will explore here, is called Creative Coding.

## Keywords

Creative Coding  
Generative  
Graphic Design  
Parameterization  
Random  
Serendipity

# AGRADECIMENTOS

Qualquer projecto digno de seu nome dificilmente é feito isoladamente, e este não é excepção. Todos aqueles que contribuíram directa ou indirectamente para a realização deste projecto têm a minha maior gratidão. Contudo, pela importância dos seus contributos, não posso deixar de referenciar e agradecer particularmente:

Ao Professor José Bártolo, pela orientação do projeto, disponibilidade demonstrada e partilha de saber.

Ao Professor Rafael Gonçalves, pela sua co-orientação do projecto, por todo o conhecimento que me transmitiu ao longo da sua realização e pela sua disponibilidade que desde o início transcenderam em muito o papel de co-orientador. É enorme a gratidão para com ele.

Aos meus pais e irmãos pelo apoio e carinho incondicional ao longo de todo o meu percurso académico.

À Maria e à Inês, pelo apoio e motivação constantes, e por me acompanharem desde sempre. Esta conquista é tanto minha como delas, tal como todas as que já passaram e as que hão-de vir.

E à minha avó, a quem devo muito de quem sou e como sou hoje.  
Faltou-lhe o tempo mas nunca a vontade de me ver concluir esta etapa.

# INDEX

|                                      |  |                                       |                                    |
|--------------------------------------|--|---------------------------------------|------------------------------------|
| <b>1. Introdução</b>                 | <b>15</b>  | <b>4. Projeto</b>                     | <b>59</b>                          |
| <b>2. Contextualização Histórica</b> | <b>21</b>  | 4.1                                   | Introdução da Proposta 61          |
| 2.1                                  | Do analógico ao digital 23                                 | 4.2                                   | Contributo 61                      |
| 2.2                                  | Design Generativo 27                                       | 4.3                                   | Processo 62                        |
| 2.2.1                                | Primeiros Relatos Generativos 30                           | 4.3.1                                 | Referências 63                     |
| 2.3                                  | Creative Coding 36   | 4.3.2                                 | Desenvolvimento dos Componentes 68 |
| 2.4                                  | Creative Coding, Design Generativo e Design Paramétrico 37 | 4.3.3                                 | Prototipagem 98                    |
| <b>3. Estado da Arte</b>             | <b>41</b>  | 4.3.4                                 | Solução 150                        |
| 3.1                                  | Design Generativo e Creative Coding Hoje 43                | 4.3.5                                 | Resultados 170                     |
| 3.2                                  | Ferramentas 53   | <b>5. Conclusão</b>                   | <b>189</b>                         |
| 3.2.1                                | Processing 54  | <b>6. Bibliografia e Webliografia</b> | <b>195</b>                         |
| 3.2.2                                | p5.js 54   | <b>7. Anexos</b>                      | <b>201</b>                         |
| 3.2.3                                | Outros Softwares/ Linguagens de Programação 55             |                                       | Glossário 202                      |
|                                      |  |                                       | Entrevista de Philip Galanter 206  |

# (1) INTRODUÇÃO

Technology is the physical and material ways of making and using things (or performing an effective action on nature or others) in their culturally meaningful social, political, and economic contexts. Technology has aspects that are human, material, and immaterial

(Rodríguez-Alegria E. 2008)

Estamos, permanentemente, a confrontar-nos com mudanças. Quando este projeto de Mestrado começou a ser pensado, o mundo alterou-se subitamente na sequência de uma pandemia. As alterações na forma de estudarmos, trabalharmos e nos relacionarmos com os outros foi crescentemente mediada pela tecnologia, suscitando, de novo, um questionamento crítico sobre como usamos as ferramentas e meios tecnológicos, como eles nos afetam, limitam ou expandem.

A prática do design gráfico foi, desde sempre, indissociável do uso de meios, equipamentos e ferramentas tecnológicas. O desenvolvimento da computação e dos media digitais colocou novas possibilidades ao design, bem como novos desafios, suscitando questões críticas sobre a própria ontologia do design e a forma como usamos, crítica e criativamente, os inúmeros recursos de hardware e software que temos à nossa disposição. Também por efeito da evolução tecnológica, a prática do design gráfico expandiu-se e aproximou-se de outras fronteiras disciplinares, das engenharias às artes. Enquanto designers devemos olhar de forma irreverente e crítica para as tecnologias que moldam o nosso dia-a-dia. Não só para o que se encontra visível à superfície mas também para perceber a estrutura que molda exatamente essa imagem que é apresentada à nossa frente. Esta metodologia e pensamento estrutural, que aqui iremos explorar, intitulamos de Creative Coding.

No âmbito do programa de estudos de 2º ciclo em Design de Comunicação, e através da relação dos campos de conhecimento do Design Generativo, Design Paramétrico e Creative Coding com a área do Design Gráfico, a presente proposta intitulada Creative Coding: Expanding visual possibilities in Graphic Design, teve como principais objetivos:

Contextualizar a evolução histórica do design digital e do creative coding, através de revisão de literatura e análise de obras dos principais artistas e designers, bem como da evolução das ferramentas e softwares utilizados pelos mesmos no desenvolvimento de obras/projetos generativos. Contribuir para a definição e estabilização dos principais conceitos dentro do universo do design, programação e creative coding, valorizando o desenvolvimento de uma literacia de design (computacional);

Explorar aplicações e possibilidades visuais, através da programação como parte integrante e estruturante do processo criativo, no campo do design gráfico, com a preocupação da componente teórica e da componente prática do trabalho se auxiliarem para uma clarificação mútua. Com o foco e exploração da metodologia projectual no processo de design orientado para a solução, este projeto e investigação procura trabalhar uma possibilidade visual que vai surgindo mediante as possibilidades que a forma/ferramenta nos permite criar. Não existe uma predefinição do objecto visual mas sim uma descoberta ao longo da sua criação.

É através do ato de observação e duma abordagem exploratória aliada ao pensamento criativo e estrutura por detrás das ações que irão surgir ideias e resultados imprevisíveis e aleatórios.

No desenvolvimento desta investigação de Mestrado, as noções de “prática exploratória” e de “imprevisibilidade” foram surgindo como conceitos centrais. Como, introdutoriamente, procuraremos mostrar, elas surgem indissociáveis quer dos projetos pioneiros de arte e design generativo, desenvolvidos ao longo da segunda metade do século XX, quer em projetos contemporâneos que exploram novas possibilidades de creative coding.

Como veremos, estes conceitos são compreendidos e explorados metodologicamente, caracterizando uma relevante dimensão processual dos projetos. É referido no **Capítulo 2 Contextualização Histórica** que o desenvolvimento do Design Generativo está diretamente associado a um contexto histórico, artístico e tecnológico na segunda metade do século XX, no qual o desenvolvimento dos novos media e os cruzamentos entre a prática artística, a investigação científica e a aplicação de novas tecnologias, nomeadamente computacionais, expandiu a compreensão e as formas de materialização artística. Com o avanço das tecnologias identificou-se uma pluralidade de definições no que diz respeito à arte e ao design e à sua aliança com as tecnologias, que se pretendeu aqui apresentar. No **Capítulo 3 Estado da Arte** procurou-se apoiar estes conceitos numa listagem de projectos, autores, exposições e publicações, de forma a clarificar e a distinguir o que é o Design Generativo e o Creative Coding. Desde o início que não existe uma predefinição do objecto visual mas sim uma prática exploratória de diferentes funcionalidades que se pretendia explorar e que visa dotar a ferramenta de uma dimensão colaborativa e imprevisível como é referido no **Capítulo 4 Projeto**.

Através da informação que acrescentamos nestes capítulos não se procurou aprofundar exaustivamente a dimensão histórica e teórica mas possibilitar ao leitor um melhor conhecimento do projeto e as devidas referências.

# (2) CONTEXTUALIZAÇÃO HISTÓRICA

## (2.1) Do analógico ao digital

A evolução tecnológica relacionada com a arte digital e com grafismos generativos iniciou-se quando foi apresentado o primeiro computador, o ENIAC (Electrical Numerical Integrator and Computer). Este foi o primeiro computador digital eletrônico no mundo, criado em 1946 pelos cientistas norte-americanos John Eckert e John Mauchly, da Electronic Control Company. Foi desenvolvido em 1943 durante a II Guerra Mundial com o intuito de calcular as trajetórias táticas e calcular tabelas de artilharia balística, mas só se tornou operacional no final da Guerra. A sua primeira utilização foi no cálculo para uma bomba de hidrogénio. Todos os trabalhos artísticos apresentados até então eram, por isso, desenvolvidos de forma analógica. Os primeiros projetos generativos, antecedentes à criação dos primeiros computadores, eram desenvolvidos com ferramentas analógicas eletrónicas onde se pré-estabeleciam parâmetros de modo a responder de forma generativa. O osciloscópio, foi uma ferramenta muito utilizada por artistas da época. É um instrumento de medida de sinais elétricos que apresenta gráficos a duas dimensões de um ou mais sinais elétricos. Este instrumento foi utilizado, nomeadamente no projeto *Oscillons* de Ben Laposky (1914-2000), onde o artista captura o som através do osciloscópio e representa através do mesmo as ondas sonoras captadas. Em 1947, a revista americana *Popular Science* publicou um artigo onde nomeia o osciloscópio como uma possível ferramenta no design, como fonte de inspiração para a execução de padrões para uma via têxtil. Mais tarde, por volta dos anos 60, começaram a surgir os primeiros computadores individuais. No entanto, prevaleciam os mainframes, os computadores de grandes dimensões, que ficavam em salas refrigeradas e eram usados por um grupo limitado de pessoas. Apenas grandes empresas e bancos podiam investir nestas ferramentas de modo a aumentarem a eficácia de algumas tarefas internas. Nesta fase inicial os computadores existentes eram utilizados numa vertente mais técnica, para execução de tarefas e não com o intuito visual.

Foi em 1965 que foi desenvolvida a primeira série de computadores pessoais. Em 1976, Steve Jobs (1955-2011) e Steve Wozniak (1950), criaram a sua empresa mudando o rumo da informática, como podemos ver e utilizar nos dias de hoje: a Apple Inc. No ano seguinte, 1977, foi lançado o primeiro microcomputador, o Apple II. A partir da década de 1980, progressivamente, assistimos ao desenvolvimento de computadores mais pequenos, portáteis, com mais capacidade de armazenamento, com um design mais ousado, tornando-se hoje numa das ferramentas mais utilizadas no desenvolvimento do design generativo.

Paralelamente na vertente artística, a arte e o design são forçosamente obrigados a alterar os métodos que utilizam de forma a acompanharem esta evolução e apresentarem diferentes soluções para o mercado, bem como diferentes conceitos. A definição da prática artística bem como a destes conceitos reside no desafio das mudanças constantes do dia-a-dia e da evolução dos meios e das ferramentas ao dispor para a criação. Muitos designam todo o tipo de arte executada desde a década de 1950 até à atualidade como arte contemporânea, dado que a arte reflete a sua era, era esta de expansão e evolução dos meios e ferramentas digitais. No entanto, mesmo que ténue, existem delimitações duns termos para os outros e aqui residirá uma tentativa de definição das mesmas.

Arte Digital - As origens da arte digital remontam às experiências da década de 1950, em particular, a Manfred Frank e Ben Laposky que criaram, por meio de um osciloscópio, uma função matemática capaz de gerar uma representação gráfica, passível de ser distorcida pela variação do comprimento de onda dos raios de luz no tubo de raios catódicos. Arte Digital descreve todas as obras que usem qualquer forma de tecnologia digital como parte do seu processo de criação ou apresentação.

Foi no final da década de 1940 e início de 1950, com o que aparentava ser o fim do Expressionismo Abstrato, que surgiram novos movimentos de vanguarda no campo artístico como Neo-Dada, Novo Realismo, Arte Cinética, Pop Art, e Arte Computacional, englobando todos estes movimentos no conceito/era de New Media Art. De forma a clarificar os conceitos acima referidos, deixamos uma definição que se pretende que seja o mais clara possível de cada um deles.

New Media Art descreve os projetos que fazem uso de meios tecnológicos emergentes e que têm uma preocupação cultural, política e social, tirando proveito da estética que as ferramentas tecnológicas possibilitam. New Media Art refere-se a projetos que utilizam meios digitais, que são gerados por algoritmos, que são dinâmicos, participatórios, colaborativos, variáveis e generativos. New Media Art insere-se na década de 1990, apesar de ter surgido em 1950, tal como o termo arte digital que surgiu na mesma década (1950), mas que apenas se viu estabelecido em 1980.

Por sua vez, a Arte digital encapsula toda a obra que use qualquer forma de tecnologia digital como parte do seu processo de criação ou apresentação, desde computadores, telemóveis, televisão ou outro qualquer tipologia de **outputs**, **inputs** ou **softwares**. Apesar de algumas reticências relativamente ao uso de computador para a execução de certas instruções em prol da construção de uma obra de arte, finalmente este mesmo instrumento foi aceite como um parceiro de criação.

In the computer, man has created not just an inanimate tool but an intellectual and active creative partner that, when fully exploited, could be used to produce wholly new art forms and possibly new aesthetic experiences.

(Noll M., 1982)

Assim, em 1965 ocorreram as primeiras exposições de arte computacional nos Estados Unidos e na Alemanha, realizadas apenas por cientistas, pois eram movidos pela curiosidade na visualização de sinais sonoros, e por serem dos poucos a compreender as linguagens de programação. Os pioneiros da arte computacional foram conduzidos pela curiosidade na junção da tecnologia com diferentes áreas, que eram até então improváveis. Devido à falta de viabilidade comercial destas experiências, isso permitiu aos artistas terem um pouco mais de liberdade criativa, dado que poderiam eles mesmos definir os seus objectivos e interesses pessoais. As primeiras obras de arte computacional eram lineares e geométricas devido à falta de conhecimento e desenvolvimento dos mesmos, bem como o grande interesse na criação de imagens abstratas, ao invés de realistas, e pela disciplina de programação ser recente e não ser ensinada aos artistas.

Para um computador executar qualquer tipo de regras/ordens era necessário uma linguagem, linguagem essa que os artistas desconheciam, daí os mesmos procurarem ajuda junto de cientistas/programadores. Tal limitação, fez com que este tipo de arte surgisse num meio restrito mas igualmente colaborativo.

Um dos pioneiros na arte computacional foi Robert Mallary (1917-1997), um escultor expressionista abstrato norte-americano, conhecido por ter criado uma das primeiras esculturas do mundo geradas por computador, *Quad 1*, que foi exibida em 1968 na exposição *Cybernetic Serendipity* no Institute of Contemporary Arts em Londres.

**Input** - O que é inserido, recebido ou operado por qualquer processo ou sistema. Um lugar onde, ou um dispositivo através do qual, energia ou informação entra num determinado sistema.

**Output** - A quantidade de algo produzido por uma pessoa, máquina, ou indústria. Um lugar onde a informação sai de um sistema.

**Software** - Conjunto de programas de computador, procedimentos e documentos associados no que diz respeito à operação de um sistema de processamento de dados.

O próprio refere-se ao computador como uma ferramenta para melhorar de imediato o poder criativo e a produtividade do artista através da aceleração do processo criativo e da disponibilização de um conjunto de opções de design que de outra forma não lhe ocorreriam. (Mallary R., 1986)

Neste mesmo ano, criou-se o EAT (Experiências em Arte e Tecnologia), formado por Billy Klüver (1927-2004) e Fred Waldhauer (1927-1993), engenheiros, bem como, por Robert Rauschenberg (1925-2008) e Robert Whitman (1935), artistas. Este grupo pretendia promover a colaboração entre a arte e a tecnologia. Juntos criaram uma série de instalações com sistemas eletrónicos, projeção de vídeo e som. Continuaram a desenvolver projetos nas décadas de 1970 e 1980, com um cariz social, e por isso menos ambicioso, como foi o caso do projeto *Children Communication* (1972) que permitiu que as crianças de Nova Iorque comunicassem por telefone, fax ou telex (um sistema internacional de comunicações escritas). Com o desenvolvimento progressivo dos computadores, bem como da facilitação da sua utilização devido ao crescimento de projetos utilizando meios digitais, na década de 1980, tornou-se mais difícil concretizar uma definição do conceito de arte digital e arte computacional. Sabemos que podemos caracterizar esta arte pelo uso da aleatoriedade, mas também da parametrização, bem como da combinação com diferentes áreas como a matemática e a ciência. A aleatoriedade permitiu aos artistas que os resultados saíssem da sua previsibilidade, mesmo que moderando o valor da aleatoriedade num intervalo de tempo à sua escolha. Este conceito da aleatoriedade surge aquando do lançamento dos primeiros computadores pessoais, pela Apple Corporation, em 1980, bem como dos primeiros programas computacionais de desenho, promovendo a criação de imagens através do computador para indivíduos que desconheciam as linguagens de programação.

O primeiro software de desenho, *Sketchpad* surgiu em 1963 possibilitando a criação de desenhos técnicos que antes eram feitos à mão, para passarem a ser feitos em computador. Após esta data, em 1980 começaram a surgir outros programas com as mesmas características, como era o caso do Catia e do AutoCad. A popularização destes programas, bem como a crescente liberdade de criação de imagens, levou os artistas e os designers a afastarem-se da programação, e ao invés de aprenderem a programar o computador para gerar arte, focaram-se na utilização do software para a criação e manipulação da mesma. O computador passou a ser uma ferramenta usada no processo de trabalho dos artistas e designers, aproximando à realidade que vivemos hoje de que o futuro do design se aproximasse mais da criação de ferramentas ao invés da criação do design em si. A expansão dos computadores e por consequência da arte computacional fez com que em 1995 os artistas Jean-Pierre Hébert (1939-2021), Roman Verostko (1929-) e Ken Musgrave (1955-2018) sugerissem o termo **Algorithms** para definir todo o artista que criava arte através dum algoritmo.

Jean-Pierre Hébert, era um artista americano, mas de origem francesa, considerado um dos pioneiros da arte digital. Iniciou a sua prática na criação de desenhos em computador em 1970, sendo que a partir de 1984 dedicou-se à exploração de desenhos através de um algoritmo.

Em 1995 criou o termo Algorithms. Nas suas obras funde a arte com a ciência, a matemática e a poesia. Nas suas instalações (vertente projetual mais utilizada pelo autor), explora a ideia de calma e meditação através, do uso de areia, água e vento, e duma linguagem visual de abstração geométrica, devido à partilha de criação com o computador.

**Algorithms** - Termo usado para artistas digitais que criam arte algorítmica. Os co-fundadores foram Jean-Pierre Hébert e Roman Verostko. Hébert é creditado por cunhar o termo e a sua definição que está na forma do seu próprio algoritmo:

```
if (creation && object of art && algorithm && one's
own algorithm) {
    return * an algorist * }
else {
    return * not an algorist *
}
```

Junto com Jean-Pierre Hébert, Roman Verostko é também um dos fundadores do termo Algorithms. Nascido em Minneapolis, iniciou a sua vida como sacerdote, tendo deixado a vida monástica em 1968. Deslumbrado com o poder do uso do algoritmo começou a explorar o mundo da programação, exibindo a sua primeira obra algorítmica em 1982 intitulada de *Hand of Chance* (exposição onde se geravam formas em tempo real através de um algoritmo). Hoje, Roman Verostko mantém um estúdio de desenvolvimento experimental de algoritmos em Minneapolis.

Ken Musgrave foi um professor e artista norte-americano focado no desenvolvimento e criação de imagens fractais. Foi CEO da Padromeda, empresa de software. A arte fractal é um exemplo de arte algorítmica, inspirada na matemática e na natureza. No entanto, aqui o algoritmo não é concebido pelo artista. A arte fractal é uma forma de arte algorítmica criada calculando objetos fractais e representando os resultados do cálculo como imagens digitais e animações. Este conceito foi desenvolvido a partir de meados da década de 1980. O artista Kerry Mitchell afirmou no seu Manifesto Fractal Art de 1999 :

Computer(ized) Art, in the sense that the computer does all the work. The work is executed on a computer, but only at the direction of the artist. Turn a computer on and leave it alone for an hour. When you come back, no art will have been generated.

(Mitchell K., 1999)

Um fractal, é um padrão que pode ser encontrado na natureza em diferentes escalas, em organismos vivos e em ambientes, que repetidos formam padrões, geralmente geométricos que contém uma estrutura de detalhes.

O rumo ao longo da história do design e da arte algorítmica, arte computacional e arte digital, ocorre par a par com o desenvolvimento da tecnologia, no entanto esta última não passa de uma utilidade, ferramenta para atingir um fim. O algoritmo é, sim, o cerne e instrumento principal dado que é intemporal e independente do meio pelo qual é utilizado, seja computacional ou não. O algoritmo por ser uma descrição, um conjunto de regras a serem seguidas para executar determinada tarefa, torna-se elementar e intemporal. Desde que existe a automatização de tarefas ou resolução de problemas pelas tecnologias/máquinas, as mesmas são programadas com um algoritmo, que é adaptado à linguagem de programação específica a ser utilizada. Desde o final do século XX até à atualidade, devido à evolução das tecnologias, bem como ao uso de algoritmos em prol do design e da arte, surgiram novos termos/conceitos como já introduzimos, tais como design generativo, arte computacional, design computacional entre outros termos, cuja base emprega processos algorítmicos. No que diz respeito à semântica destes conceitos, o termo arte digital hoje leva-nos até à década de 1960 e à primeira experiência com arte computacional e arte cibernética exibida na ICA Londres, intitulada de Cybernetic Serendipity. Termos como “arte e tecnologia” levam-nos ainda mais para trás no tempo dos movimentos neo-vanguardas.

Atualmente, a definição de New Media Art ainda não se encontra estabilizada, sendo que o que a mesma descreve é a arte que é produzida, discutida, relacionada, criticada e visualizada num mundo de arte específico, que chamamos de New Media Art World, que irá ser moldado mediante as problemáticas ao seu redor.

Podemos considerar que a New Media Art, é melhor descrita como algo transicional, uma forma de arte híbrida, uma micro prática multidisciplinar. No fundo, hoje desenvolvemos uma comunidade que não produz arte e que, ao invés, testa e explora meios artísticos (do futuro), para o benefício de gerações futuras. Como refere Domenico Quaranta, na publicação Beyond New Media Art:

The media critic Geert Lovink, for example, devotes a whole chapter of his book Zero Comments to the “crisis of New Media Art”. The essay opens with a few unsettling questions:

«Why is new media art perceived as an obscure and self-referential subculture that is in the process of disappearing? Why is it so hard for artists that experiment with the latest technologies to be part of pop culture or ‘contemporary arts’? [...] New media art has positioned itself in between commercial demo design and museum strategies, and instead of being crushed, it has fallen into an abyss of misunderstanding.

(Quaranta D., 2013)

Podemos concluir que a definição de New Media Art, ainda se encontra em construção, pois há uma controvérsia de categorização dos projectos.

## (2.2) Design Generativo

Aleatório é a palavra mais utilizada quando falamos de Design Generativo, o que não é verdadeiramente correto. O Design Generativo está relacionado com qualquer prática de arte onde o artista utiliza um sistema como um conjunto de regras de processamento de linguagem natural.

Este sistema pode ser um programa de computador, uma máquina ou outra invenção processual, e assim, quando o mesmo executa determinada tarefa, estamos cientes de um certo grau de autonomia que contribui para o resultado final do trabalho. Este grau de autonomia é identificado na literatura por diferentes termos: aleatoriedade, acaso, caos, chance, generativo. Mas a autonomia processual serve apenas para não existir uma padrão organizado, e permitir assim alguma imprevisibilidade, Galanter torna-o mais claro afirmando que:

Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art. Whether considered from the top down or the bottom up, the defining aspect of generative art seems to be the use of an autonomous system for art making.

(Galanter, P., 2003)

Processamento de linguagem natural - O processamento de linguagem natural (PLN) usa machine learning para processar e interpretar texto e dados.

Outros termos como Arte Algorítmica - Como arte gerada por um computador, que é um subconjunto da arte generativa, a arte algorítmica como o próprio nome indica é arte gerada por um algoritmo. Iniciou-se nos anos 60, juntamente com o desenvolvimento da tecnologia computacional, onde algumas das primeiras práticas foram realizadas por uma plotter. No entanto, esta arte precede o uso dos computadores. Hoje, a arte algorítmica refere-se a obras de arte determinadas por algoritmos ou geradas por computador, ou por um sistema que executa um conjunto de regras.

Desenho Assistido por Computador - É o uso de computadores para auxiliar na criação, modificação, análise ou otimização de um projeto. Este software é bastante utilizado no setor da engenharia, na indústria automóvel, na construção naval, aeroespacial, e no design arquitetónico e industrial, devido ao seu rigor na criação de objectos/próteses 3D.

Computação Gráfica - É um campo entre a ciência da computação e o design, que é responsável por gerar imagens, na forma de pixels, desenhadas nos computadores. Dentro do processo de criação de imagens é responsável também pelo processo de texturas, renderização e iluminação desses modelos, até à exibição digital final. Com o desenvolvimento dos softwares e hardwares, a precisão, velocidade e qualidade dos renders criados tem sofrido um aumento exponencial.

O design generativo é uma forma contemporânea de criação artística, na qual a obra de arte ou produto final não estão necessariamente no centro, mas sim o processo de criação e as ideias subjacentes ao desenvolvimento desse produto final. O trabalho ou o produto é criado através de um conjunto de regras criadas pelo artista ou um programa na forma de, por exemplo, processamento linguagem natural, linguagem musical, código ou mecanismos.

É frequentemente usado e associado à arte algorítmica. A Arte algorítmica ou também conhecida como arte gerada por um computador, que é um subconjunto do design generativo, ou por um algoritmo, iniciou-se nos anos 60, juntamente com o desenvolvimento da tecnologia computacional, onde algumas das primeiras práticas foram realizadas por uma plotter. No entanto, esta arte precede o uso dos computadores.

Hoje, a arte algorítmica refere-se a obras de arte determinadas por algoritmos ou geradas por um computador, ou por um sistema que executa um conjunto de regras. Assim como as plotters foram usadas para produzir variações de linhas, o sistema da arte algorítmica executa um algoritmo para gerar peças visuais com diferentes formas, posições e cores. Esta arte acaba por exigir do artista um afastamento do controlo e autonomia, passando a arte final a ser uma colaboração entre o artista e o sistema/computador, o que ainda traz controvérsia relativamente a esta arte ser uma arte gerada por computador em oposição a uma arte assistida por computador. Dada a liberdade e autonomia do computador face às decisões visuais do produto visual, esta é ainda uma definição em aberto.

Esta arte costuma servir os artistas como meio para potenciar a aleatoriedade. Apesar do processo da mesma ser organizado e autónomo, correndo uma série de instruções por via, por exemplo, de um computador, o resultado revela um processo diferente, pois apresenta resultados imprevisíveis neles mesmos. Está implícito neste tipo de arte que o autor deve abdicar parcialmente do controlo para que o sistema possua algum grau de autonomia. Nos dias de hoje, o computador está de tal modo integrado na nossa rotina diária que se tornou indispensável tanto para uso pessoal como em inúmeras áreas e profissões, criando aqui uma dependência do mesmo, e uma cedência de autonomia pessoal.

Atualmente qualquer designer vê o computador como uma ferramenta de trabalho imprescindível devido às suas capacidades, sendo amplamente utilizado no Design Generativo. Aqui o designer constrói um processo através de um programa de computador sendo o resultado final executado pela própria ferramenta. O uso do computador reduz significativamente o tempo para criar composições complexas e repetitivas.

Tem havido também uma crescente criação de ferramentas cada vez mais autónomas, como é o caso do ChatGPT. Estas ferramentas, bem como outras, têm gerado alguma polémica dado o abandono parcial de criação do designer para com a ferramenta, levando a uma redefinição do papel do designer enquanto criador ou orientador. Existe um questionamento e adaptação do valor das capacidades humanas, devemos saber usar as ferramentas e preencher as falhas que as mesmas tem ao invés de tentar ultrapassá-las. A criação de resultados diferentes, sempre que é necessário ou feito esse pedido, origina um acréscimo no valor estético de cada resultado diferente.

Através de um conjunto de regras e de limites impostos, é possível a criação de trabalhos únicos, outrora repetidos, elevando o Design Generativo como uma das escolhas principais aquando da procura de resultados finais únicos.

Plotter - Uma plotter é um tipo de impressora que recebe instruções de um computador, por meio de números precisos ou aleatórios, para criar desenhos. Estes desenhos são mais geométricos constituindo padrões organizados ou irregulares, tendencialmente complicados de reproduzir manualmente. Estes padrões introduziram uma nova forma experimental de expressão criativa, alcançada apenas por aqueles que possuísem formação mais técnica, dado que as linguagens de programação não eram ensinadas aos artistas.

ChatGPT - O ChatGPT é uma tecnologia de inteligência artificial (AI) criada pela OpenAI, que permite a geração automática de texto. É uma tecnologia baseada numa rede neural chamada Transformer, que é treinada para aprender a escrever de forma semelhante a um ser humano.

O Design Generativo é o ponto de encontro entre a arte e a programação, cujo conteúdo é desenvolvido segundo métodos lógicos e rigorosos, resultando assim em criações imprevisíveis.

Desta forma, o design generativo preocupa-se com a metodologia da sua produção e não com o estilo propriamente dito do resultado final. (Pearson M., 2011)

O design generativo, incorpora uma teoria da complexidade não vinculada a nenhuma tecnologia específica. As obras de design generativo podem, portanto, ser não digitais, orgânicas e assim por diante. Esta inclusão de obras não digitais aconteceu devido a uma retrospectiva, depois da generatividade se manifestar como uma tendência dentro da arte digital. Por uma questão de sequência histórica, é certamente verdade que começamos a pensar sobre o design generativo no sentido mais amplo apenas depois de primeiro encontrarmos métodos generativos específicos. Mas mesmo no caso de Philip Galanter, pioneiro no Design Generativo, o seu trabalho parte de uma exploração de processos e sistemas analógicos e só posteriormente inicia a exploração de projetos generativos integrando ferramentas digitais. Philip Galanter (1953) é um dos pioneiros na definição e explicação do design generativo, área que atualmente ensina na Texas A&M University. Os seus projetos/obras de arte derivam entre sistemas de hardware generativos, exposições, instalações, som, luz, etc. Foi dos primeiros a executar uma tentativa de definição da própria área, bem como posteriormente deu inúmeras conferências e entrevistas relativas ao tema.

Sistemas analógicos, como sintetizadores de música e feedback de vídeo, vieram primeiro, admite o mesmo.

Não há dúvida de que a tecnologia digital oferece aos artistas generativos uma alavanca sem precedentes na criação e exploração de sistemas. A desvantagem é que acabamos por criar esses sistemas no mundo virtual/digital do computador e não no atraente mundo físico visceral onde vivemos e onde vivem as nossas questões emocionais e espirituais. Em última análise, porém, o que diferencia o design generativo de outros tipos de arte é o uso de sistemas. E é aqui que residem as novas noções da ciência da complexidade com uma visão crítica. Alguns sistemas são altamente ordenados, outros sistemas são altamente desordenados e aleatórios. Mas ambos são, à sua maneira, bastante simples. Para os artistas generativos de hoje, o desafio é o complexo domínio intermediário da ordem parcial e da desordem parcial.

É aí que, segundo Philip Galanter, existe a própria vida. O *Condensation Cube* de Haacke sugere isso, assim como o interesse de alguns artistas em *A-Life*. Mas há muito mais trabalho a ser feito. (Galanter P., 2003)

O Design Generativo permite evitar a intencionalidade, pois mesmo com uma limitação no algoritmo, por haver uma cedência da nossa liberdade para a “máquina” (seja qual for a forma que esta assume), a mesma age de forma autónoma e por isso imprevisível para nós.

### (2.2.1) Primeiros Relatos Generativos

Como já referido anteriormente, está implícito no Design Generativo que o autor deve abdicar parcialmente do controlo para que o sistema possua algum grau de autonomia, gerando uma grande quantidade de possibilidades combinatórias a grande velocidade estando assente também no conceito de complexidade. Philip Galanter diz-nos que devido a esta definição temos várias razões para interpretar “formas de arte antigas” como generativas. Galanter afirma que o Design Generativo surge antes do avanço tecnológico e da era dos computadores, proferindo mesmo que esta forma de arte é tão antiga como a própria “Arte”. Assim podemos olhar para esta arte como algo que nos tem acompanhado há muito tempo. Geralmente associamos o aparecimento do Design Generativo com o surgimento dos próprios computadores, contudo, isso não é verdadeiramente correto.

Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.

(Galanter P., 2003)

O termo Design Generativo começou a ser utilizado em 1965 por Margaret Boden (1936), professora e investigadora no Departamento de Informática da Universidade de Sussex, Inglaterra, cujo trabalho gira em torno da inteligência artificial, psicologia, filosofia e programação procurando definir os limites da arte generativa, constituindo uma definição etimológica, e Ernest Edmonds (1942), artista britânico cujo trabalho se insere no campo da arte computacional, arte generativa, arte algorítmica, no contexto da representação de formas gráficas através da automatização, começando pelos trabalhos exibidos por Georg Nees (1926-2016) e Frieder Nake (1938), em Fevereiro de 1965, na exposição *Generative Computergraphik* em Stuttgart, Alemanha. Neste contexto a palavra generativo foi usada para identificar a arte produzida a partir de um programa de computador e, portanto, foi, pelo menos em parte, produzido automaticamente. Defendem que nem todo o design generativo envolve o uso de computadores e que este conceito surge bem antes dos mesmos. Alguns exemplos de design generativo antes da era dos computadores incluem casos, como o mencionado por Phillip Galanter na entrevista *Generative art is old as art*.

O mesmo refere que a experiência de arte generativa mais importante com que se deparou nos últimos anos é uma obra da qual nunca saberemos quem é o autor. Com aproximadamente 70.000 anos, descoberta em 1999, é considerada a obra de arte mais antiga conhecida, a qual consiste numa inscrição em ocre vermelho com a forma de mosaicos triangulares, sendo uma exploração estética e de padrões. É considerado generativo, na medida em que um sistema autónomo e abstrato criou a forma em vez de a mesma ser criada no momento pela intuição do artista. Segundo Galanter os sistemas de ladrilhos são considerados algoritmos que existem há muito tempo antes dos computadores. (Galanter, P. (2003) Se procuramos reconstruir uma genealogia mais alargada das origens do design generativo, a obra *Musical Dice Game* (1757), de Johann Philipp Kirnberger's (1721- 1783), em alemão, *Musikalisches Würfelspiel*, pode ser



Fig 01 - Sistemas de grades/ladrilhos triangulares, com cerca de 70.000 anos considerada uma das primeiras obras de Design Generativo segundo Phillip Galanter



Fig 02 - Condensation Cube de Hans Haacke

considerado um projeto pioneiro de exploração de sistemas generativos baseados em aleatoriedade.

Este projeto consistia num sistema de lançamento de dados que gerava músicas aleatórias, dentro de parâmetros já pré-estabelecidos, sendo um jogo bastante popular na Europa no século XVIII. O *Condensation Cube* (1963-1965), de Hans Haacke, é um exemplo de arte generativa não computacional bastante conhecido, e um dos projetos pioneiros neste tipo de arte. Esta obra consiste num cubo de perspex (material semelhante ao plástico) selado com uma pequena quantidade de água no seu interior, que explora interações entre sistemas físicos e biológicos e os seus processos naturais. O cubo sofre interações não derivadas do público, mas sim devido ao movimento interno do mesmo. Este movimento é

influenciado por factores como a luz e temperatura que provocam a evaporação e consequente condensação da água que escorre pelas paredes do cubo, mantendo-o em constante mudança de uma forma imprevisível devido ao grau de autonomia criado pelo ambiente onde a obra está colocada. Deste modo nunca é possível prever o aspeto da obra, devido a ser um “sistema” que atua sobre si mesmo sem necessitar de intervenção, tanto por parte do artista, como do público. A sua autonomia e constante mudança são características que lhe conferem o seu carácter generativo.

Sem pretendermos ser exaustivos na análise de projetos que marcam a sua evolução histórica, é razoável concluir que a Arte Generativa nos tem vindo a acompanhar discretamente ao longo dos séculos e que a mesma não tem de ser realizada por via de um computador. Projetos mais ou menos complexos, com o uso ou não de um computador, de cariz físico, psicológico, sociocultural, biológico ou abstrato, que utilizam sistemas de química, biologia, mecânica e robótica, randomização, matemática, mapeamento de dados, simetria, são agora, após análise, considerados Arte Generativa. Já no século XX, e após algumas experiências com base na arte generativa analógica, a arte digital surgiu como consequência da evolução tecnológica dos computadores na década de 1960. Com estas ferramentas já era possível desenvolver obras artísticas generativas totalmente digitais.

Pela sua influência e importância, um conjunto de autores merecem uma referência especial, pois contribuíram para a consolidação do conceito de design generativo: George Nees, A. Michael Noll e Frieder Nake foram dos primeiros nomes associados a esta representação visual digital generativa, bem como John Whitney, Herbert Franke, Ben Laposky e Vladimir Bonacic. Na seguinte secção enumeram-se, à semelhança dum arquivo, estes autores, bem como as suas concretizações relativas ao Design Generativo:

### Ben Laposky (1914 - 2000)

Benjamin Laposky foi pioneiro na representação do som em 1953 através da computação gráfica. Foi um matemático, artista e designer em Cherokee, Iowa. Foi autor da primeira computação gráfica, utilizando um osciloscópio como meio de criação de arte abstrata. Todo o processo era feito de uma forma analógica, uma vez que nesta data os avanços



Fig 03 - Oscillon por Ben Laposky

tecnológicos ainda eram poucos. O som era capturado por um osciloscópio e representado pelo mesmo consoante as ondas sonoras transmitidas. **John Whitney (1917 - 1995)**

John Whitney, fundou em 1960 a sua empresa chamada Motion Graphics Incorporated. Foi considerado um pioneiro nas animações digitais, realizando-as em computadores analógicos. *Vertigo* foi a primeira animação feita por um computador e foi realizada por Whitney.

Juntamente com o engenheiro Jerry Reed, ainda em 1980, desenvolveu também um programa de computador que permitia trabalhar composições visuais e música simultaneamente, em tempo real. *Whitney-Reed RDTD (Radius-Differential Theta Differential)* foi o nome atribuído a este sistema áudio-visual.

### Georg Nees (1926-2016)

Georg Nees foi, em 1965, o primeiro a expor publicamente a sua arte computacional, hoje intitulada de arte digital, na sala da Studiegalerie of Technische Hochschule Stuttgart (atualmente Universidade de Stuttgart) com desenhos gerados por um algoritmo num programa de computador.

Os desenhos apareciam de forma codificada numa fita de papel perfurada antes de serem gerados fisicamente por uma máquina de desenho, Zuse Graphomat Z64, intitulada na altura nos EUA como uma plotter.

### Herbert Franke (1927 - 2022)

Nascido em 1927, o cientista, escritor e artista trabalhou desde 1973 até 1997 na Universidade de Munique como professor de arte computacional.

O trabalho artístico de Herbert Franke, é conseguido através de abstrações feitas com um osciloscópio, fazendo o cruzamento entre a vertente artística e científica, através do uso de algoritmos, produzindo imagens. Em 1969, Franke e Peter Henne, cientista de computação, juntaram-se para criar um projeto chamado *KAES (Kurven AEsthetische)*, que consiste numa série de desenhos digitais.

Peter Henne estava encarregado de criar o programa onde é possível manipular curvas algébricas. Após a programação estar concluída, fizeram a ligação a uma plotter de desenho que originou os grafismos.

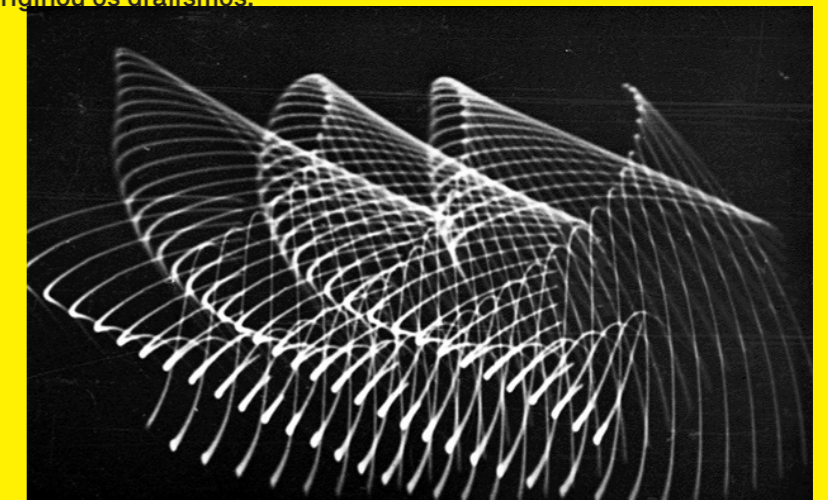


Fig 06 - Imagens computacionais com o osciloscópio, por Herbert Franke

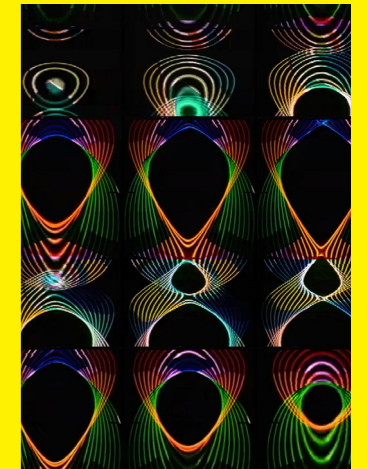


Fig 04 - Keyframes da série Montage, 1961 por John Whitney



Fig 05 - Desenho computacional de Georg Nees

## Vladimir Bonacic (1938 - 1999)

Vladimir Bonacic, foi um cientista, artista e cibernético, que começou por estudar eletrônica nuclear na Faculdade de Engenharia Elétrica e de Computação em 1964. Quatro anos depois, em 1968, começou a desenvolver as suas primeiras obras de arte cibernéticas utilizando sistemas de computação. O seu interesse como cientista foi perceber a comunicação entre seres humanos, estudar a fundo a Teoria da Percepção, a cibernética, a eletrônica e os computadores. Juntando todas estas suas áreas de interesse, através da arte digital abstrata, criou *Dynamic Objects*, que consiste na composição de uma série de luzes sobre um painel de alumínio, que são controladas através de um computador. Cada lâmpada está associada a um algoritmo no computador que controla a sua atividade. É gerada uma função no computador que determina quais as lâmpadas que são ativadas e a duração de transmissão de luz. Com isto, Vladimir Bonacic consegue criar composições generativas que trabalham com cor, composição e velocidade.



Fig 07- Exposição One-Man no Museu de Israel, Jerusalem, 1975 por Vladimir Bonačić

## Frieder Nake (1938)

Frieder Nake pertence aos fundadores da arte algorítmica, que ainda era chamada de arte computacional, quando surgiu na primeira metade da década de 1960. Ele produziu os seus primeiros ensaios visuais em 1963. Expôs na Galerie Wendelin Niedlich em Stuttgart de 5 a 26 de Novembro de 1965. Declarou em 1971 que iria deixar de produzir arte computacional, pois não conseguia contribuir ativamente na mesma e ser um ativista político contra o capitalismo. Apenas retomou em 1999 com o projeto *compArt: a space for computer art*, onde retomou às suas origens como teórico, escritor, professor, artista e criador de arte digital.

## A. Michael Noll (1939)

Juntamente com Georg Nees, foi um dos pioneiros do uso do computador e de meios digitais para obtenção de resultados visuais. Expôs a sua arte computacional em 1965, na Howard Wise Gallery, em Nova York, de 6 a 24 de abril, juntamente com Bela Julesz. Esta foi das primeiras exposições desse género nos Estados Unidos.

*Computer-Generated Ballet* foi o primeiro projeto com o uso de um computador digital para criar uma animação de personagens num palco virtual.

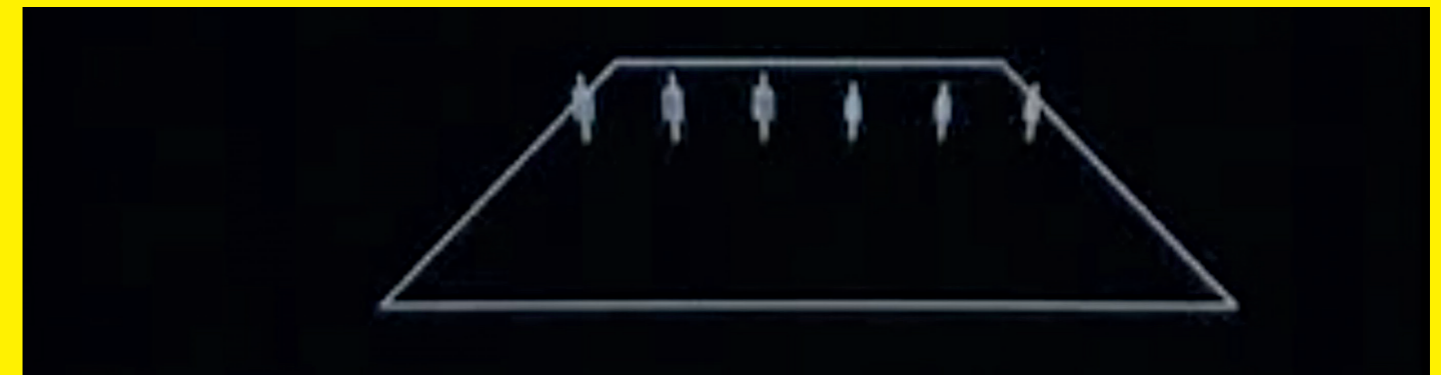


Fig 09 - Computer-Generated Ballet por A.Michael Noll

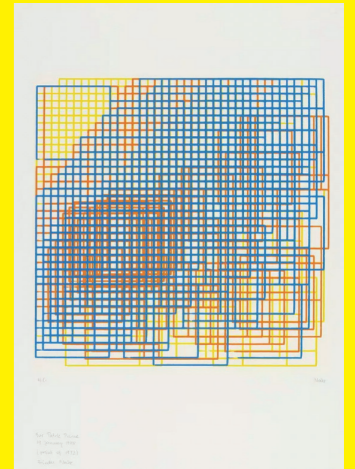


Fig 08 - Walk Through Raster  
Vancouver Version, 1972,  
Frieder Nake

### (2.3) Creative Coding

Creative Coding, as “adiscovery-based process consisting of exploration, iteration, and reflection, using code as a primary medium, towards a media artefact designed for an artistic context.”

(Mitchell M., Bown O. 2013)

Tal como outras ferramentas de design gráfico que nos são próximas, como Adobe Indesign, Adobe Photoshop, Adobe Illustrator, AfterEffects, Figma, Adobe Xd, entre outras, o creative coding, independentemente da linguagem utilizada, constitui um sistema de grelhas e coordenadas, para a colocação de formas 2D, 3D e texto. Dependendo da linguagem de programação utilizada, a sintaxe destes elementos pode diferir, mas a funcionalidade é semelhante.

Esta linguagem de programação adotada é o que permite o diálogo entre o programador/criativo e o computador. Ela permite estabelecer um conjunto de regras para a obtenção do objeto visual pretendido. É fulcral a compreensão dos conceitos básicos, da linguagem e do ambiente de programação escolhido, para conseguir identificar relações causa-efeito entre a escrita de um certo algoritmo e os elementos visuais que se geram a partir dele. Como já vimos, com o surgimento do computador, o acesso ao mesmo era limitado e o seu uso era restrito ou auxiliado por engenheiros e/ou cientistas que compreendiam as linguagens de programação para construir as regras para o computador executar determinada tarefa. Isto fez com que a programação por norma fosse destinada a engenheiros pelo seu carácter prático e funcional. Mas o creative coding, por ter este carácter da experimentação, motiva a sua utilização por pessoas de vertentes fora da engenharia, como é o caso das artes e do design, abrindo um mundo de possibilidades e resultados visuais para o design.

O Creative Coding é um método que não está vinculado a um único meio, pois através de uma única ferramenta (a linguagem de programação/código) conseguimos produzir uma variedade de resultados que pode ser usada para uma multiplicidade de propósitos.

Os resultados (visuais) do creative coding não são diretamente previsíveis com antecedência e muitas vezes são apenas vagamente predefinidos, pois é algo que apesar de vir da programação, contrariamente à programação tradicional, é algo que resulta da experimentação e da expressão. Por isso, a funcionalidade fica muitas vezes colocada em segundo plano, dado que o maior objetivo é gerar resultados criativos/visuais, daí a aleatoriedade ser um dos conceitos que se insere dentro do espectro do creative coding, bem como o motiva e leva à sua procura. O avanço das tecnologias, dos computadores e da web trouxe consigo também a rápida expansão da informação. Devido à saturação da mesma torna-se hoje cada vez mais difícil de ser “original”. De forma a “fugir” da previsibilidade, o creative coding torna-se cativante por possuir este carácter aleatório. O seu carácter multidisciplinar com a enorme quantidade de saídas é também o que torna este método tão apelativo. Algumas das ferramentas e das linguagens são mais apropriadas para criar aplicações interativas e conteúdo para a web como é o caso do p5.js, outras são mais apropriadas para a produção de animações 3D, como é o caso do Processing.

A interatividade, que está ligado aos novos meios de comunicação, permite que enquanto utilizadores tenhamos influência sobre o conteúdo ou a forma, conseguindo dar instruções ao sistema/ferramenta, originando diferentes respostas por parte da “máquina”. Ora, o creative coding carrega também consigo esta capacidade. Comparativamente com outros programas, permite que, enquanto utilizadores as nossas ações alterem o resultado visual final, mas não só: com a ligação ao som do exterior ou um input de música podemos produzir alterações e reações no nosso resultado final, tornando soluções estáticas em reativas, característica que não encontramos noutras ferramentas de design gráfico ou animação.

São estes fatores de multidisciplinaridade, de interatividade, de imprevisibilidade e de aleatoriedade que cativam as pessoas a utilizarem e a aprenderem mais sobre creative coding, e foram sem dúvida também estes os conceitos, objetivos e motivações para este projeto, bem como a infinidade de resultados possíveis de obter com esta ferramenta.

### (2.4) Creative Coding, Design Generativo e Design Paramétrico

Um conceito que muitas vezes pode ser utilizado para se referir ao Creative Coding ou mesmo ao Design Generativo é o Design Paramétrico. Apesar do creative coding ter uma base de parâmetros, e/ou de algoritmos, isso não significa que o mesmo possa ser considerado design paramétrico.

A base do Design Generativo é a criação dum algoritmo/sistema/regras que projeta algo de forma autónoma. Este algoritmo é, no fundo, um conjunto de passos precisamente definidos e ordenados de forma a serem executados pela máquina. Quando este mesmo algoritmo gera mais do que um resultado, chamamos de Design Generativo. Ou seja, no fundo entende-se por Design Generativo todo o Design Paramétrico que através dos seus parâmetros produz mais do que um resultado. O Design Paramétrico isoladamente confere uma característica distinta do Design Generativo, pois enquanto o Design Generativo nos permite criar diversas soluções e soluções dinâmicas, o Design Paramétrico cria um algoritmo que gera uma única solução estática, geralmente geometricamente complexa, daí a precisão na colocação de parâmetros específicos definidos pelo utilizador como largura, altura e curvaturas. Para conseguir criar formas mais complexas, sobretudo curvilíneas, irregulares e abstratas, o design paramétrico apoia-se em programas 3D.

Por outro lado, o design generativo é também uma técnica que se concentra em encontrar uma solução ideal ou diversas variações da mesma, ao invés do design paramétrico que procura encontrar uma solução personalizada mediante os parâmetros colocados. Pela flexibilidade e adaptabilidade de alteração dos parâmetros a meio da criação, bem como da urgência de prazos e a velocidade de concepção exigido nos dias de hoje, a parametrização permite-nos obter resultados precisos de forma rápida, com a inserção de alguns parâmetros. Esta metodologia tem sido aplicada principalmente na arquitetura. Através desta metodologia e ferramenta, os arquitetos e designers libertam-se da repetição de tarefas para explorar e testar diferentes formas no design.

Quando se desenvolve um projeto, um dos primeiros passos a estimar é o orçamento, algo por vezes difícil, pois o design pode sofrer alterações a meio da sua criação. Com o design paramétrico, através do algoritmo, o mesmo consegue prever as variáveis que um projeto pode ter e por isso o custo do projeto pode ser bem definido desde o início, o que facilita a realização do orçamento e a estimativa e discrepância de custos.

For example, a tree is parametric: it has a trunk, branches, leaves, and roots, all of which are generated by a set of parametric equations. These equations take into account the tree's environment, such as the amount of sun and water available, to generate a unique solution that is optimized for that particular tree.

In the same way, parametric design takes into account all of the variables in a design problem to generate a unique solution. Technology allows engineers to analyze and mimic nature's complexity in the digital world, and the potential applications are endless. With this data, it can be used in applications from architectural designs to aerospace engineering

(Pickard N., 2022)

Apesar do termo ter começado a ser empregue na década de 1970, aquando do surgimento do computador e dos termos algoritmos, algorists e arte computacional como vimos anteriormente, acredita-se que o pioneiro do design paramétrico tenha sido o arquiteto Antoni Gaudí com o seu modelo invertido na Sagrada Família, em Barcelona.

No século XIX, ainda que não se dispusesse dos meios digitais para a criação do design paramétrico com o auxílio de um algoritmo, a sua obra é considerada design paramétrico, dada a sua abordagem e os modelos físicos criados. Para a criação da Sagrada Família como a conhecemos hoje, o mesmo criou um modelo feito de cordas pesadas como uma versão invertida do edifício. Através deste modelo invertido, António Gaudí conseguia manipular cada ponto da sua obra e fazer com que a sua obra, como um todo, se adaptasse mediante as suas alterações, derivado também da força gravitacional. Atualmente, com o avanço tecnológico, nomeadamente dos computadores, e com o surgimento de softwares como o Grasshopper3D, Autodesk 3DS Max, Autodesk Maya, Autodesk Dynamo, os projetos paramétricos de arquitetura permitem-nos criar uma versão atualizada do mesmo de forma automática, em vez de realizarmos esse trabalho de forma manual.

Paralelamente o Creative Coding é muitas vezes comparado e confundido com Design Generativo. As duas áreas têm semelhanças, mas algumas distinções. A base do Design Generativo é a criação dum algoritmo/sistema/regras que projeta algo de forma autónoma. No entanto, contrariamente ao Creative Coding, este não se encontra vinculado a um computador. Como vimos nos exemplos anteriores, muitos dos primeiros designs generativos não foram criados com um computador, alguns deles foram feitos manualmente ou com o uso de outros objetos/tecnologias. No Design Generativo a base de criação é um sistema, regras que se repetem de forma a produzir algo.

No Creative Coding é imperativo a utilização do computador. Aqui o computador desempenha um papel principal, em igualdade com as instruções dadas pelo designer, sobre a forma de código.

Enquanto que o Design Generativo se foca na utilização de um algoritmo, seja ele na sua forma literal ou utilizando ferramentas não digitais, o Creative Coding, apesar de utilizar algoritmos, tem o seu foco no processo criativo. Ou seja, no sistema que produz o(s) resultado(s) visual(is) final(is), e não necessariamente neste mesmo resultado, sendo, no fundo, um diálogo de forma criativa com o computador. Rodenbröcker, T. (2022).

O foco do creative coding é sempre o de resultados inesperados e criativos, onde é possível antever mas não ver com precisão clara daí estar assente no conceito de emergência onde a combinação de fatores origina algo novo com um pensamento intencional mas não controlado. No design generativo, que pode ser meramente pragmático e técnico, existe um propósito específico, um produto final que se procura, logo todo o algoritmo é gerado de forma a obter isso. No creative coding o design não é intencional, com a sua base na experimentação e na imprevisibilidade, é um processo interativo em todos os seus estágios. O fator não intencional do creative coding dá-lhe abertura para alguma aleatoriedade e imprevisibilidade dos resultados, em oposição ao Design Generativo cujos resultados podem ser os esperados, quando formalizado o sistema/conjunto de regras, no início da sua produção. Baseado na experimentação e na reflexão de cada estágio do seu processo, o Creative Coding é visto como algo mais estético, ao invés de funcional. No entanto, dada a sua expansão tem vindo a ser utilizado na criação de publicidades.

Podemos concluir que o Creative Coding, o Design Paramétrico e o Design Generativo partilham semelhanças. No entanto como já referido anteriormente apesar de partirem de um conjunto de parâmetros/regras, o Design Paramétrico produz apenas um resultado premeditadamente pensado. Por sua vez, o Design Generativo e o Creative Coding produzem mais do que um resultado. No entanto o Design Generativo contém um certo grau de previsibilidade comparativamente ao Creative Coding que tem como base a aleatoriedade. Neste último é também imperativo o uso de um computador enquanto que no Design Generativo as ferramentas podem ser diversas, desde digitais a analógicas.

# (3) ESTADO DA ARTE

### (3.1) Design Generativo e Creative Coding hoje

O design generativo é um processo de criação de um algoritmo, com regras, parâmetros e alguns limites relativos ao processo de criação e geração dos resultados da própria ferramenta que é criada, que acaba por ser isoladamente também uma produção de arte em si. Em contraste com os artistas tradicionais ( não utilizam ferramentas generativas e cujo processo de trabalho gira em torno de um único resultado), os artistas generativos tiram proveito do poder do código para gerar milhares de resultados e ideias, numa fase inicial de exploração.

Com o avanço tecnológico, o design generativo que anteriormente era feito com processos manuais/analógicos, hoje gira maioritariamente em torno de processos digitais. Segundo Michael Hansmeyer, arquiteto e artista, o computador adquire o poder de nos surpreender. O processo de design encontra um equilíbrio entre o esperado e o inesperado, entre o controlo e a renúncia, mesmo que os processos sejam determinados, os resultados não são previsíveis. Hansmeyer, M. (n.d.).

O design generativo é cada vez mais aplicado a vários setores, à medida que a tecnologia é cada vez mais integrada no fluxo de trabalho. E, mesmo sendo algo que está em constante alteração, devido aos progressos tecnológicos diários, já oferece aos artistas, engenheiros e designers uma forma de ver e pensar o produto/resultado final de forma distinta do habitual. São diversos os projetos que utilizam design generativo, mas não é apenas esta a tipologia de aplicações deste método que podemos ver hoje. Desde publicações, exposições e plataformas de ensino, irão deixar-se aqui alguns como referência:

Jared Schiffman foi aluno de John Maeda, é um “creative technologist”. Atualmente é co-fundador e CEO da Phenomena, empresa de tecnologia educacional. É um artista que acredita que a criatividade é tão essencial para a engenharia como para as artes.

Trabalhou e trabalha atualmente com equipas de designer e programadores no desenvolvimento de diversos projetos, nos quais se destaca um dos mais recentes: *Sunglass Hut Interactive Product Shelves*. Este projeto baseia-se na colocação de inteligência artificial nas lojas para dinamizar as compras a personalizar a experiência do utilizador de forma a torná-la mais individual e interativa.

No entanto, como já referido anteriormente, tanto o conceito de design generativo, como o de creative coding, não são recentes. Se retrocedermos às décadas de 1950 e 1960, Herbert Franke, artista norte-americano desenvolveu vários projetos que eram conseguidos através de abstrações feitas com um osciloscópio, cruzando a sua vertente artística com a ciência utilizando algoritmos para obter as imagens. Para além da junção improvável de diferentes áreas temos também a junção de diferentes ferramentas para a obtenção de resultados imprevisíveis, como era o caso do osciloscópio nesta década.

No âmbito de projetos individuais temos *Electric Sheep*, feito por Scott Draves, que é um engenheiro de software conhecido por criar *Electric Sheep*, uma animação abstrata infinita realizada através de um algoritmo. Este é um projeto voluntário de computação que visa ser distribuído aos computadores, de forma a ser exibido como protetor de ecrã. Podemos referir também *Dreamcatcher* da AutoDesk, um sistema de design generativo, utilizado pelos designers para a definição dos seus problemas de design, através de metas e restrições.

**Redes Neurais** - As redes neurais, também conhecidas como redes neurais artificiais ou redes neurais simuladas, são inspirados no cérebro humano, imitando a maneira como os neurônios biológicos enviam sinais uns para os outros. As redes neurais artificiais são compostas por camadas de nós, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada nó, ou neurônio artificial, conecta-se a outro e tem um peso e limite associados. Se a saída de qualquer nó individual estiver acima do valor limite especificado, esse nó é ativado, enviando dados para a próxima camada da rede. Caso contrário, nenhum dado é transmitido para a próxima camada da rede. As redes neurais dependem de dados que são inseridos e “aprendidos”, estes vão progredindo ao longo do tempo. Tarefas de reconhecimento de fala ou reconhecimento de imagem podem levar minutos versus horas quando comparadas à identificação manual por especialistas humanos. Uma das redes neurais mais conhecidas é o algoritmo de busca do Google.

Neste são apresentados ao designer diversas soluções para a resolução dos seus problemas, sintetizando as soluções que já atendem os seus objetivos. O designer pode colocar objetivos como requisitos funcionais, materiais, método de fabricação, restrição de custos, etc. Este sistema é mais utilizado no desenvolvimento de produtos.

No âmbito interativo/reactivo, um dos pioneiros do design generativo e do movimento artístico de inteligência artificial, é hoje Mario Klingemann, com o seu projeto *Memories of Passersby* by Mario Klingemann. Os seus projetos resultam de programação, redes neurais, código e algoritmos, aliado ao design. Na sua obra *Memories of Passersby*, Klingemann gerou retratos em tempo real, através de **redes neurais**.

*Memories of Passersby I* é um trabalho pioneiro de inteligência artificial e do design generativo. Totalmente autónomo, usa um complexo sistema de redes neurais para gerar um fluxo interminável de retratos, visões inquietantes de rostos masculinos e femininos criados por uma máquina. A obra de arte é apresentada como uma peça de instalação: a máquina de inteligência artificial está colocada num armário de madeira de castanheiro feito por medida, a conectar dois ecrãs. Ao contrário das instalações de arte generativa anteriores, *Memories of Transersby I* não contém um banco de dados, sendo um cérebro de inteligência artificial, desenvolvido e treinado pelo mesmo artista, que cria novos retratos, pixel por pixel, em tempo real. As saídas exibidas no ecrã não são combinações aleatórias ou programadas de imagens existentes, mas sim, obras de arte únicas geradas por IA. O fluxo de imagens apresentado não segue uma coreografia pré-definida, mas é o resultado da interpretação da própria inteligência artificial; a natureza complexa desse ciclo de feedback significa que nenhuma imagem será repetida. *Memories of Passersby I* contém todos os algoritmos e GAN's (redes generativas contraditórias) necessários para produzir uma sucessão infinita de novas imagens enquanto estiver em execução. Nesse sentido, *Memories of Passersby I* marca um passo significativo no campo emergente da arte da inteligência artificial. Até agora, os colecionadores foram capazes de adquirir peças resultantes de redes neurais com curadoria humana; *Memories of Passersby I* é um agente criativo independente, sendo que cada edição irá gerar retratos infinitos à sua maneira e, como tal, podem ser considerados únicos. Para desenvolver *Memories of Transersby*, Klingemann treinou o seu modelo de inteligência artificial usando milhares de retratos dos séculos XVII a XIX. Ele criou uma aplicação semelhante ao Tinder para acelerar o processo de aprendizagem e ensinar à máquina as suas próprias preferências estéticas, influenciadas por figuras surrealistas como Max Ernst. Como resultado, *Memories of Transersby I* apresenta interpretações misteriosas do rosto humano, exemplos gerados por inteligência artificial ao que André Breton chamou de “beleza convulsiva”. Às vezes, as imagens fundem-se em arranjos abstratos de pixels enquanto a máquina se esforça para criar um novo retrato. Para o espectador, *Memories of Transersby I* é uma experiência hipnótica, a oportunidade de ver um cérebro de IA “pensar” em tempo real e ver retratos verdadeiramente únicos que não são gravados nem repetidos. (Kistler, P., & Benney, M. (n.d.))

A convite da L'Avant Galerie Vossen, em Paris, Ribbie Barrat desenvolveu a exposição *Infinite Skulls* onde trabalhou com o pintor Ronan Barrot, desenvolvendo a exposição *Barrat/Barrot: Infinite Skulls*. Esta exposição apresentou um número infinito de caveiras. A partir das 450 caveiras já desenhadas pelo pintor Ronan Barrot, Robbie digitalizou-as e desenvolveu uma rede neural para gerar novas imagens a partir dessas obras. É também relevante destacar as plataformas de exposição de projetos de Design generativo, como é o caso da *Generative Gallery*. Esta é uma galeria online que existe desde 2008, que alberga tanto projetos de arte digital locais (Moscou), como internacionais, com exposições imersivas e publicações, permitindo para além da sua visualização online também a experienciar fisicamente. No âmbito das publicações, temos várias publicações que muitas vezes transbordam o tema do Design Generativo para o Creative Coding, pela linha tênue que os distingue. No entanto, podemos destacar dois livros que abordam ambos os temas, como o *Generative Design: Visualize, Program, and Create with JavaScript in p5.js* e o *Graphic Design in the Post-Digital Age: A Survey of Practices Fueled by Creative Coding*, que reúnem entrevistas, bem como projetos das duas vertentes.



Fig 10 - Memories of Passersby I de Mario Klingemann

Ao analisarmos o Estado da Arte sobre design generativo e creative coding, identificámos estes e outros autores que assumem uma grande influência e relevância.

Os artistas, geralmente carregam um estigma em relação à programação tal como os engenheiros também carregam um estigma em torno das dificuldades de expressão criativa. Nos dias de hoje, a capacidade, versatilidade e flexibilidade que a programação tem, permite realizar projetos expressivos. Ao contrário do foco funcional da maioria do uso de código, o creative coding usa linguagens de programação para usos maioritariamente artísticos e não só.

O campo do design e da programação deixaram de ser entidades separadas, e estão agora intimamente ligados. Os projetos de creative coding podem passar por projetar vídeo em superfícies de formato irregular, como esculturas ou edifícios, por projetos interativos no qual um programa, geralmente usando algoritmos, gera um certo número de saídas com base num conjunto de restrições, ou até mesmo uma espécie de arte performativa na qual um código é criado em tempo real, geralmente envolvendo som, imagem e luz.

## John Maeda (1966)

John Maeda é um designer gráfico americano de origem japonesa. Foi criado em Seattle, Washington, onde ajudava o pai, que trabalhava numa fábrica de tofu, a documentar todos os papéis da mesma, no computador. Esta foi a primeira interação do mesmo com a computação, levando-o a matricular-se no Instituto de Tecnologia de Massachusetts (MIT), onde estudou engenharia elétrica e ciência da computação. Depois disso mudou-se para o Japão onde estudou arte e design.

Formou um estúdio de pesquisa denominado de ACG (Aesthetics Computation Group), em 1990, no MIT, onde explorou a interseção das áreas que estudou, como o design, a arte, a computação e a engenharia. Foi o primeiro grupo a defender o uso de computadores como ferramenta na criação de objectos de arte e design, algo que era bastante polémico na época, pois acreditava-se que o computador distorcia o processo, e mesmo que facilitando a cópia de elementos, o seu processo poderia gerar falhas comprometendo o rigor.

Ficou conhecido pelo lançamento das *Ten Laws of Simplicity*, onde permite repensar a prática do design das tecnologias e dos negócios, ensinando a precisar de menos para obter mais elementos, e no seu livro *How to Speak Machine* explora o pensamento do computador e como qualquer indivíduo, mesmo que não seja programador, deve entender como este raciocínio funciona.

Hoje é vice-presidente de design e inteligência artificial da Microsoft. É importante referir que foi professor no MIT, onde ajudou designers a programar e engenheiros a projetar. Teve como alunos no Creative Coding e no Design Generativo referências de hoje, como Golan Levin, Jared Schiffman, Casey Reas e Ben Fry.

## Casey Reas (1972)

Casey Reas é um artista americano (de Ohio) que também, por volta dos anos 2000, revolucionou o mundo da codificação criativa e do design generativo, não só através dos seus projetos, podendo destacar-se o projeto *The Path Software*, bem como na criação do Processing, linguagem de programação que desenvolveu com Ben Fry, designer especializado em visualização de dados. Junto com a própria linguagem de programação (Processing), foi também criada a Processing Foundation que visa promover a aprendizagem de software no âmbito das artes, e a aliança entre estas e outras áreas. O principal foco é sem dúvida ajudar pessoas, que queiram aprender linguagens de programação de forma a realizarem projetos criativos, especialmente aqueles que não tem acesso facilitado às ferramentas e recursos relativos ao mesmo, ou aqueles que não tiveram um acesso devido às suas etnias, género, classe social, sexualidade, idade, localização geográfica e/ou deficiências. No Processing Foundation, para além do Processing (Java), linguagem primitiva, existem outros softwares desenvolvidos posteriormente como p5.js (Javascript) e Processing Android. Segundo os mesmos, a sua filosofia e política é baseada no FLOSS (free open source software), onde vem o software como um meio para pensar e fazer algo, bem como todo o processo criativo e exploratório. (Processing Foundation, n.d.)

A Infinite Command Team deriva do projeto anterior Signal to Noise, onde o mesmo explora uma espécie de colagem, utilizando sinais de televisão como matéria-prima para transcrever estes inputs visuais. Para Infinite Command Teamn, Reas desenvolveu um novo mecanismo de colagem mais evoluído para criar imagens. Aparecem ocasionalmente imagens de televisão, e a velocidade com que a animação e as sequências de cores mudam lembra uma rápida navegação de canal. Usando pixelização de diferentes pesos e tamanhos, a peça cria um mosaico digital de sinais de televisão que se tornam abstratos e semelhantes a colagens, reminiscentes da navegação nos canais de TV. A peça é uma celebração da arte e da tecnologia que mostra o potencial de combinar fragmentos digitais num trabalho holístico, combinando uma estética familiar da era moderna.

## Daniel Schiffman (1973)

Daniel Schiffman é um dos membros mais antigos da Processing Foundation e atualmente o diretor da mesma, que deu uma das maiores contribuições a nível de documentação e ensino da linguagem/ferramenta Processing é Daniel Schiffman. Para além da extensa documentação na própria plataforma da fundação, é autor de mais do que um livro sobre o creative coding, *The Nature of Code* (2012) e *Learning Processing* (2008).

## Karsten Schmidt (1975)

Karsten Schmidt, é um designer computacionais que combina código, arte e interatividade, constituindo-se, deste modo, como um designer transdisciplinar.

If we don't take responsibility as makers we sacrifice everything sooner or later. We have the power! The people who create things, who make things work, we have the power. No politician has that.

I'm not a very linear person I'm afraid,' says Karsten Schmidt halfway into a wildly discursive interview that takes in everything from literature and architecture to education, designers' social responsibility and the arrogance of creative directors who talk dismissively about 'techies'. Hailed as a virtuoso among new-media designers, Schmidt has collaborated on some of the most striking projects of the past few years, including the installation Forever at London's Victoria & Albert Museum, and Advanced Beauty: Enerugii (both in collaboration with Universal Everything); the identity for Onedotzero 2009 in London (with Wieden + Kennedy); EDA award-winning projects for the London College of Fashion and KEF's Muon speakers (while at Moving Brands).

Sometimes hailed as a 'Processing guru', Schmidt is at pains to underline his desire not to be defined by any one program or language; 'code is far more flexible than any tool'. He is currently building a collection of Toxiclibs – 'building blocks' for Java and Processing development.

(Eye Magazine, 2009)



Fig 11 - Amsterdam Sinfonietta Identity, Studio Dumbar, 2019

## Studio Dumbar (1977)

Numa vertente entre o design gráfico ligado com o creative coding temos o Studio Dumbar. Fundado por Gert Dumbar em 1977, é uma um estúdio com sede em Roterdão que nos apresenta uma quantidade de projectos com uma ligação direta entre estes dois conceitos enaltecendo a aplicação e as diversas possibilidades que o creative coding pode ter quando aplicado em projectos de design gráfico. Como exemplos temos a criação da identidade visual para a Amesterdam Sinfonietta, que é reativa á música e consequentemente com uma multiplicidade de resultados.

## Tim Rodenbröker (1985)

Tim Rodenbröker é um designer alemão que ensina e trabalha na prática do Creative Coding aplicado ao Design Gráfico. Dedicou-se a estudar o que é o Creative Coding, desenvolvendo diversas ferramentas e desafios em torno desta temática. Possui hoje uma plataforma de partilha sobre Creative Coding, desenvolveu o desafio do *Programming Poster*, bem como a ferramenta/plataforma *p5studio*. Focou-se inicialmente na ferramenta do Processing, e mais recentemente no p5.js, estabelecendo um comparativo entre as duas. *Programming Poster* é mais do que uma superfície com letras, cores e imagens: é um interface interativo, animado e orientado a dados. Envolve todos os sentidos e disciplinas e funde o design gráfico com as tecnologias. Desafiado pelos seus próprios alunos, que desacreditaram que era possível criar uma peça de design gráfico com uma linguagem de programação e sem conhecimentos prévios sobre a mesma, Tim Rodenbröker desafiou os alunos a criarem um poster programado em Processing, limitando-os ao uso da cor azul. Este desafio deu origem a inúmeros resultados em tudo distintos uns dos outros. Por sua vez, *p5studio*, é uma plataforma gerada em p5.js que representa um software de design com o objetivo de demonstrar as capacidades técnicas e criativas da linguagem de programação, bem como de um software dedicado a um sistema de design. O projeto consiste na criação de posters com alguma parametrização da parte do autor. Apesar de conter alguma limitação de escolhas tipográficas e cromáticas espelha a potencialidade desta linguagem de programação.

The capabilities of web-technologies changed drastically in the last 10 years. New open-source tools enable programmers to develop enterprise-level applications with just HTML, CSS and Javascript. More and more software-products popup on the market to compete and subverse the major players, who felt safe in their monopoly positions for decades.

(Rodenbröker T., 2019)

Hoje em dia, Tim Rodenbröker ajuda as pessoas a tornarem-se ao que intitulou de *Creative Technologists*, auxiliando-as a quebrar o medo que existe da programação, bem como este território digital ainda hoje desconhecido. Mais recentemente terminou a sua tese de Mestrado na Universidade de Bielefeld, na Alemanha, com o projeto Creative Coding na Schule des Denkens, onde o mesmo explorou o Creative Coding, na sua forma prática e filosófica, como ferramenta de design e método de ensino educacional, o que vai sem dúvida ao encontro do que é pretendido explorar com este projeto. Acredita que existe um espaço entre a criatividade e a tecnologia iniciando-se uma necessidade de criação de trabalho na ponte/interseção entre estas duas áreas, o design e a programação. Esta necessidade de ligação das áreas prende-se não só com questões económicas, mas também pelo próprio questionamento do digital e das artes e do papel que representam hoje.

**p5studio** 0.1.1

|          |      |
|----------|------|
| HEADLINE | HIDE |
| SUBLINE  | HIDE |
| IMAGE    | HIDE |
| IMAGE2   | HIDE |
| GRID     | HIDE |
| BASE     | HIDE |

TEXT

FONTSIZE

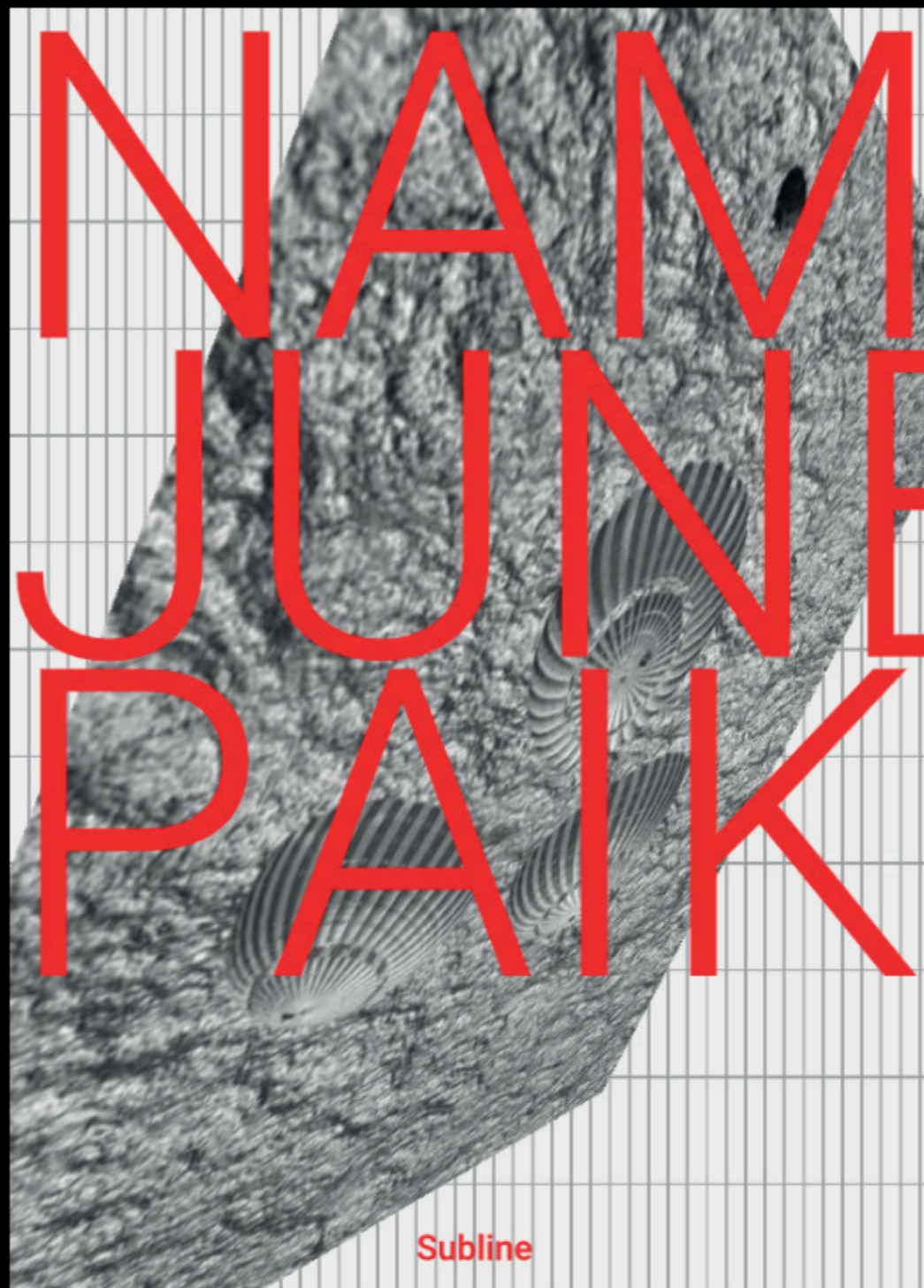
LINEHEIGHT

COLOR

FONT

SELECT FONT

ALIGN



**VIEW**

ZOOM

BRIGHTNESS

**OUTPUT**

STATE

PNG

JPG

VIDEO

Fig 12 - p5studio por Tim Rodenbröker

## André Burnier (1995)

André Burnier é um designer e creative coder brasileiro que tem focado a sua atividade na interseção entre o design gráfico e a programação, focando-se na programação como uma ferramenta para o design. Ao que o próprio intitula de experiências, e como a definição da palavra indica, um teste feito de modo experimental, uma tentativa, uma aprendizagem obtida e aprimorada com o passar do tempo, Burnier tem criado uma série de animações usando o Processing e o P5 e trabalha essencialmente com tipografia, dado que este é um elemento fulcral no design gráfico. Estas experiências não têm qualquer intuito ou aplicação final, senão a mera experimentação em si e a percepção da potencialidade destas ferramentas.

## Random International (2002)

A Random International é um estúdio de arte experimental, com sede em Londres, e foi criado em 2002 por Stuart Wood, Flo Ortkrass e Hannes Koch. O foco deste coletivo é a criação de projetos experimentais, através da fricção entre o digital e o analógico.

Entre a ciência, arte e o design, eles desenvolveram uma série de instalações, produtos e tecnologias que reinterpretam a natureza “fria” do digital e enfatizam o analógico, examinando de perto as qualidades comportamentais humanas em ambientes e objetos inanimados. Com diversos projetos com um carácter social profundo, já exibiram internacionalmente os seus trabalhos em espaços como o MoMa, sendo por isso também considerados pioneiros na área do creative coding.

*Audience* é um projeto de 2008 que utiliza um software de deteção de movimento para criar uma experiência antropomórfica até certo ponto desconfortável. Consiste numa instalação concebida pela Random International e executada em colaboração com Chris O'Shea, encomendada pelo coreógrafo Wayne McGregor para o *Deloitte Ignite Festival* na Royal Opera House London em Setembro de 2008.

Quando se entra no espaço encontramos 100 espelhos espalhados pelo espaço que giram aleatoriamente. No momento em que estes objetos detetam o nosso movimento fixam-se instantaneamente na nossa direção, tornando desse modo o utilizador o ponto de focagem e de observação. Os espelhos, que metaforicamente representam pessoas, focam-se na pessoa que entra no espaço até que se desinteressem, procurando de seguida outro ponto de focagem. O comportamento coletivo repentinamente sincronizado dos objetos está além do controlo do observador, pois fica inteiramente a seu critério. A instalação visava inverter os papéis do espectador e do visto, fazendo com que algumas pessoas se pudessem sentir ignoradas e excluídas de serem o centro das atenções/de observação, criando uma sensação de desconforto e inquietação.

## fuse\* (2007)

fuse\* foi fundado em 2007 por Luca Camellini e Mattia Carretti e é um estúdio que desde o seu início se concentrou na fusão da ciência, com a tecnologia e a arte, para desenvolver novas linguagens e heurísticas para discutir a natureza humana e criar experiências artísticas emocionalmente carregadas que dão ao público um sensação de conexão e coletivização por meio da participação. Focam-se na criação de obras multimédia e live performances que interligam elementos virtuais, artísticos e arquitetónicos, explorando as possibilidades ilimitadas das tecnologias digitais emergentes, enquanto amplificam o impacto emocional da narrativa no espectador. As obras são criadas para inspirar as pessoas, ultrapassar limites e procurar uma nova interação entre a luz, o espaço, o som e o movimento. Esta exposição individual de fuse\*, *Everything in Existence*, que estreou em Washington DC em 2019, cria experiências digitalmente interativas independentemente do artista. A sua natureza autossuficiente e generativa sugere uma forma inteiramente nova de expressão artística.

## Marpi Studio (2019)

Marpi Studio é um estúdio formado em 2019 pelo artista Marpi, cujo foco é a criação de instalações que gerem ambientes de co-criação entre o seu próprio projeto e o público envolvente. O resultado dos seus projetos é sempre inesperado devido ao facto de girar em torno das interações e gestos do público.

*New Nature Digital Petting Zoo* é uma instalação que representa um zoo interativo em torno de organismos virtuais com deteção de movimento em tempo real. O espaço contém criaturas e plantas que necessitam da interação física dos espectadores para ganharem dimensão, simulando movimento como se estas mesmas criaturas/plantas estivessem a ser acarinhadas.

### (3.2) Ferramentas

Para iniciar o projecto foi necessário perceber de forma global quais as ferramentas que havia ao dispor para a concretização do mesmo. No capítulo seguinte de forma não muito extensa, são enumeradas as ferramentas mais utilizadas na prática do creative coding, bem como alguma informação relevante sobre cada uma, enumerando também as suas vantagens e desvantagens e as vertentes em que cada uma delas é mais utilizada.

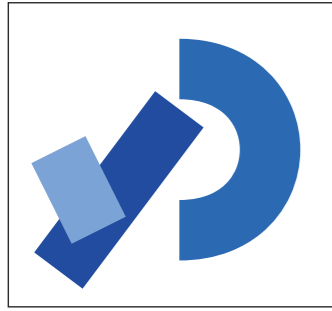


Fig 13 - Processing

### (3.2.1) Processing

Through this time, we've promoted software literacy, particularly within the visual arts, and visual literacy within technology. Initially created to serve as a software sketchbook and to teach programming fundamentals within a visual context, Processing has also evolved into a development tool for professionals. The Processing software has always been free and open source and has always run on Mac OS, Windows, and Linux.

(Fry B., Reas C., n.d).

Por ser um software gratuito, de código aberto, que promove a participação e colaboração da comunidade para o seu crescimento e contributo com código e bibliotecas de código, bem como o facto de ser um IDE (Integrated Development Environment) onde o utilizador pode ter acesso ao editor de texto e a uma janela para visualizar o resultado gráfico gerado, foi o software escolhido para a fase inicial de experimentação neste projeto. Para além de ser um software no qual já estava familiarizada, bem como pela sua linguagem de programação (Java).

O Processing permite-nos acompanhar os resultados visuais ao longo das fases de programação, dando uma maior consciencialização do processo criativo e domínio do código.

A simplicidade da sua linguagem de programação permite que seja utilizado por indivíduos mais iniciantes ou avançados, levando a resultados com um grau de maior ou menor complexidade.

### (3.2.2) p5.js

p5.js e Processing parecem em tudo idênticos, no entanto, p5.js é uma biblioteca de JavaScript, contrariamente ao Processing que é um ambiente e linguagem de programação. Este software possui diversas funcionalidades para a implementação de creative coding no contexto do desenvolvimento Web, sendo que esta biblioteca permite integrar conceitos de design generativo na prática de Web Design. Elementos como texto, vídeo e imagem podem ser criados e podem ser geradas soluções visuais generativas ou paramétricas a partir dele. A biblioteca p5.js, sendo uma ferramenta que pode ser disponibilizada online e acessível em qualquer browser, permite que as ferramentas ou soluções desenvolvidas pelos programadores criativos tenham um alcance maior relativamente aos seus visualizadores/utilizadores e número de dispositivos. O Processing é um ambiente baseado na linguagem de programação Java, enquanto que o

p5.js é uma biblioteca baseada na linguagem de programação Javascript. Este ambiente e linguagem de programação, por ter estas características, torna-se mais utilizado para desenvolvimento de soluções e ferramentas web, ao invés do Processing que é mais utilizado numa vertente estática e de animação.



Fig 14 - p5.js

### (3.2.3) Outras Ferramentas/Linguagens de Programação

#### Drawbot

DrawBot é uma aplicação gratuita para MacOSX que incentiva a escrita e aprendizagem de Python simples para gerar gráficos bidimensionais. Estes gráficos podem conter formas como retângulos, ovais, curvas (bezier), polígonos, objetos de texto e transparência. DrawBot é uma ferramenta ideal para ensinar os fundamentos da programação, pois há uma espécie de valorização (com atributo de prémios) que os alunos recebem enquanto os mesmos se familiarizam com variáveis, declarações condicionais e funções. Os resultados podem ser guardados numa variedade de formatos de arquivo designadamente PDF, svg, filme, png, jpeg, tif.

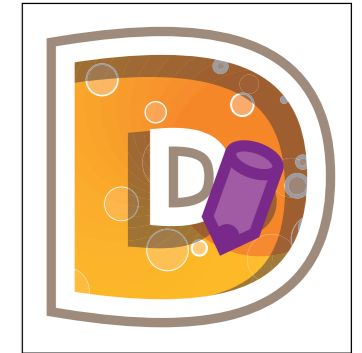


Fig 15 - Drawbot

#### Html, Css e Javascript

Apesar de não serem softwares, mas sim linguagens de programação, como já referido nos exemplos anteriores, merecem também aqui o seu destaque pela liberdade criativa ilimitada que dão ao utilizador para desenho de interfaces e respectivas animações/interações. Enquanto que o Html é uma linguagem de marcação utilizada na construção de páginas na Web, que podemos considerar como a estrutura base/esqueleto de qualquer página web, o CSS, Cascading Style Sheets, é um mecanismo para adicionar estilos a uma página web, ou seja, colocar de forma visual os elementos de Html, e o Javascript é uma linguagem de programação interpretada de script, que atribui dinamismo e animação aos elementos. Estes três elementos são as três linguagens mais utilizadas no contexto da Web. É importante ressaltar a importância do Javascript, dado que tanto o Processing como o p5.js tem a sua base nesta mesma linguagem.

#### Max

Max é uma linguagem de programação visual mais direcionada para a vertente da música. Foi desenvolvida originalmente na década de 1980 por Miller Puckette, e hoje é mantida pela Cycling '74, uma empresa americana de desenvolvimento de software fundada em 1997 por David Zicarelli, com sede em San Francisco. Foi utilizada inicialmente por compositores e produtores de música.

Max includes full-featured, expandable video and graphics tools with Jitter. Jitter is optimized for realtime audiovisual work, and is easy to combine with audio, sequencing, and modulation like everything else in Max.

(What is Max?, Cycling '74., 2020)



Fig 16 - Max



Fig 17 - Open Frameworks

## Open Frameworks

O openFrameworks foi criado em 2005 por Zachary Lieberman. Este é uma ferramenta de código aberto C++ (linguagem de programação), que tem como objetivo auxiliar no processo criativo, atribuindo estruturas simples promovendo a experimentação. O foco desta ferramenta é o da programação de aplicações multimédia, bem como o de sistemas interativos que envolvam áudio, vídeo e imagem. É uma ferramenta opensource (código aberto) que tem vindo a ser desenvolvida publicamente e de forma colaborativa.

openFrameworks is designed to work as a general purpose glue, and wraps together several commonly used libraries, including:

- OpenGL, GLEW, GLUT, libtess2 and cairo for graphics
- rtAudio, PortAudio, OpenAL and Kiss FFT or FMOD for audio input, output and analysis
- FreeType for fonts
- FreeImage for image saving and loading
- Quicktime, GStreamer and videoInput for video playback and grabbing
- Poco for a variety of utilities
- OpenCV for computer vision
- Assimp for 3D model loading

(openFrameworks. (n.d.))

Esta ferramenta tem compatibilidade com cinco sistemas, Windows, OSX, Linux, iOS, Android e quatro IDEs, XCode, Code::Blocks, Visual Studio e Eclipse. Foi desenvolvido com o intuito de ser colaborativo, simples, consistente, intuitivo e extensível. Desenvolve-se com as contribuições de várias pessoas, seja a título individual, seja em projetos colaborativos. Esta colaboração acontece no Github (plataforma de hospedagem de código), que permite que experimentemos o código, acrescentando-lhe ou não mais ramificações.

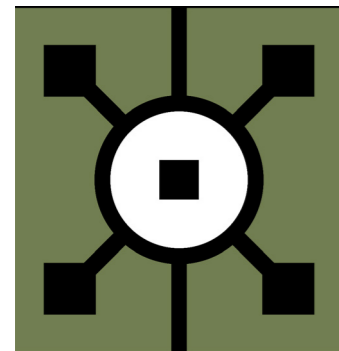


Fig 18 - TouchDesigner

## TouchDesigner

O TouchDesigner foi fundado por Greg Hermanovic, Rob Bairos e Jarrett Smith em 2002, e é um software de linguagem de programação visual que permite a criação de conteúdos multimédia interativos.

Devido à sua base assente na experimentação e interação é uma ferramenta flexível e com uma prototipagem rápida e, por isso, é um software utilizado recorrentemente para Live Performances.

Contém uma linguagem VPL (Visual Programming Language) onde o utilizador manipula elementos gráficos que executam determinada tarefa já definida. Contrariamente à programação com código escrito, neste caso conseguimos construir projetos onde estabelecemos ligações entre elementos que reagem a certos inputs, produzindo o resultado final.

Neste software existe um menu com funções/elementos gráficos estabelecidos, onde, através da diferente junção dos mesmos, obtemos resultados visuais distintos. Existem muito outros softwares de creative coding, que serão apenas referenciados aqui, Cinder, generativepy, Max MSP, Nannou, OpenFrameworks, OPENRNDR, Pure Data, Shoebot, SuperCollider, Vvvv e Zim, que não mereceram o mesmo desenvolvimento que os anteriores por nunca ter trabalhado neles.

# (4) PROJETO

#### (4.1) Introdução da Proposta

Este projeto, além de procurar entender quais são as capacidades do creative coding no contexto no design gráfico, visa chamar a atenção de alunos e professores da área de design para os diversos aspectos práticos inerentes à codificação criativa.

From a design and art standpoint, programming was recently considered enfant to terrible. She was the kid who just didn't want to be controlled.

(Rodenbröker T., 2022)

A programação, por ser algo desconhecido, era considerado algo errático, descontrolado. Enquanto designers, estamos habituados a ferramentas que se adaptam às nossas necessidades e não o oposto. Daí, até há uns tempos, cingimos a programação apenas às nossas necessidades, e de forma muito controlada essencialmente com o objetivo de resolução de problemas mais operacionais e não necessariamente como uma forma de expressão criativa. Estas ferramentas, que normalmente usamos quando fazemos design gráfico, designadamente InDesign, Illustrator, Photoshop, After effects, são utilizadas através de um interface gráfico, ao contrário da programação criativa onde a produção de resultados é feita através da escrita. Esta diferenciação na forma de interagir com a ferramenta faz com que a sua aplicação tenha sido explorada apenas por um número mais reduzido de designers. No entanto, com a evolução tecnológica foi necessário compreender e estudar melhor os softwares, códigos, algoritmos, as diferentes vertentes que este mundo da programação pode tomar, e a potencialidade que tudo isto tem na prática, para profissões das áreas criativas e não só. A aplicação e progresso do creative coding sentiu-se sobretudo no design de comunicação, onde ao longo dos anos tem ganho algum destaque. No âmbito do programa de estudos de 2º ciclo em Design de Comunicação, e dado um panorama de estudos anteriores em programação e desenvolvimento web, faz sentido um projeto a fundir as duas vertentes.

#### (4.2) Contributo

As a teacher, I would like to dedicate myself to this task to allow a new look at code and programming, to make the inherent creative potential of both recognizable and to make them attractive to a young audience with the help of graphic design.

(Rodenbröker T., 2022)

O contributo deste projeto reside na reflexão do papel do designer, como criador de ferramentas e sistemas de design através da programação. O futuro do design tendencialmente desenvolve-se na criação destas ferramentas ao invés do foco exclusivo na criação de soluções estáticas, não generativas e não paramétricas, mais comuns nos outros processos de design já estabelecidos.

Podemos refletir, ainda, que qualquer que seja o nosso desenvolvimento na criação da ferramenta, está intrínseco o nosso papel enquanto designers e os inputs que nela atribuímos, e, por isso, nunca nos distanciamos do design em si.

Esta prática permite-nos obter resultados inesperados ao invés de premeditadamente pensados. Para além disso, a criação duma ferramenta permite colocar em segundo plano o resultado final em si, e pensar nos processos e sistemas que o constituem. Contrariamente a outros softwares de conhecimento geral de designers/artistas, como o Adobe Indesign, Adobe Illustrator, Adobe Photoshop, After Effects, cinema 4D, em que a generalidade das ferramentas e funcionalidades dos softwares estão disponíveis a priori, tendo nós, enquanto utilizadores a liberdade de os utilizar e misturar de forma a produzir algum tipo de resultado, no creative coding essas ferramentas e funcionalidades têm de ser pensadas e desenvolvidas pelos designers. Este conceito/método permite-nos distanciar um pouco do aspeto estético/resultado final, pelo menos numa fase inicial, e permite-nos pensar no sistema, processo e pensamento da máquina/código para execução das ordens atribuídas.

O projeto também permite uma reflexão do posicionamento do designer hoje, bem como a de um enriquecimento e esclarecimento de vocabulário inerente ao tema, muitas vezes confundível ou mal empregue em algum contexto.

**(4.3) Processo**

O processo de desenvolvimento do projeto, inicialmente, começou pelo desenvolvimento de um calendário com as diversas fases do mesmo. Esta organização inicial das diversas fases foi essencial para o desenvolvimento do projeto. A primeira fase seria a de recolha de definições sobre os conceitos como arte, arte generativa, sistemas generativos, como estes se aliam ao design, e como permitem criar elementos visuais, bem como a recolha de literatura relativa às temáticas mencionadas. Aliada a esta recolha de conceitos teóricos estaria também uma recolha e análise de projetos que explorassem a questão da imprevisibilidade e aleatoriedade como parte do processo criativo, aliado à interseccionalidade de áreas e ferramentas.

Numa segunda fase, e após a recolha de toda a informação acima descrita, formalizou-se um glossário. Este glossário teria as diversas terminologias encontradas ao longo de toda a pesquisa, de modo a ficar esclarecido para a comunidade criativa todos os diversos termos empregues neste projeto, como em outros tantos. Dado que por diversas vezes certos termos são empregues de forma incorreta e de facto não existe uma uniformização dos mesmos, bem como a sua tradução de inglês para português e vice-versa, sentiu-se a necessidade de execução deste glossário para pesquisas e projetos futuros.

Numa terceira fase, esta já mais direccionada para a parte prática e projectual, realizar-se-ia uma série de experiências visuais das quais, após uma seleção, resultariam os elementos empregues na ferramenta final. Por fim, numa quarta fase, implementar-se-iam as melhores soluções da fase anterior, de forma a constituir uma ferramenta com o intuito de colocar em prática os conceitos explorados e desenvolver as possibilidades de creative coding no contexto do design.

**(4.3.1) Referências**

**Generative System Poster (2007)**

*Generative System Poster* resulta duma experiência com elementos generativos, onde no final só se poderia utilizar um número limitado de elementos. Resultou numa série de cartazes, construídos com Indesign Scripting Interface e Processing, onde o foco era conectar todas as palavras de uma linha com todas as palavras da linha seguinte, bem como a de construir grelhas flexíveis, com um único script.

**Font Gauntlet (2019)**

*Font Gauntlet* é uma ferramenta para gerar e animar fontes (variáveis), publicada por Dinamo, de uso gratuito. Permite que enquanto utilizadores possamos testar as tipografias e perceber a dinâmica e funcionalidade das mesmas. Também contém publicados tutoriais sobre tipografia, software e tipos de softwares através do site.

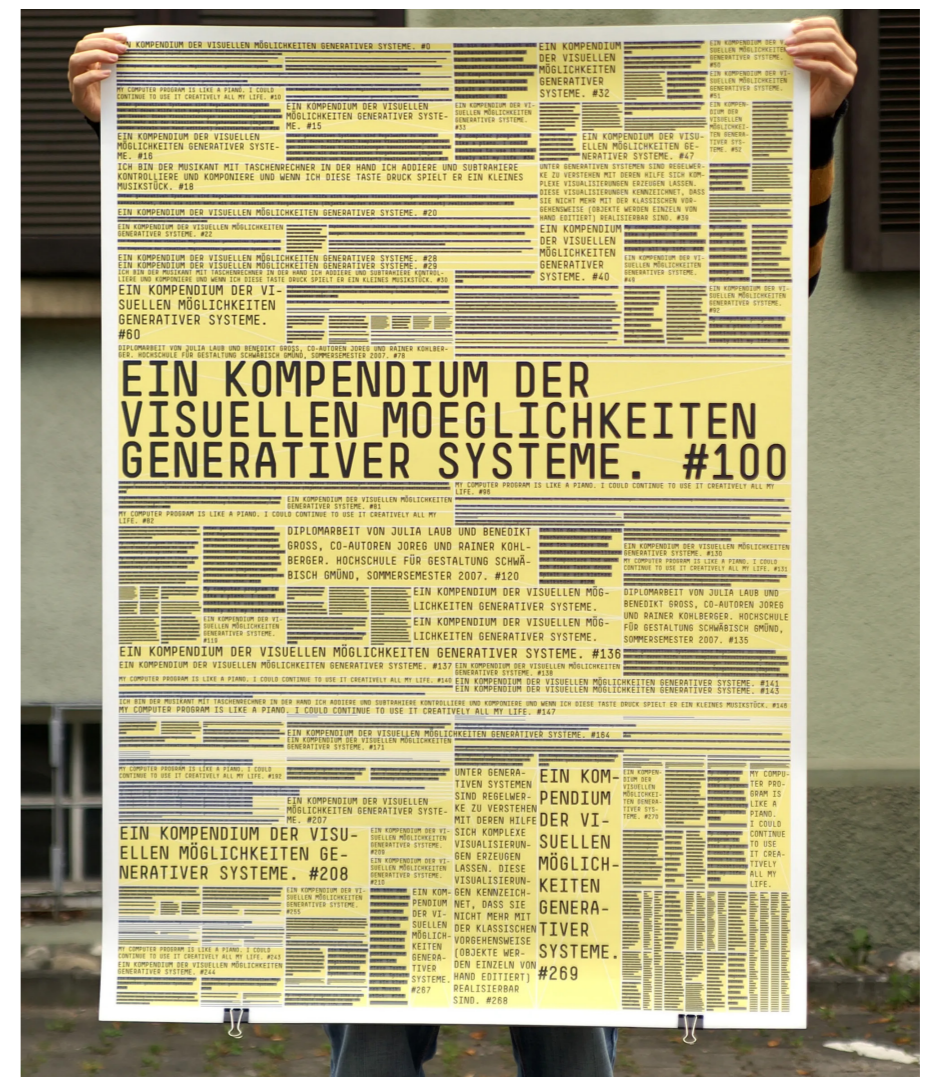


Fig 19 - Generative System Poster por Benedikt Gross

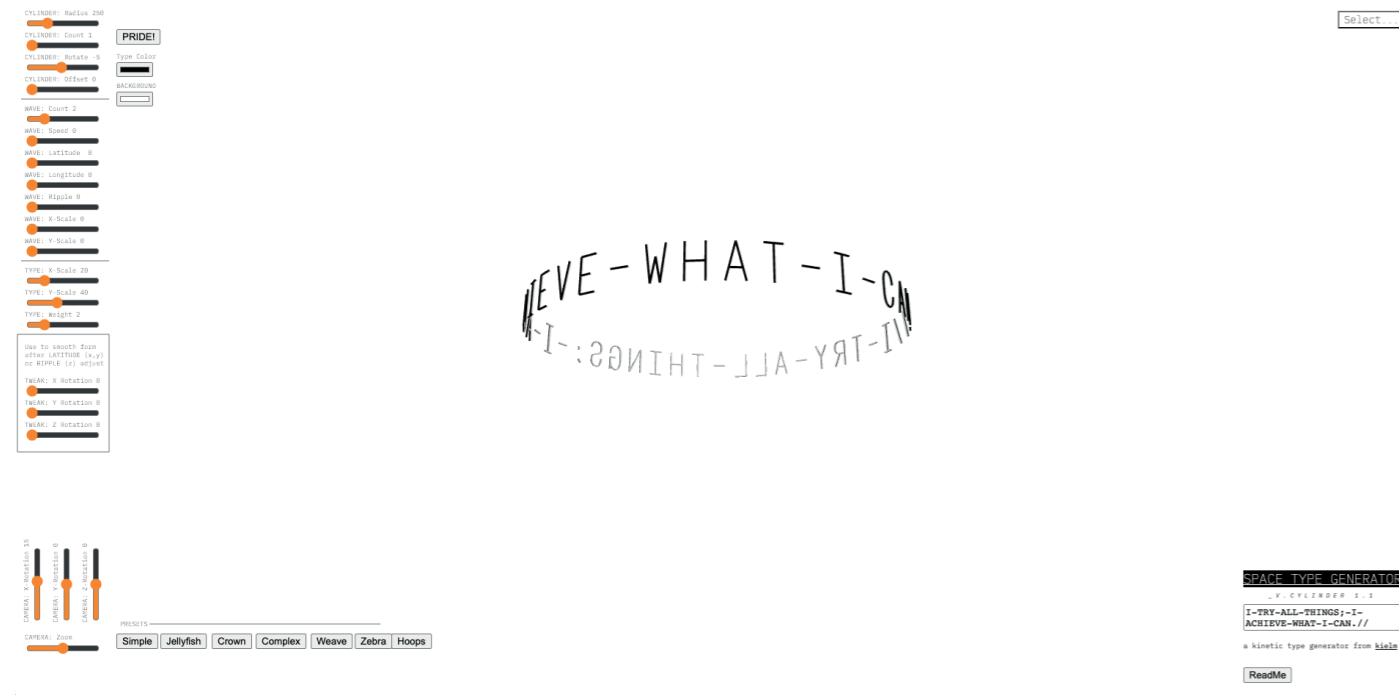


Fig 20 - Space Type Generator por Kiel Mutschelknaus



Fig 21 - Space10 Generative App

## Space10 Generative App (2020)

*Space10 Generative App* é um projeto desenvolvido pelo estúdio e laboratório de investigação e design SPACE10 como ferramenta para o desenvolvimento da própria identidade visual. Esta ferramenta possibilita a construção de layouts completos, com imagens e textos em tempo real bem como a exportação de vídeos, gifs, png's, etc.

## Variantype.tool / Posters: Deconstructing Iterations in p5.js (2021)

*Variantype.tool* é uma ferramenta desenvolvida por Vânia Oliveira no âmbito do seu projeto final de mestrado em Design Gráfico e Projetos Editoriais, na Faculdade de Belas Artes da Universidade do Porto), que permite editar e interagir com tipografia estática ou cinética e produzir resultados visuais de design editorial ou gráfico, como capas de livros e/ou posters.

## Kevin Kosh (2023)

No âmbito do desafio lançado por Tim Rodenbröcker, *Programming Posters*, Kevin Kosh, atualmente a estudar na Fachhochschule Dortmund – University of Applied Sciences, Alemanha, desenvolve uma série de pequenos projetos construindo poster's / pequenas animações que conseguem ser alteradas em tempo real através dum sistema de parametrização (controlP5).

## Space Generator (2023)

Space Generator é um gerador de diferentes tipografias e disposições/ animações da mesa. Foi criado por Kiel Mutschelknaus, um artista multidisciplinar cujo trabalho abrange design, animação, tipografia e, recentemente, o creative coding. Esta ferramenta oferece diferentes parâmetros de animações, que é possível personalizarmos como as cores, a velocidade, o texto e a animação em si.

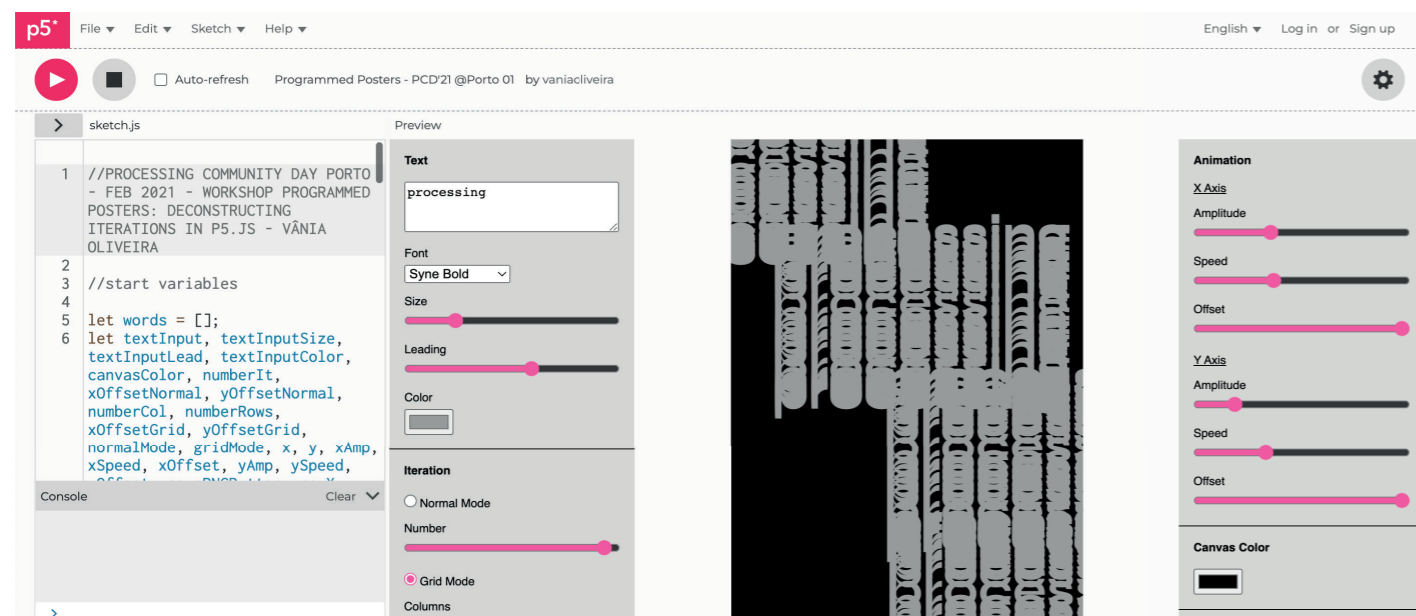
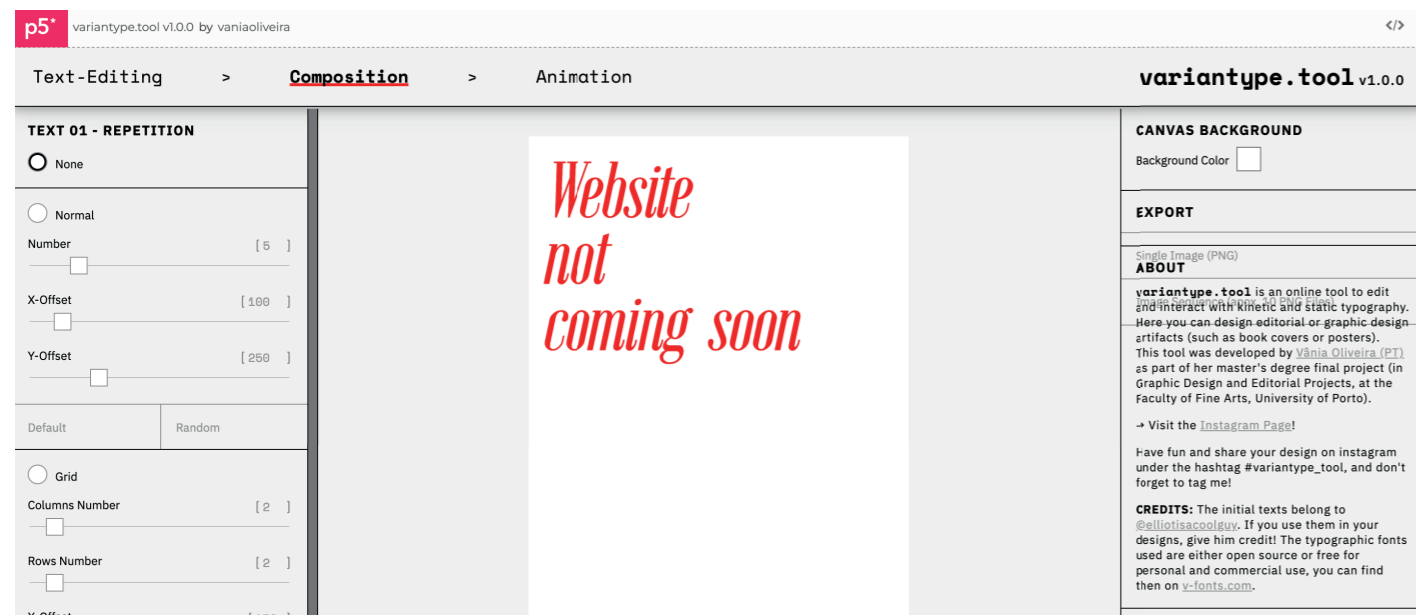


Fig 22- Variantype.tool por Vânia Oliveira

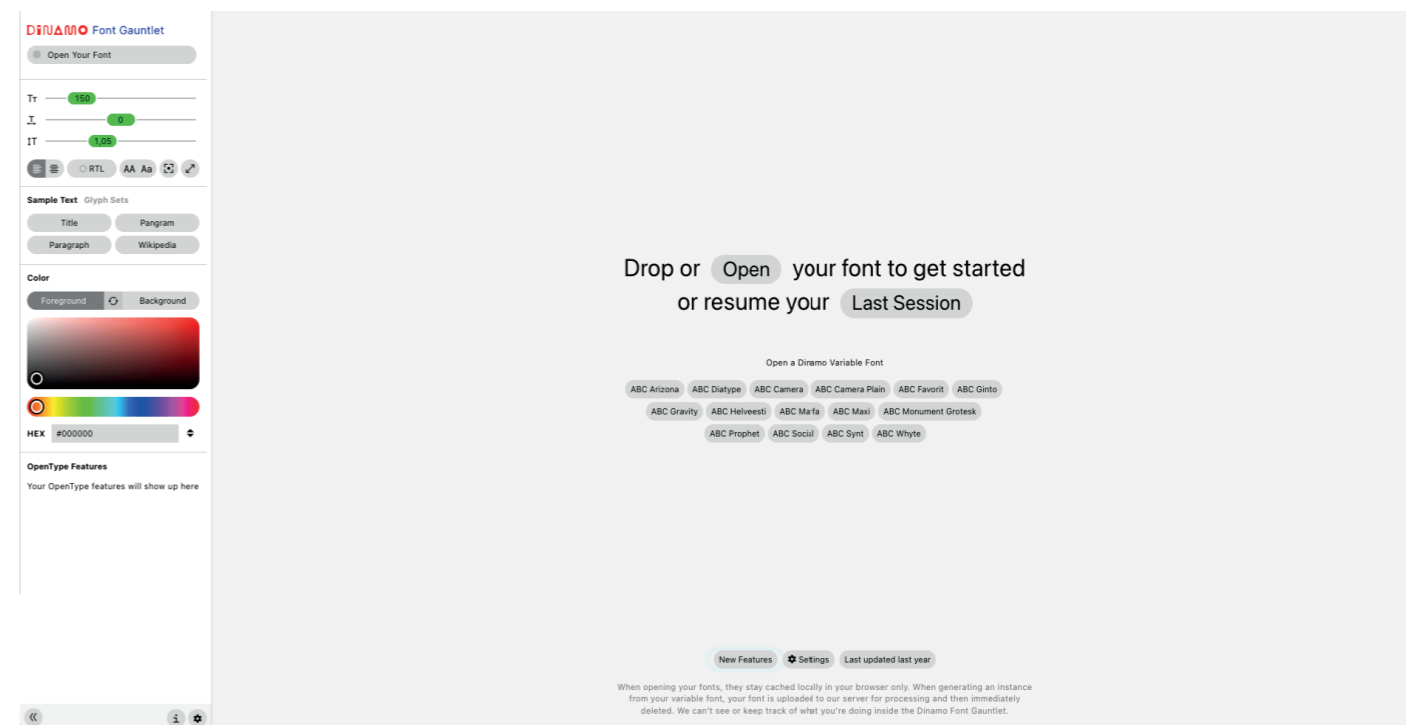


Fig 23 - Font Gauntlet por Dinamo

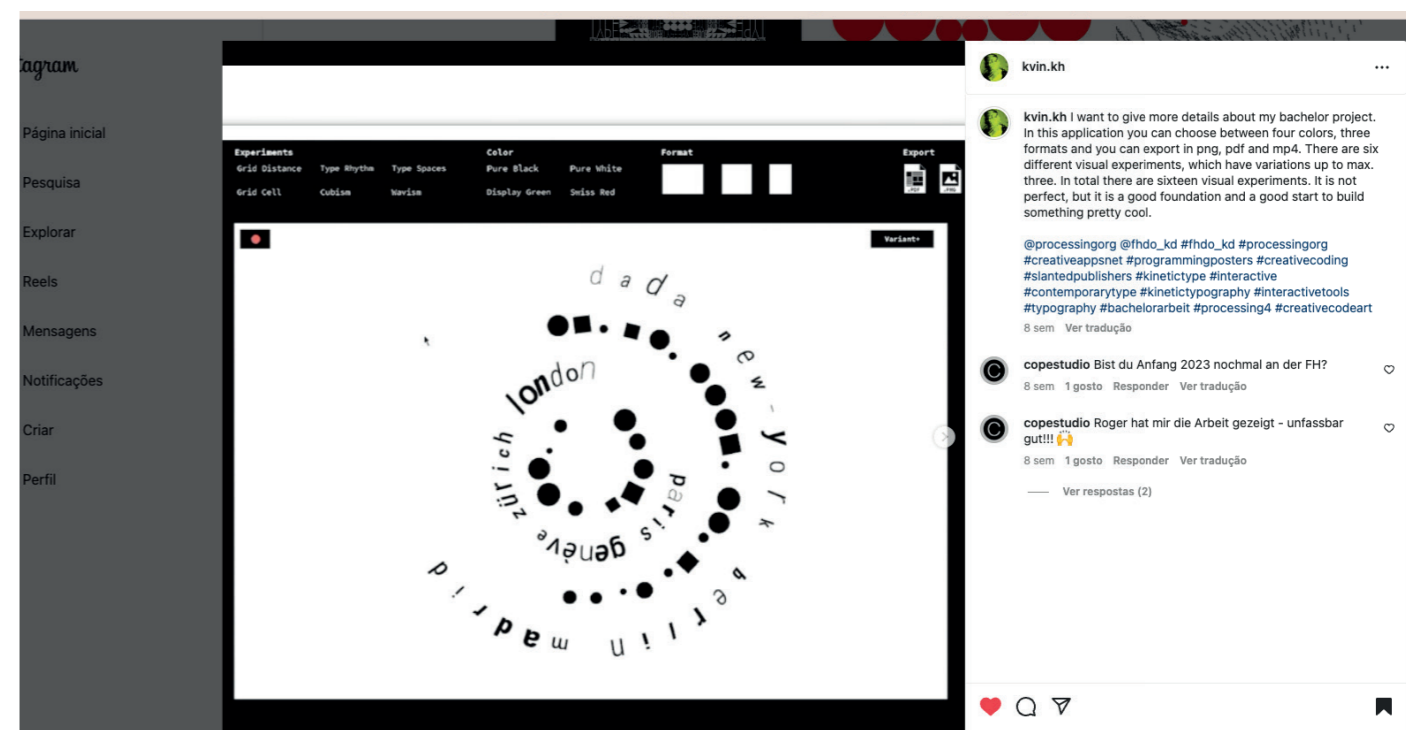


Fig 24 -Kevin Kosh

### (4.3.2) Desenvolvimento dos componentes

Como parte da aplicação das boas práticas de programação, foi iniciado um plano de divisão dos diferentes conceitos a serem trabalhados em Processing. (Fig. x)

Inicialmente foi feito um estudo isolado de cada um destes conceitos, pequenas experiências visuais, que demonstrassem o potencial de cada um, bem como da sua aplicação no âmbito do Design gráfico e permitissem o desenvolvimento de uma literacia maior no que toca ao ambiente de desenvolvimento e linguagem de programação Processing.

Os componentes enunciados foram: coordenadas (x, y), formas 2D e 3D, formas complexas (vértices), Cores, Gradientes, Sobreposições e Transparências, texto / tipografia, Imagem / Pixeis, Random, Cálculos (sommas, divisões), Animações, Repetições (For, Loop, While), Condições (If, Else), Interatividade, Motion (Velocidade, Direção, Tempo e Força), transformações, Noise, botões de parametrização e Rendering / Exportação, e cada um destes possui um ou mais ensaios visuais das suas possibilidades. A listagem destes componentes partiu do livro *Processing, A Programming Handbook for Visual Designers and Artists*, de Casey Reas e Ben Fry, bem como de toda a documentação que se encontra disponível online na própria plataforma do Processing. A criação destas experiências visuais começou por ser isolada, no sentido de fazer explorações mais profundas e que pudessem tirar conclusões sobre os conceitos anteriormente definidos. Posteriormente houve a tentativa de progressão de desenvolvimento gráfico de uns para os outros, bem como o cruzamento destes conceitos ou certas funcionalidades de uns para os outros. Por exemplo, sendo o primeiro exercício de coordenadas, e o segundo de formas 2D, no segundo exercício as formas em si já são colocadas numa estrutura de coordenadas. Nas Sobreposições e Transparências trabalhou-se a adição e sobreposição de cores (transparências e alterações de cor).

No seguinte, de tipografia, elaborou-se uma possível composição tipográfica, com algumas sobreposições/transparências trabalhadas no exercício anterior. Depois de realizadas pelo menos uma experiência visual para cada um dos componentes, houve uma revisão dos mesmos de modo a otimizar o código, por exemplo usando variáveis em vez de valores absolutos, no sentido de tornar a ferramenta mais ajustável a diferentes formatos (responsividade). Todos os valores que fossem de certa forma “aleatórios” foram colocados sobre uma variável que seria mapeada em relação ao valor de altura e largura (width e height), de forma a garantir uma posição dos elementos desenhados dentro da área de visualização. Além de permitir que os resultados se ajustem melhor a diferentes formatos e suas proporções, esta otimização permitiu diminuir a extensão do código e o tempo necessário para a sua manipulação.

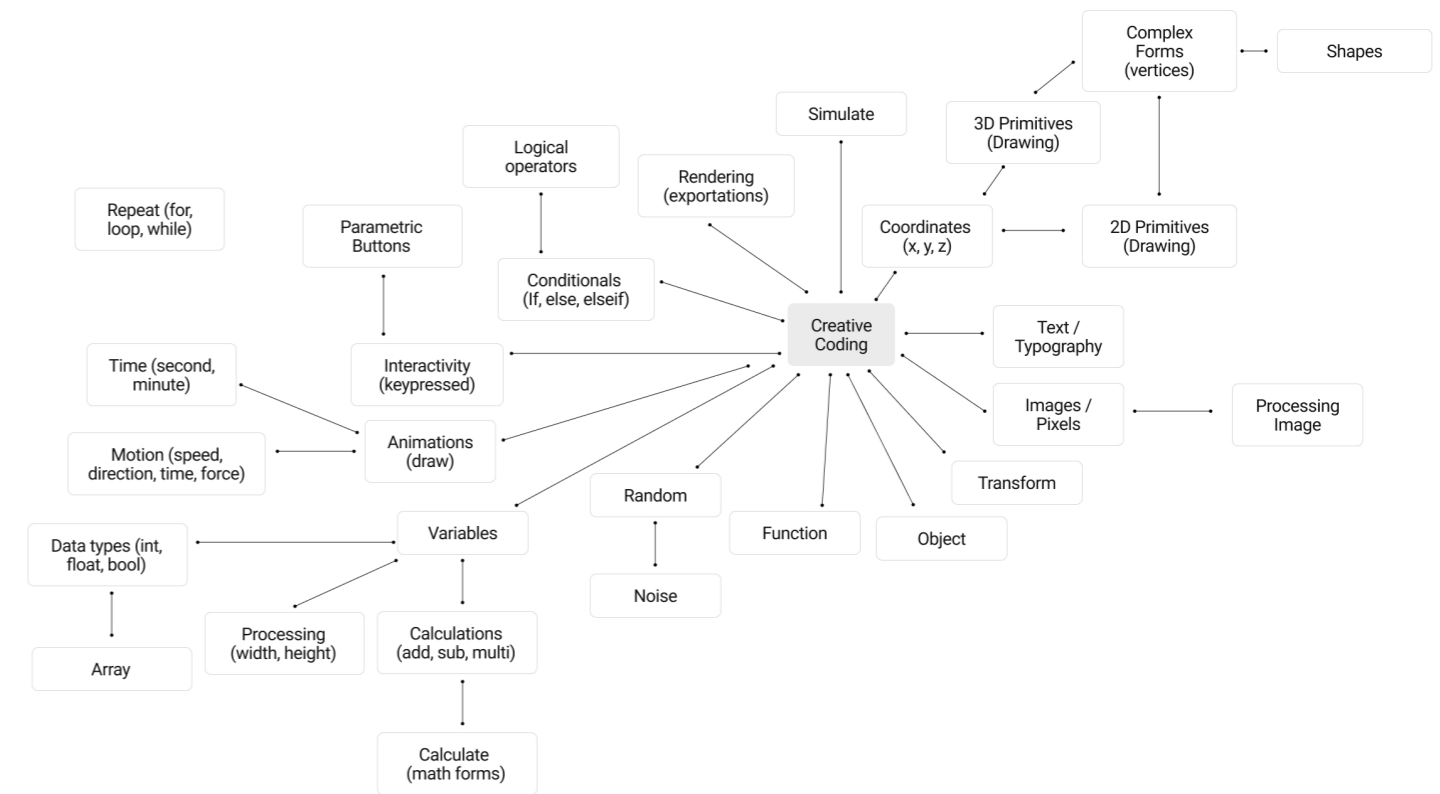


Fig 25 - Mapa inicial de estruturação dos componentes a desenvolver

```

void setup (){
  int x=0;
  int y=0;
  size(400,400);
  while(x< width){
    rect(x, 0, 20,20);
    x= x+20;
  };

  while(y< height){
    stroke(153);
    //passar o primeiro valor para uma variável
    rect(0, y, 20,20);
    rect(20, y, 20,20);
    rect(40, y, 20,20);
    rect(60, y, 20,20);
    rect(80, y, 20,20);
    rect(100, y, 20,20);
    rect(120, y, 20,20);
    rect(140, y, 20,20);
    rect(160, y, 20,20);
    rect(180, y, 20,20);
    y= y+20;
  };
};

void draw (){
  //point
  strokeCap(ROUND);
  strokeWeight(4);
  point (20,20);
  point (260,360);
  point (300,200);
  point (20,260);
  point (80,160);

  //text
  PFont font;
  font = createFont("ABCDiatypeMono-Light-Trial.otf", 14);
  textFont(font);
  text("20,20", 30, 35);
  text("260,360", 265, 375);
  text("300,200", 310, 215);
  text("20,260", 30, 275);
  text("80,160", 90, 175);

  fill(0);

  //line
  strokeWeight(1);
  stroke(0);
  line(20,260,80,160);
};

```

Fig 26 - Código inicial

```

int x=0;
int y=0;
//text
PFont font;
int size = 20;

void setup (){
  stroke(200);
  size(800,800);
  pixelDensity(displayDensity());
  font = createFont("ABCDiatypeMono-Light-Trial.otf", 14, true);

  for (int x = 0; x < width; x = x+size) {
    for (int y = 0; y < height; y = y+size) {
      rect(x, y, size,size);
    }
  }

  //point
  stroke(0);
  strokeCap(ROUND);
  strokeWeight(4);
  point (20,20);
  point (260,360);
  point (300,200);
  point (20,260);
  point (80,160);

  textFont(font);
  fill(0);
  text("20,20", 30, 35);
  text("260,360", 265, 375);
  text("300,200", 310, 215);
  text("20,260", 30, 275);
  text("80,160", 90, 175);

  //line
  strokeWeight(1);
  stroke(0);
  line(20,260,80,160);

  save("export/coordinates.jpg");
};

```

Fig 27 - Código depois de adaptado de forma a promover a responsividade

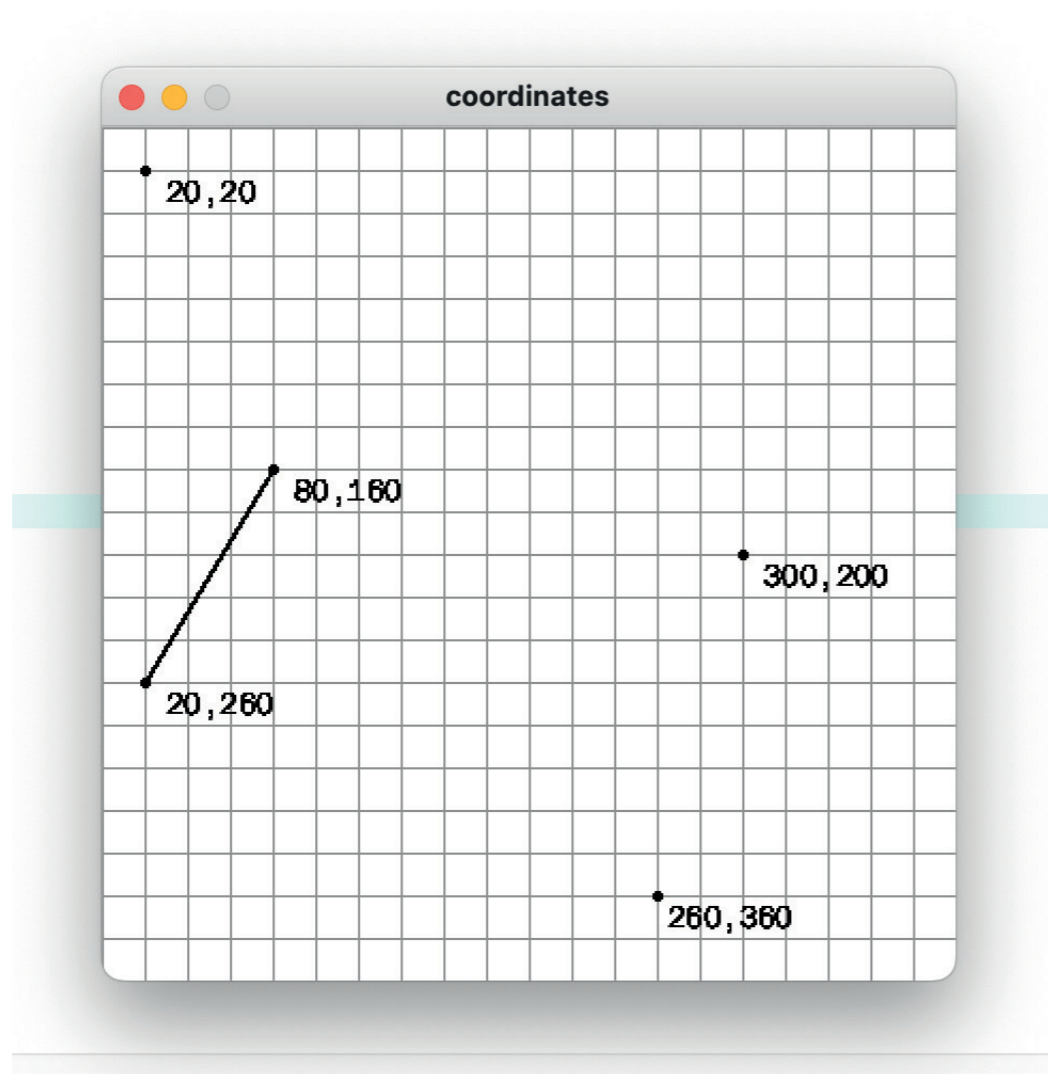


Fig 28 - Experiência Visual das Coordenadas (x,y)

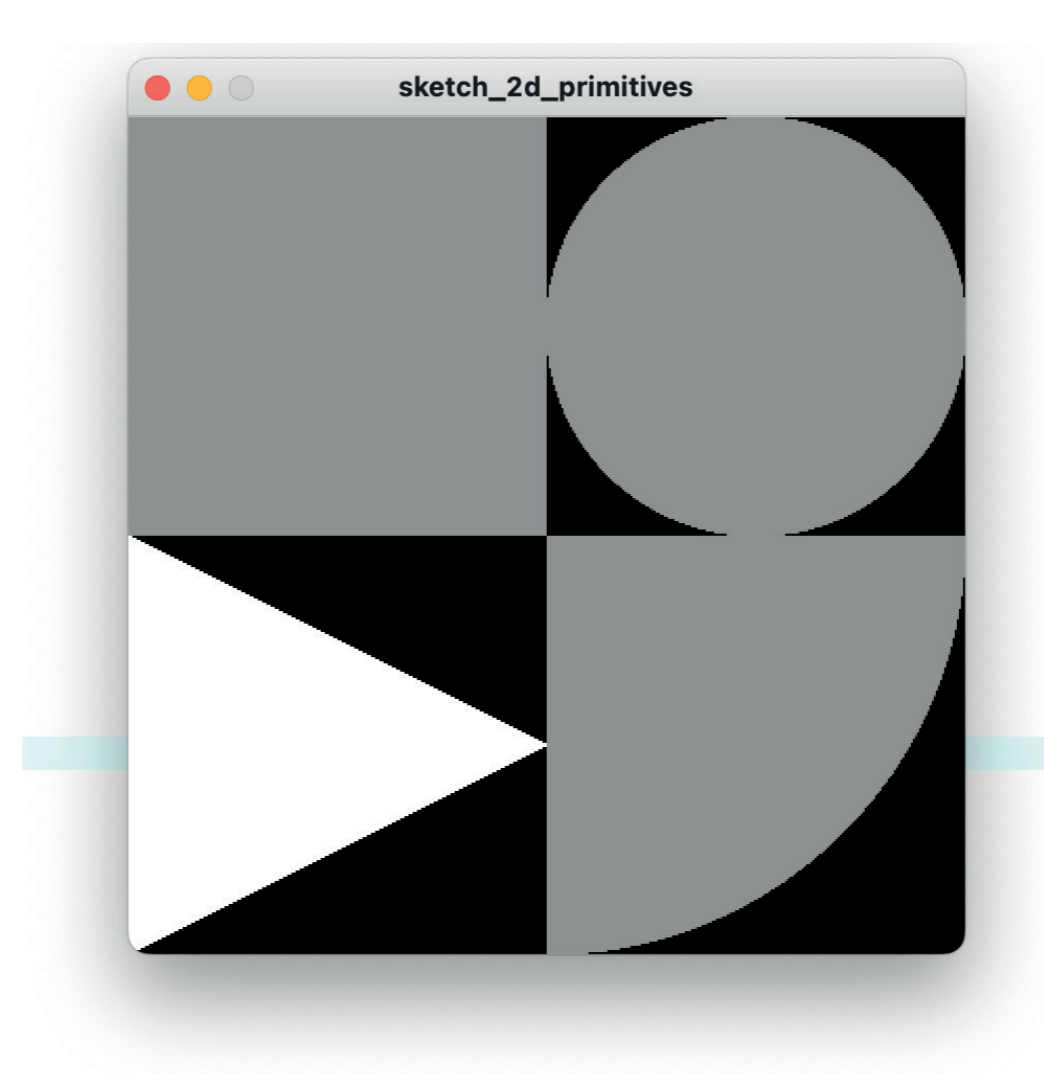


Fig 29 - Experiência Visual das formas 2D

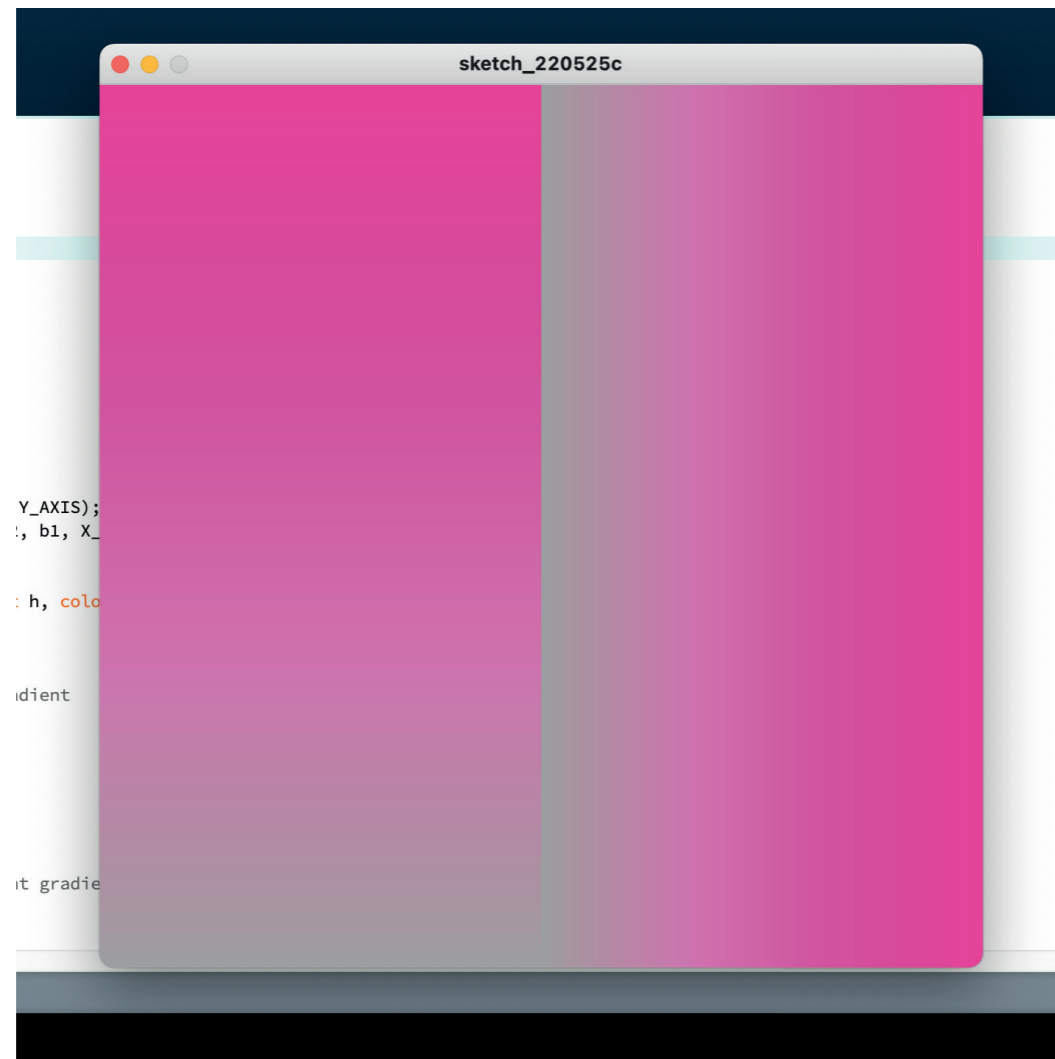


Fig 30 - Experiência Visual dos gradientes

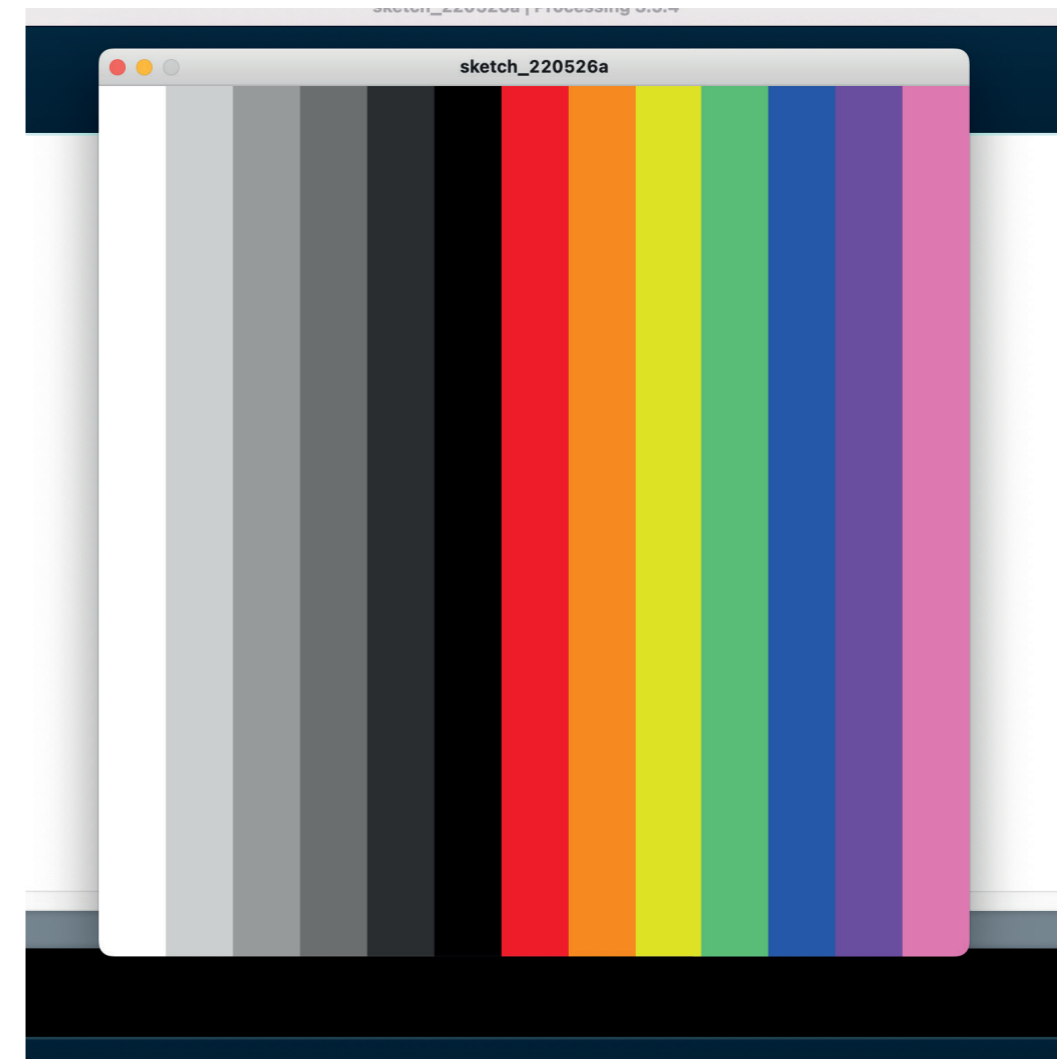


Fig 31 - Experiência Visual das cores

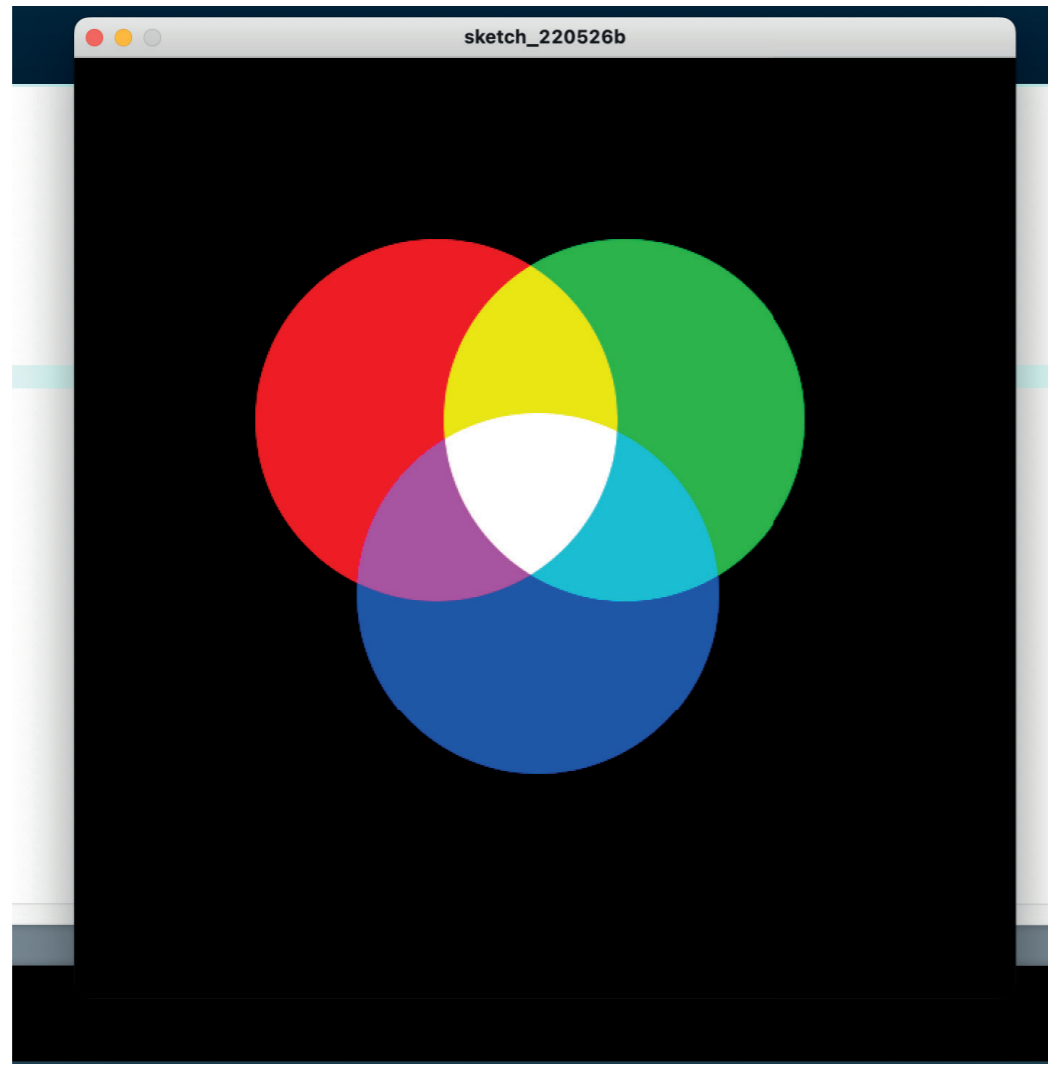


Fig 32 - Experiência Visual das sobreposições e transparências

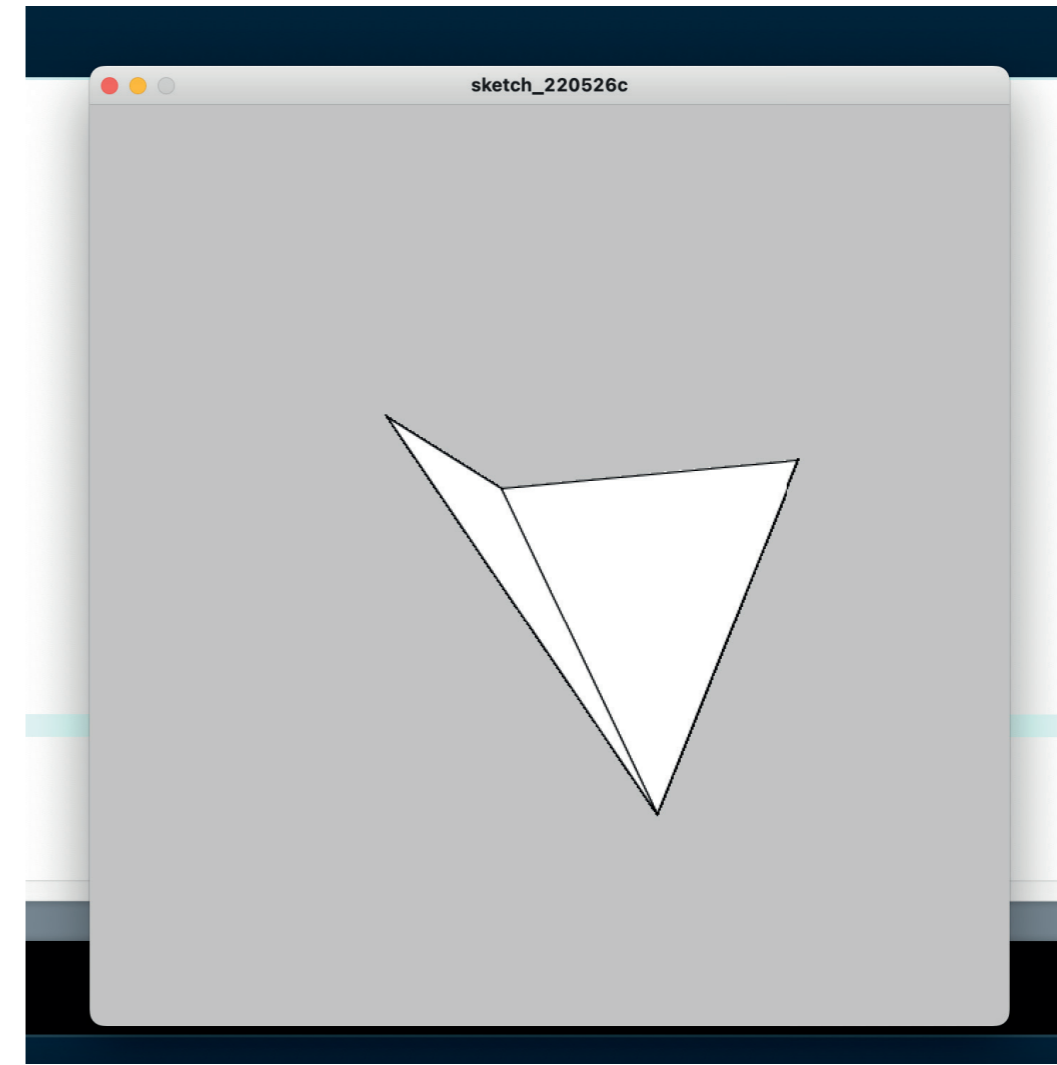


Fig 33 - Experiência Visual das formas complexas (vértices)

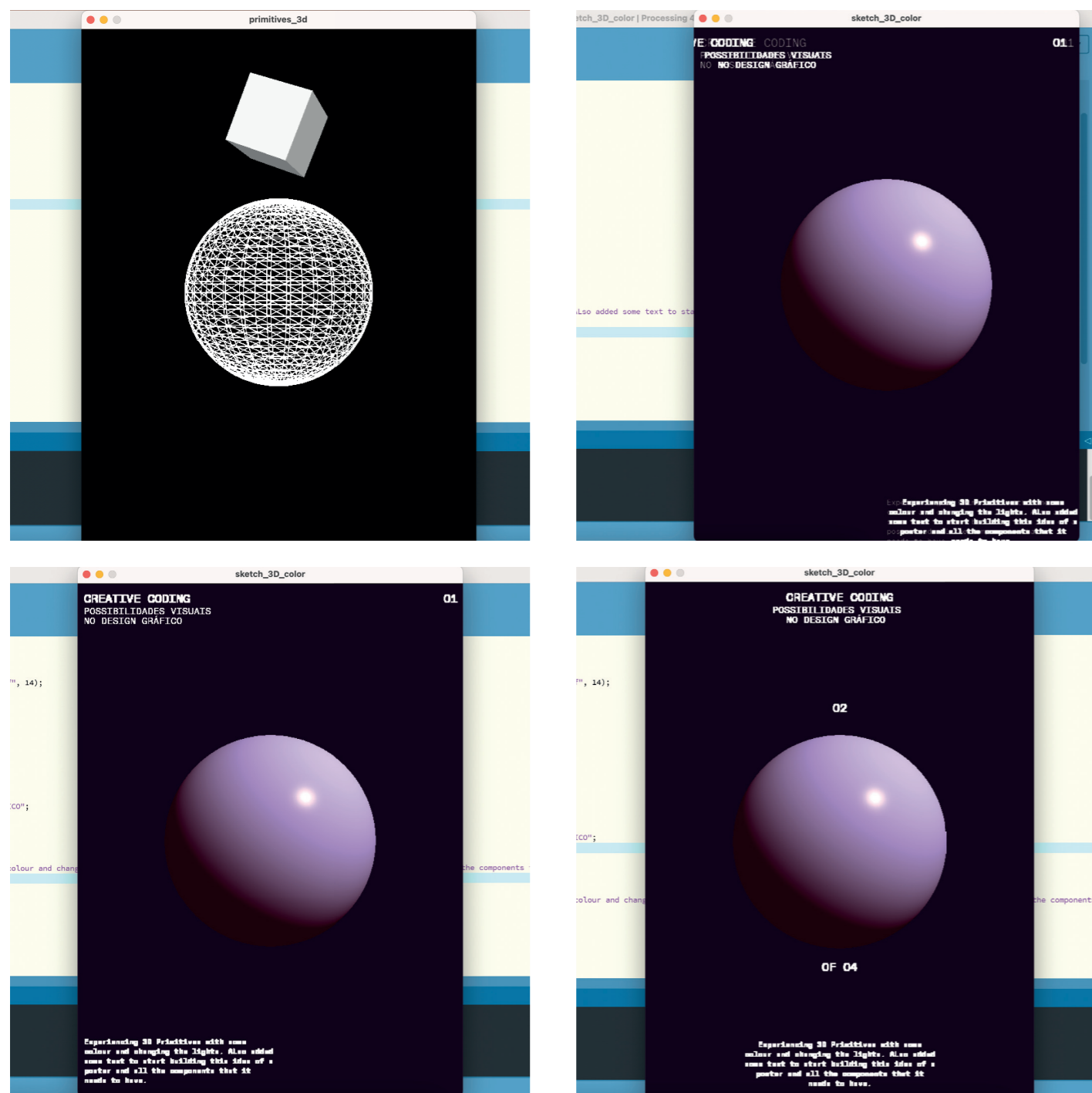


Fig 34 - Experiência Visual das formas 3D

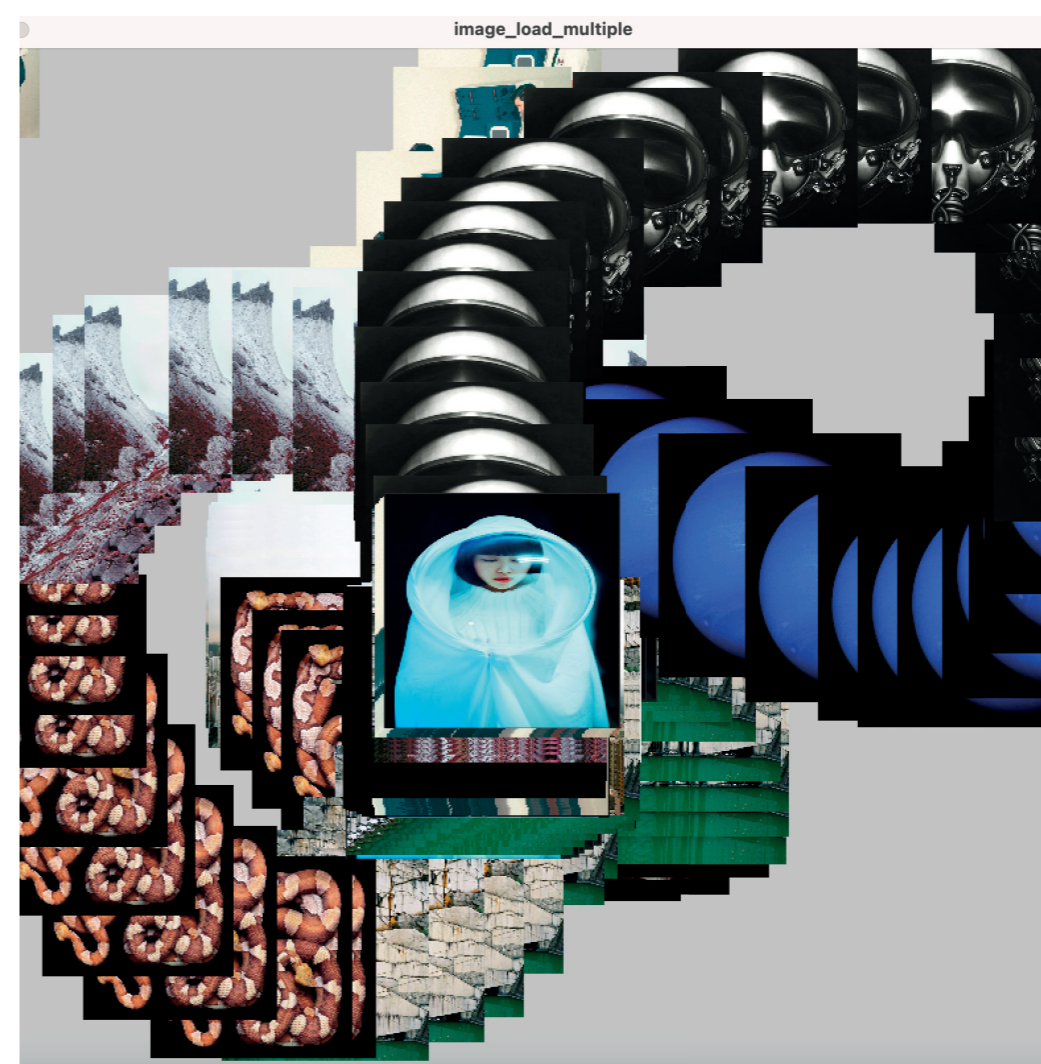


Fig 35 - Experiência Visual de imagens random

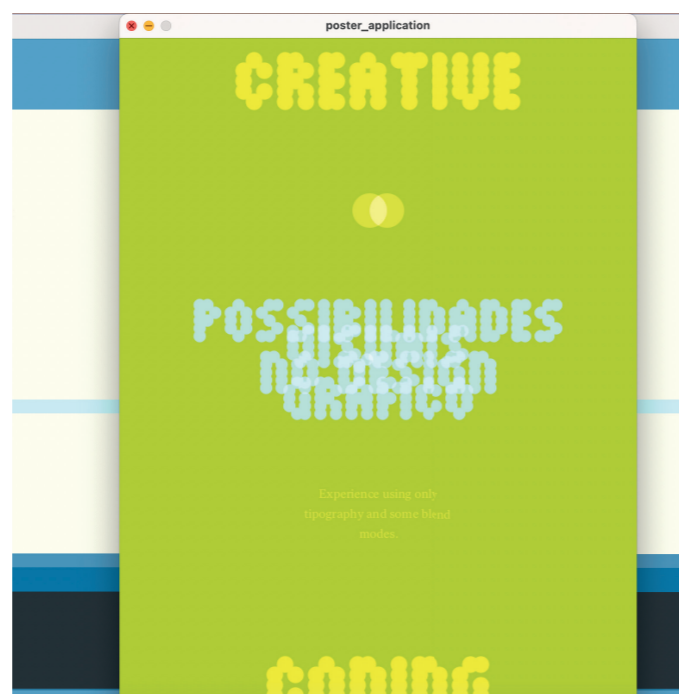
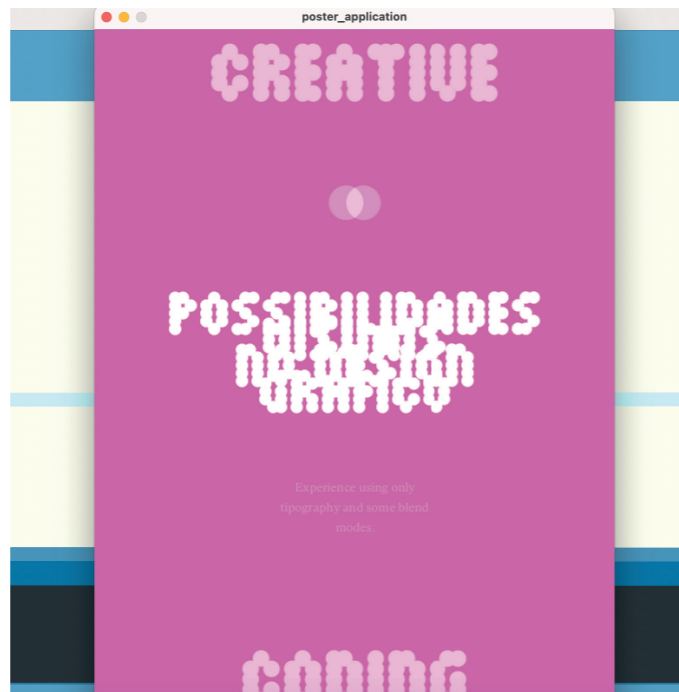
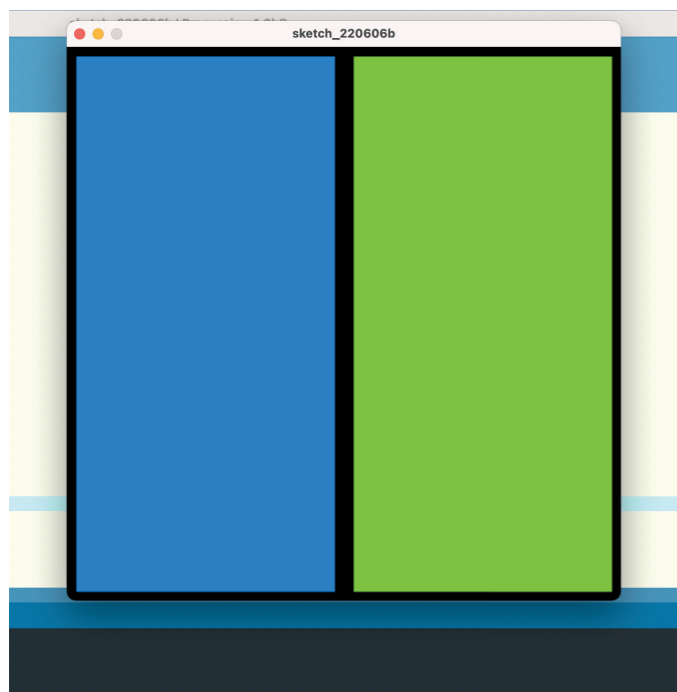


Fig 36 - Experiência Visual das cores random

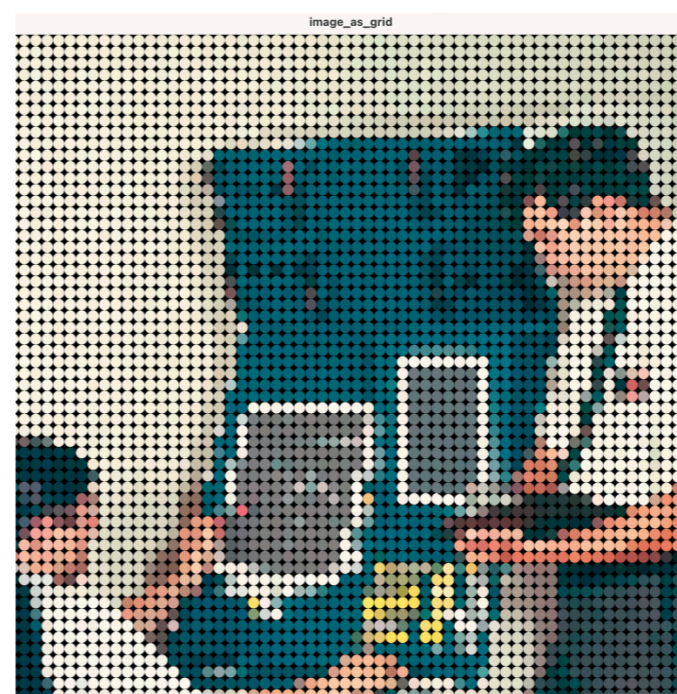
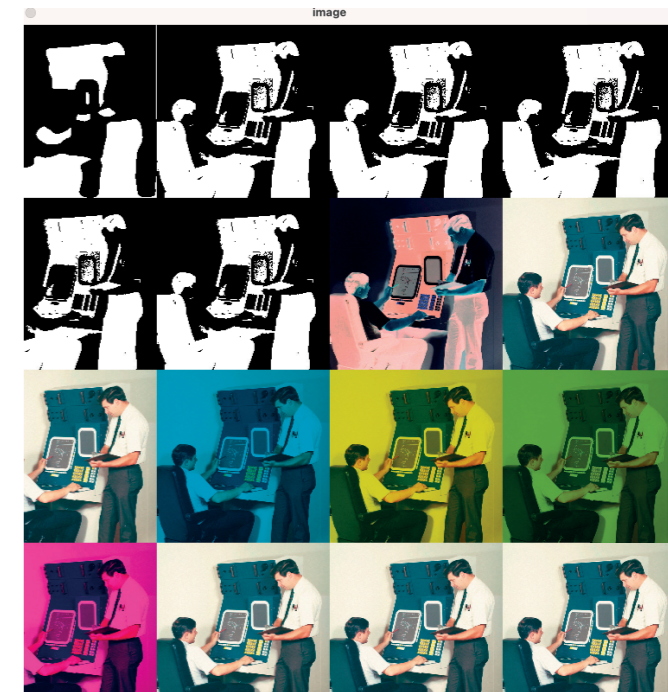


Fig 37 - Experiência Visual das imagens/pixeis



```

1 String s1 = "Nome";
2 String s2 = "Apelido";
3 String sc1 = s1 + s2;
4 int e = 50 * 5; // Sets 'e' to 250
5 int f = e * 5; // Sets 'f' to 1250
6
7 println(s1);
8 println(s2);
9 println(sc1);
10 println(e);
11 println(e + "x" + "5 =" + f);
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

```

Nome  
 Apelido  
 NomeApelido  
 250  
 250x5 =1250

Fig 38 - Experiência Visual dos cálculos (somadas, divisões)



Fig 39 - Experiência Visual do texto/tipografia

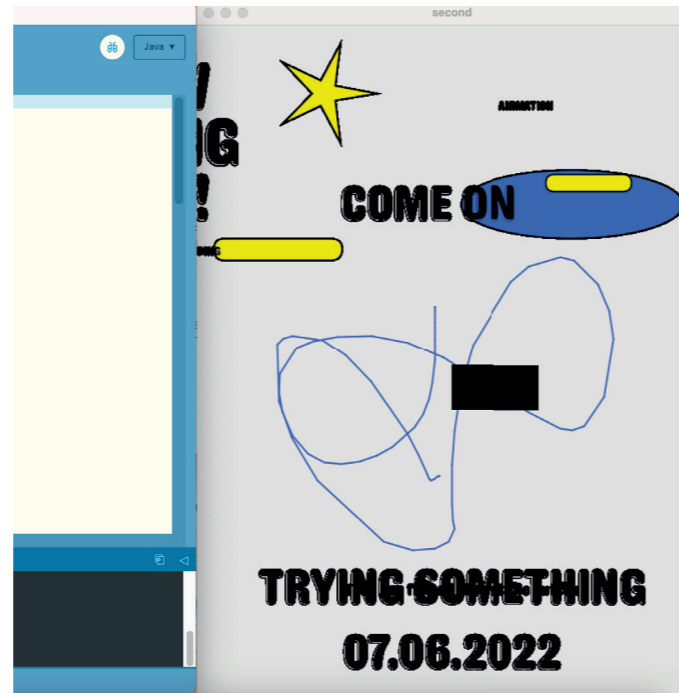
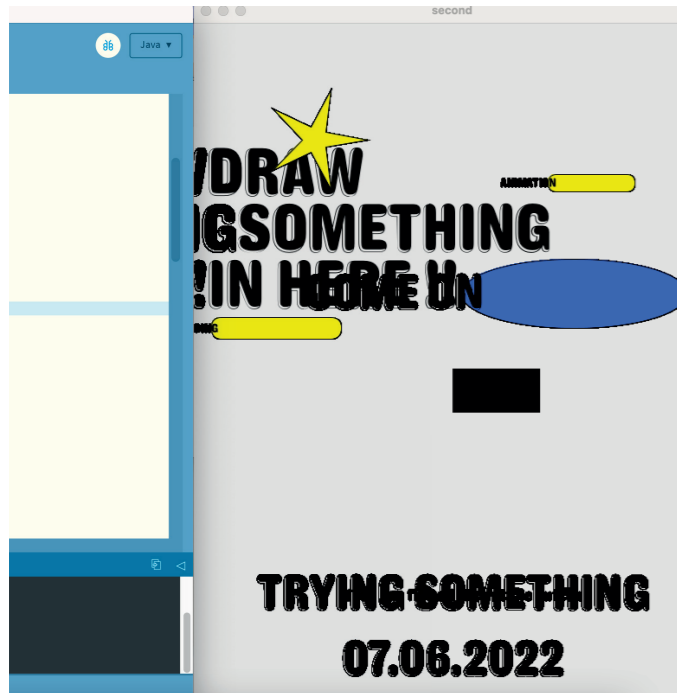


Fig 40 - Experiência Visual das animações

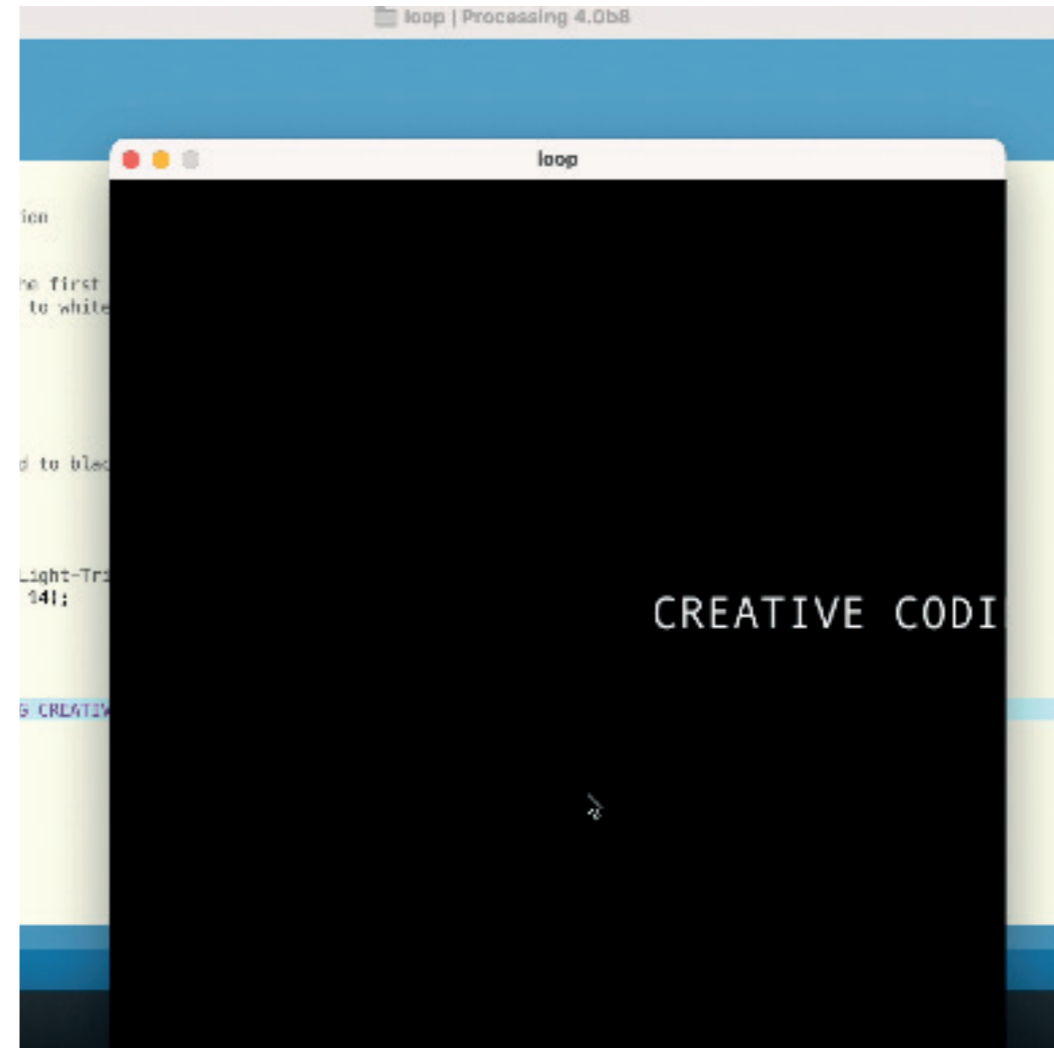


Fig 41 - Experiência Visual das repetições (for, loop, while)



Fig 42 - Experiência Visual das condições (if & else)

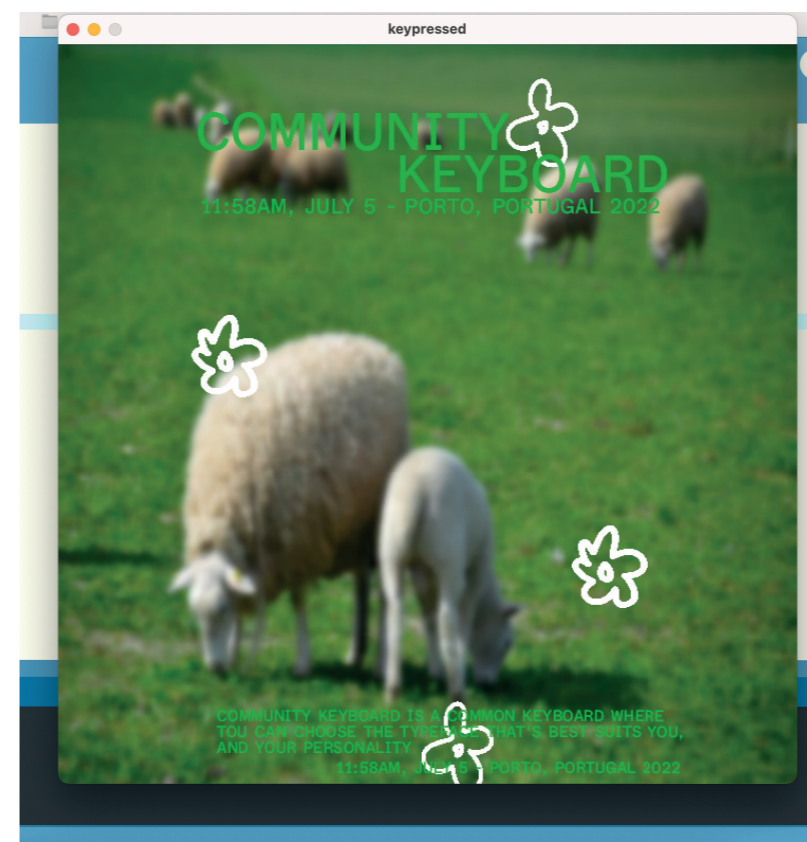


Fig 43 - Experiência Visual das interatividade

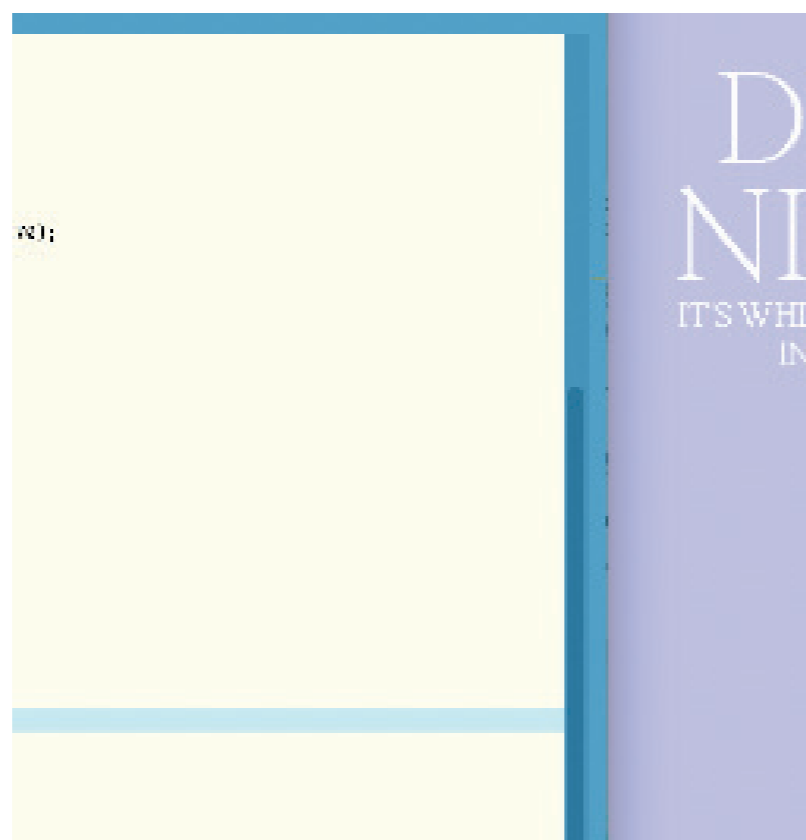


Fig 44 - Experiência Visual de motion (velocidade, força, tempo)



Fig 45 - Experiência Visual de transformações

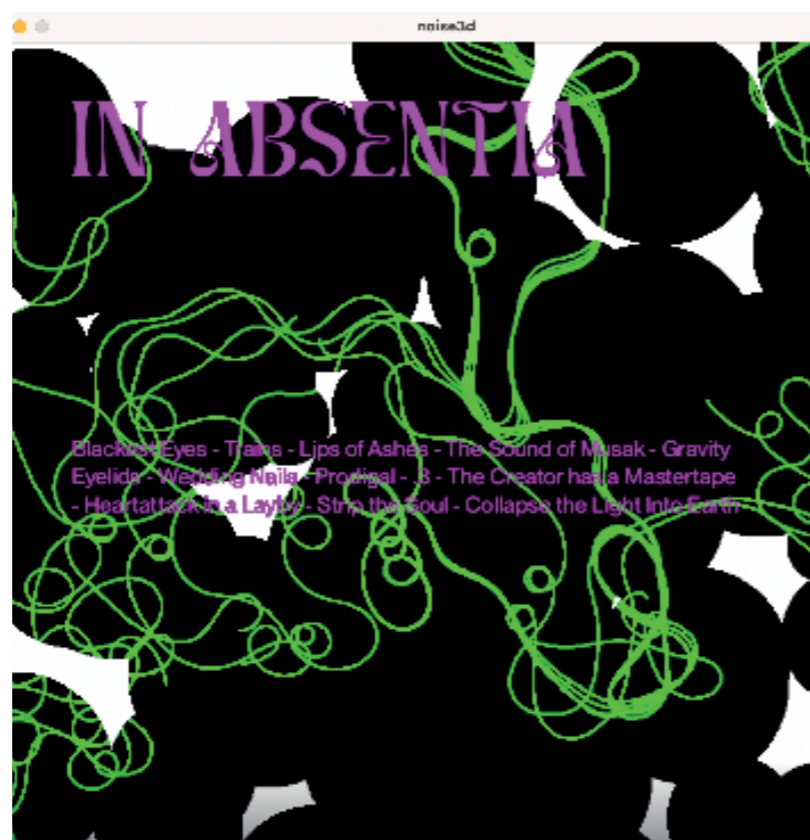
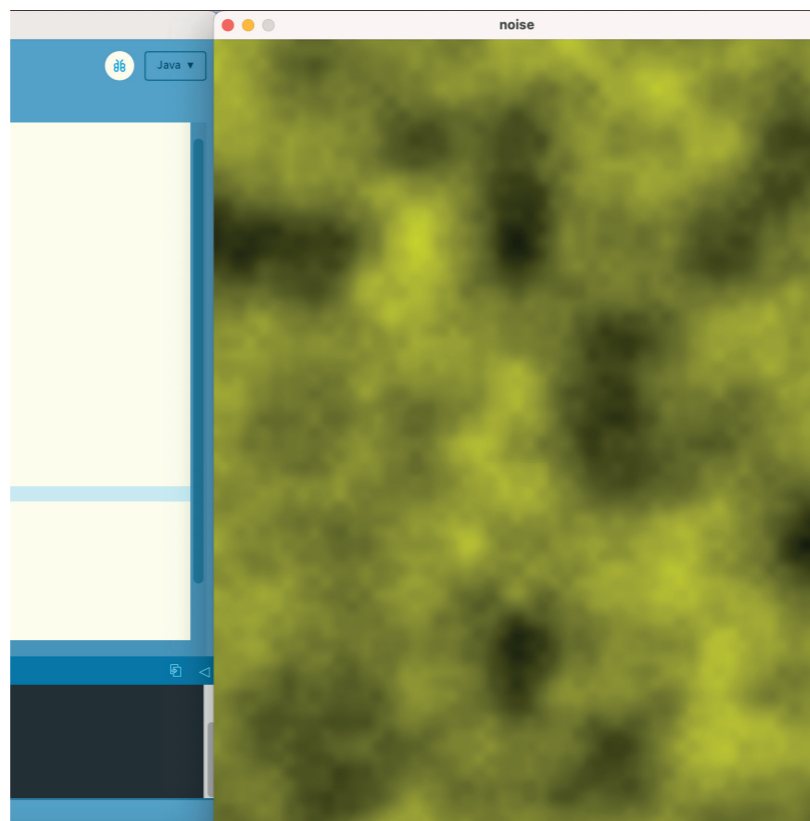


Fig 46 - Experiência Visual de noise

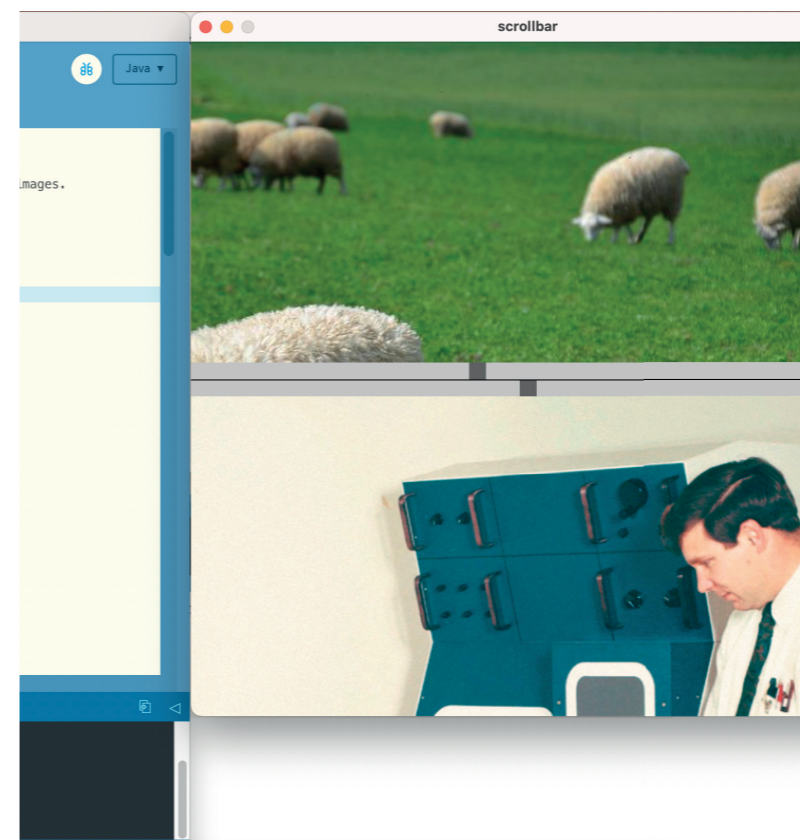


Fig 47 - Experiência Visual de parametrização (botões)

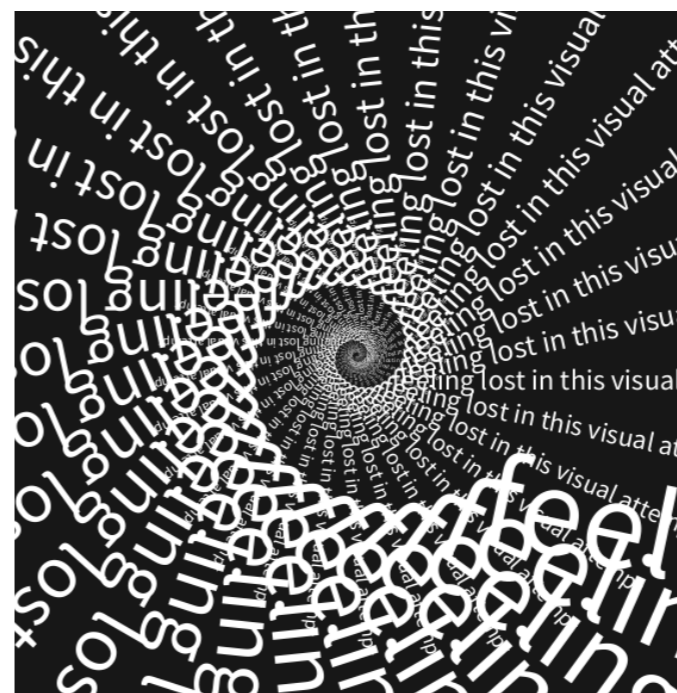
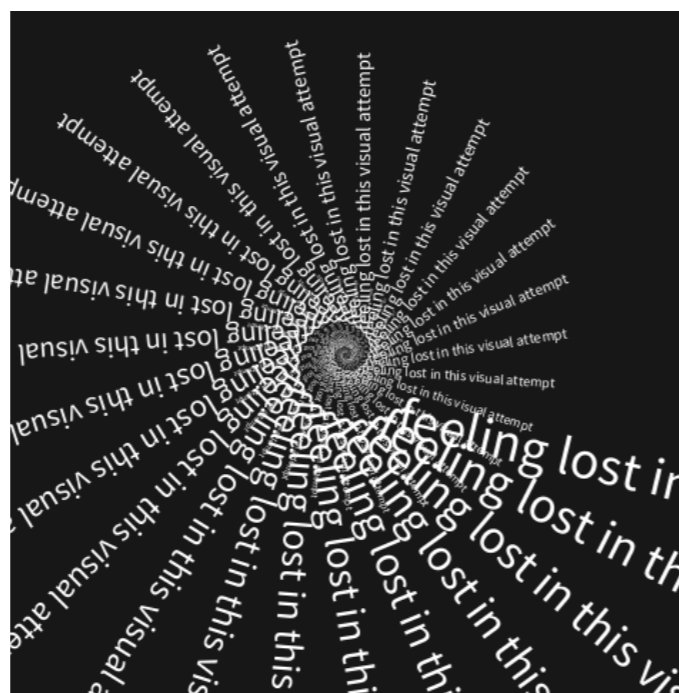
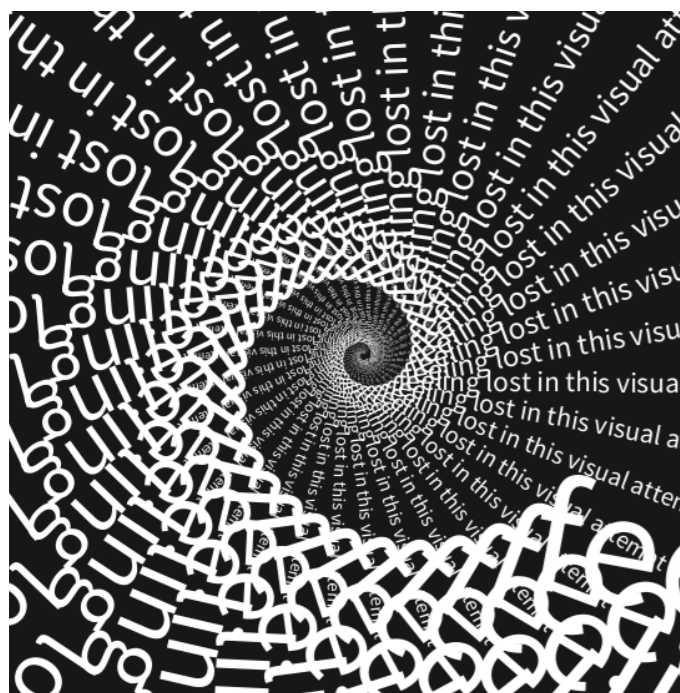
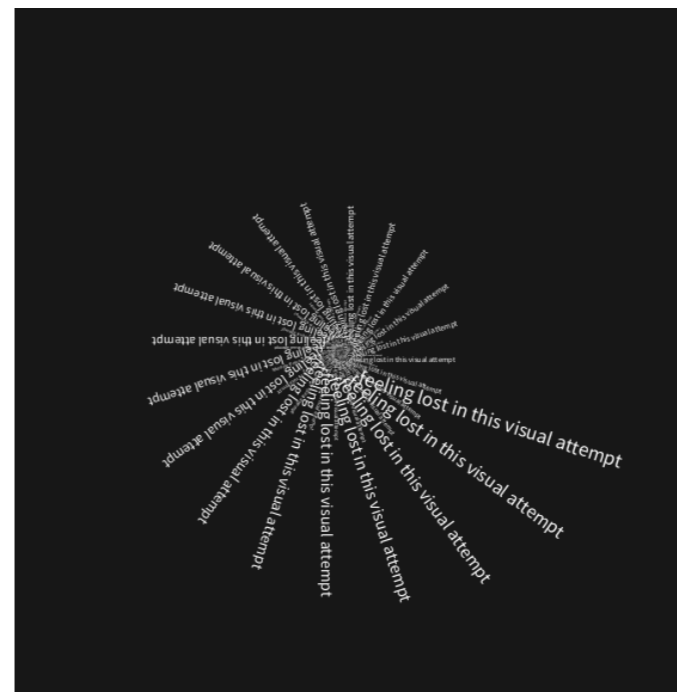
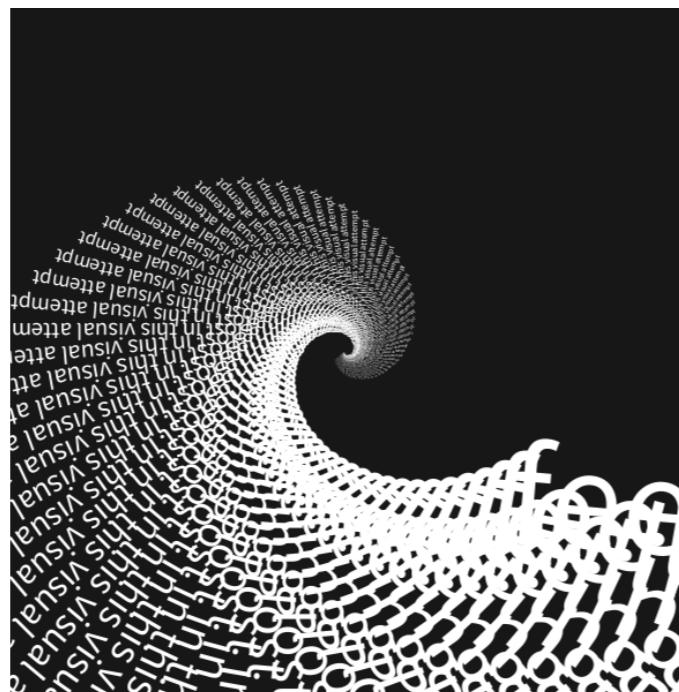


Fig 48 - Experiência Visual de transformações (rotação e escala)

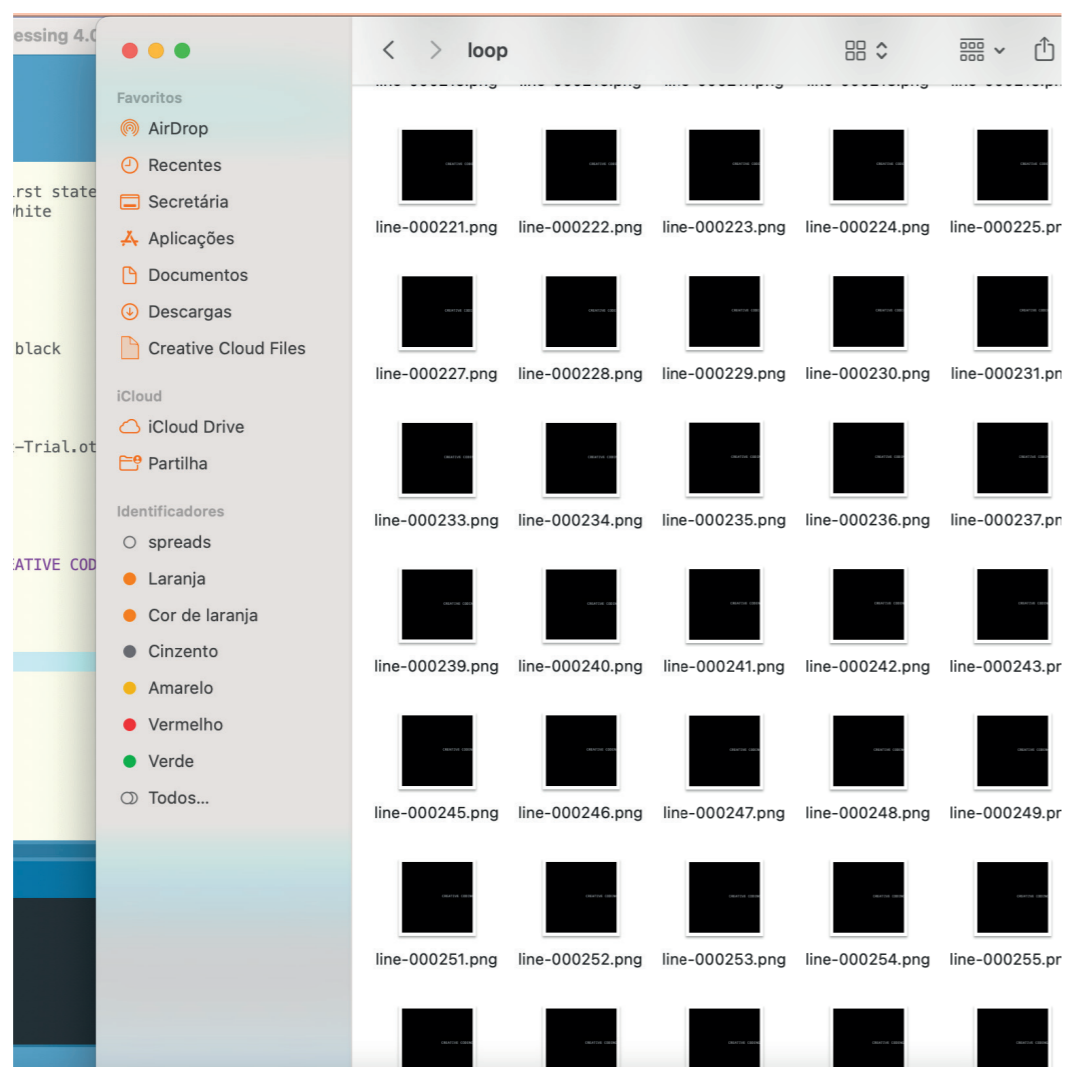


Fig 49 - Experiência Visual de exportação



### (4.3.3) Prototipagem

Após a fase de desenvolvimento das componentes, fez-se uma listagem das funcionalidade que se ambicionava que a ferramenta contivesse. Depois de concluída, seguiu-se para a fase de desenvolvimento da ferramenta, um desenho e prototipagem em wireframe, primeiramente em papel, recorrendo à técnica de sketching, e posteriormente utilizando o **Figma**. Separou-se a ferramenta em algumas secções: imagem, texto, objetos e animações. Estas quatro fases constituem as layers que se pretendia que existessem no projecto, bem como a abordagem que se pretendia na constução de um ensaio visual na ferramenta.

**Figma** - O Figma é uma plataforma colaborativa com standards de indústria focada no Desenho e prototipagem de Interfaces, que permite simular pequenas animações, e interações. Foi lançada em 2016 por Dylan Field e Evan Wallace com o objectivo de criar uma ferramenta gratuita que permitisse a colaboração entre pessoas e equipas que criasse um produto para diversas plataformas e formatos. Em Setembro de 2022 foi comprada pela Adobe. Até ao momento e dada a brevidade do sucedido, mantém-se uma plataforma gratuita e sem quaisquer alterações.

**Parametric Buttons: Control P5**

Como é que este botão/botões pode permitir realizar mudanças no design?

- Aumentar/diminuir tamanho
- Deslocar x, y, z
- Acrescentar/retirar elementos (download de fotografias, escrever texto)
- Escolher Tipografia
- Animar elementos speed, amplitude
- Effects pré-definidos (?) (wave, bubbly)
- Exportar imagens/videos
- Criar formas 2D e 3D (triangle, circle, square)
- Pentool, to draw your shape
- Cor
- Repetir informação no Eixo x, y e z
- Escolher resolução de exportação

Fig 50 - Listagem de componentes que se pretendia aplicar

| Imagem                   | Texto                            | Formas/Vetores   | Animar             | Exportar       |
|--------------------------|----------------------------------|--|--------------------|----------------|
| inserir   eliminar       | inserir   eliminar               | Inserir 2D ou 3D, retângulo, triângulos, círculo, quadrado | inserir   eliminar | Jpg, png, tiff |
| aumentar/diminuir        | aumentar/diminuir width e height | eliminar   | animação 1         | Gif            |
| Deslocar no eixo x,y e z | Deslocar no eixo x,y e z         | Desenhar   | animação 2         | Video, mp4     |
| Repetir no eixo x, y e z | Repetir no eixo x, y e z         | Deslocar no eixo x,y e z                                   | animação 3         |                |
| Filtros nas imagens      | Escolher Tipografia              | Repetir no eixo x, y e z                                   | animação 4         |                |
|                          | Alterar cor                      | Cor  |                    |                |
|                          | Line height                      | Espelhar uma imagem  |                    |                |
|                          | Text Alligment                   | Alterar tamanho  |                    |                |
|                          | Background Color                 |  |                    |                |

Fig 51 - Segmentação e Categorização dos componentes nas devidas secções

Definir-se-iam as imagem, depois o texto, os objectos e, por fim, animar-se-iam os mesmos. Os objectos, texto e imagem também representam os três níveis de conteúdos que podemos ter na criação de um ensaio.

Inicialmente o protótipo encontrava-se muito aberto e livre. O utilizador tinha liberdade para alterar todos os parâmetros, o que em nada distinguia esta ferramenta de outras já utilizadas e do conhecimento de utilizadores dentro da área do design gráfico. Era necessário promover a generatividade, mas que esta fosse previamente controlada, limitando as escolhas do utilizador. Nesta primeira versão o utilizador, por exemplo, conseguia adicionar mais do que uma imagem. Relativamente a cada imagem conseguiria definir a sua posição no eixo X, Y, bem como a sua altura e largura. Conseguiria repeti-la no eixo X, Y e Z, bem como adicionar-lhe alguns filtros. Algumas das opções para editar esta imagem eram colocar um tint, inverter a cor, blur, posterizar, bem como colocar a imagem numa escala de cinzas. Para além disso, haveria também os botões de exportação de imagem, vídeo e gif. O mesmo acontecia com o Texto. Aqui, tal como nas imagens, poderia colocar mais do que um texto, editar o seu tamanho, o seu line height, o espaçamento da própria mancha tipográfica em altura e largura, a sua posição no eixo X e Y, repetir o texto no eixo X, Y e Z, definir a sua cor e a cor de fundo de todo o **canvas**. Haveria também uma secção de objetos onde poderíamos desenhar formas já pré-definidas como um círculo, um quadrado/retângulo, um triângulo, uma esfera, um cubo e desenhar vetorialmente alguma forma desejada. Aqui, tal como nas outras secções, poderíamos editar a altura e largura do objeto, definir a sua posição no eixo X e Y, repetir o objeto no eixo X, Y e Z, definir a sua cor de fundo e de contorno, bem como aplicar uma imagem sobre o mesmo.

Por fim, definiu-se uma secção de animações, onde haveria 4 animações pré-definidas, na qual apenas poderíamos alterar o tempo e a velocidade da mesma. Depois de fechados os wireframes e de forma a começar a perceber as dificuldades do próprio projeto, as funcionalidades começaram a ser implementadas tendo por base os wireframes criados, num minimal viable product (**MVP**).

Nesta MVP, foi de imediato detectado que o projeto seria mais simples e eficaz de desenvolver usando p5.js em vez deProcessing. Apesar de o Processing possuir a biblioteca Control P5 que permite a adição de elementos para parametrização através de um interface gráfico como botões, campos de texto, sliders, entre outros, onde muitas destas funções já estão pré-estabelecidas, a mesma não permite uma estilização tão extensiva dos elementos, que enquanto designer, e tendo esta proposta uma base e conexão direta com o design gráfico, se torna importante.

Transitou-se para o p5.js, cuja linguagem é semelhante e permite uma estilização tota, dada a sua ligação mais direta com linguagens como Html e Css. Esta linguagem permitiu também repensar um pouco a estrutura e divisão do projeto e todas as partes que o constituem, bem como também a de um pensamento para um desenvolvimento de uma ferramenta com um design responsivo, isto é, que funcione não só em computadores pessoais mas também para outros tipos de dispositivos como os telemóveis.

Algumas alterações imediatas que se fizeram foi no posicionamento dos elementos. Depois duma pesquisa de ferramentas semelhantes já existentes, alterou-se a disposição do menu de ferramentas para o lado direito e o canvas para o esquerdo.

**Canvas** - Canvas significa uma tela, um quadro no qual descrevemos e dispomos diferentes elementos.

**MVP** - Minimal Viable Product, ou produto viável mínimo é a versão mais simples de um produto que pode ser criada e lançada com uma quantidade mínima de esforço e desenvolvimento. Esta criação mais rápida do produto permite receber feedback dos utilizadores de forma rápida, de quando é posto em prática em testes de usabilidade.

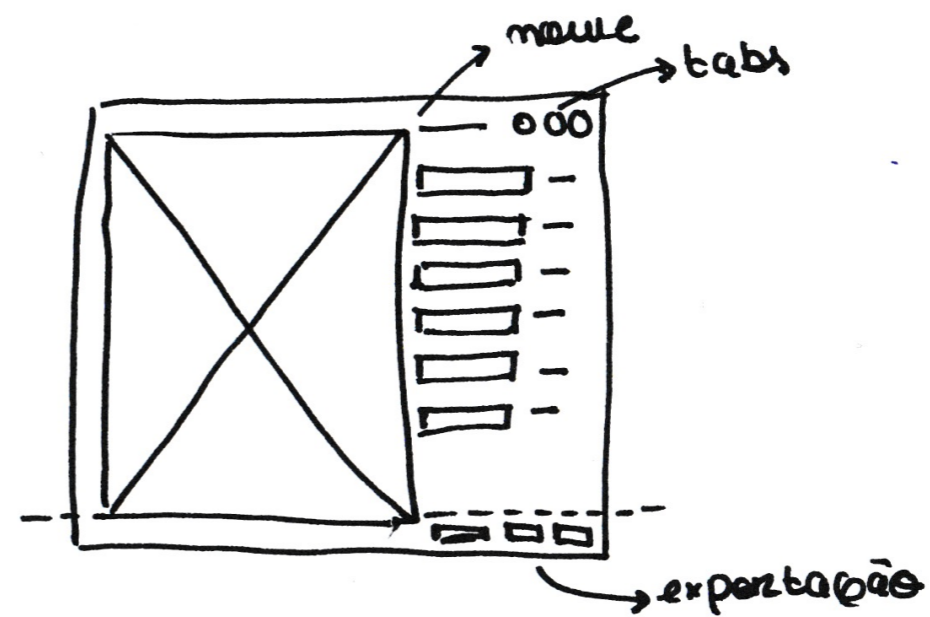
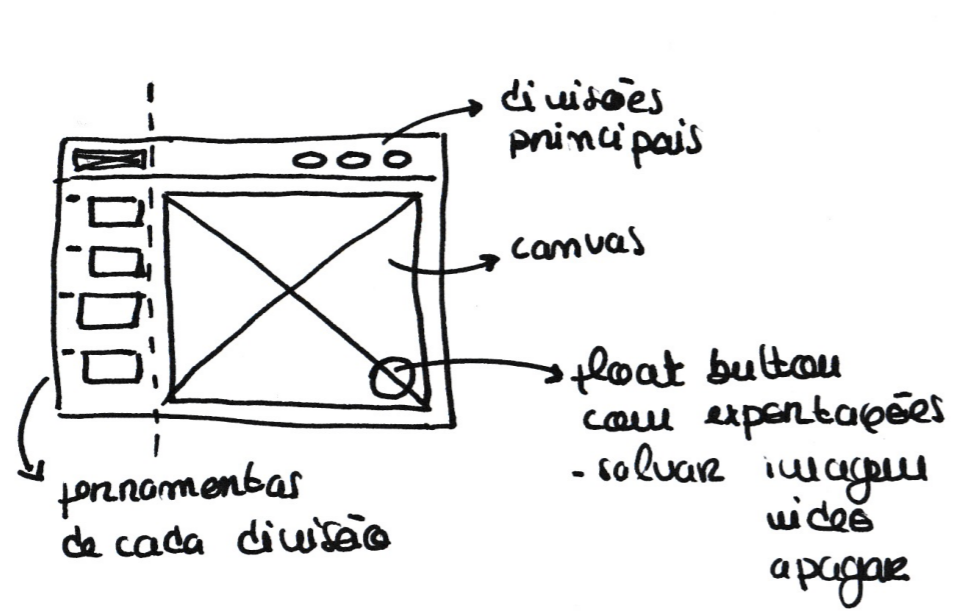
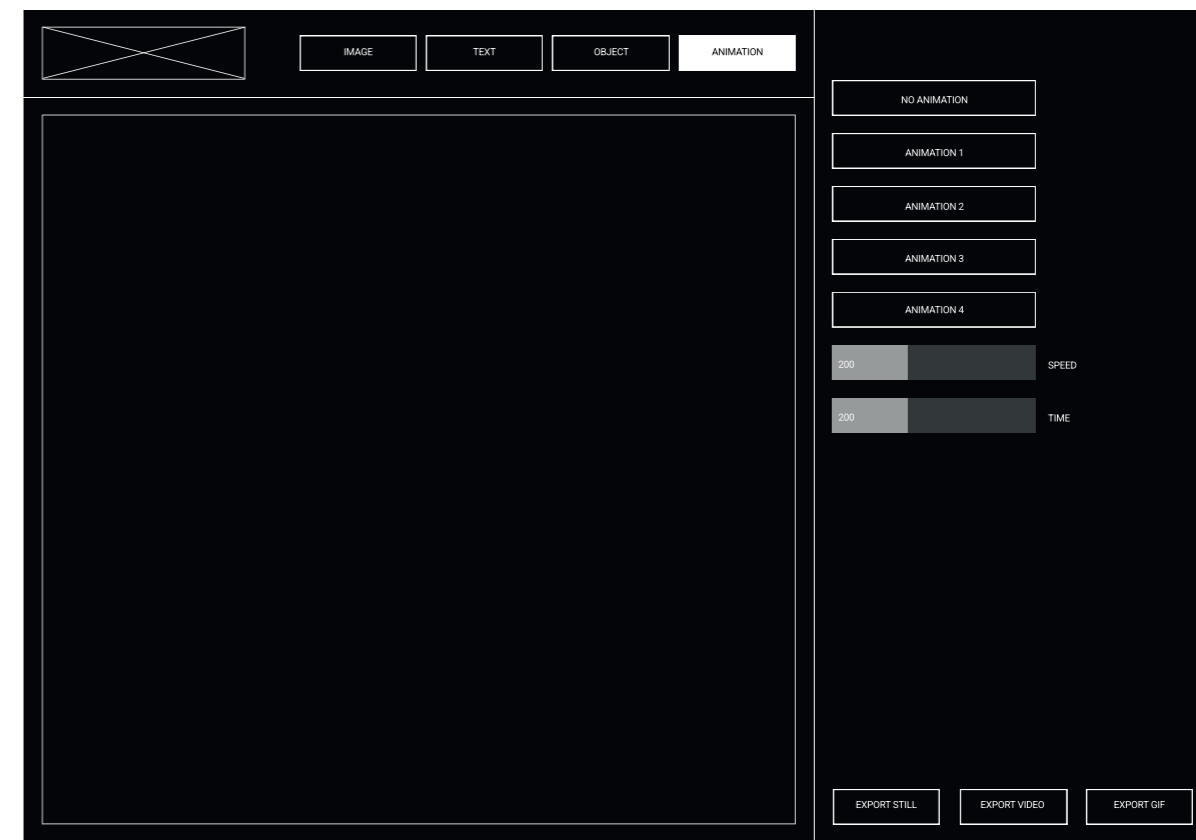
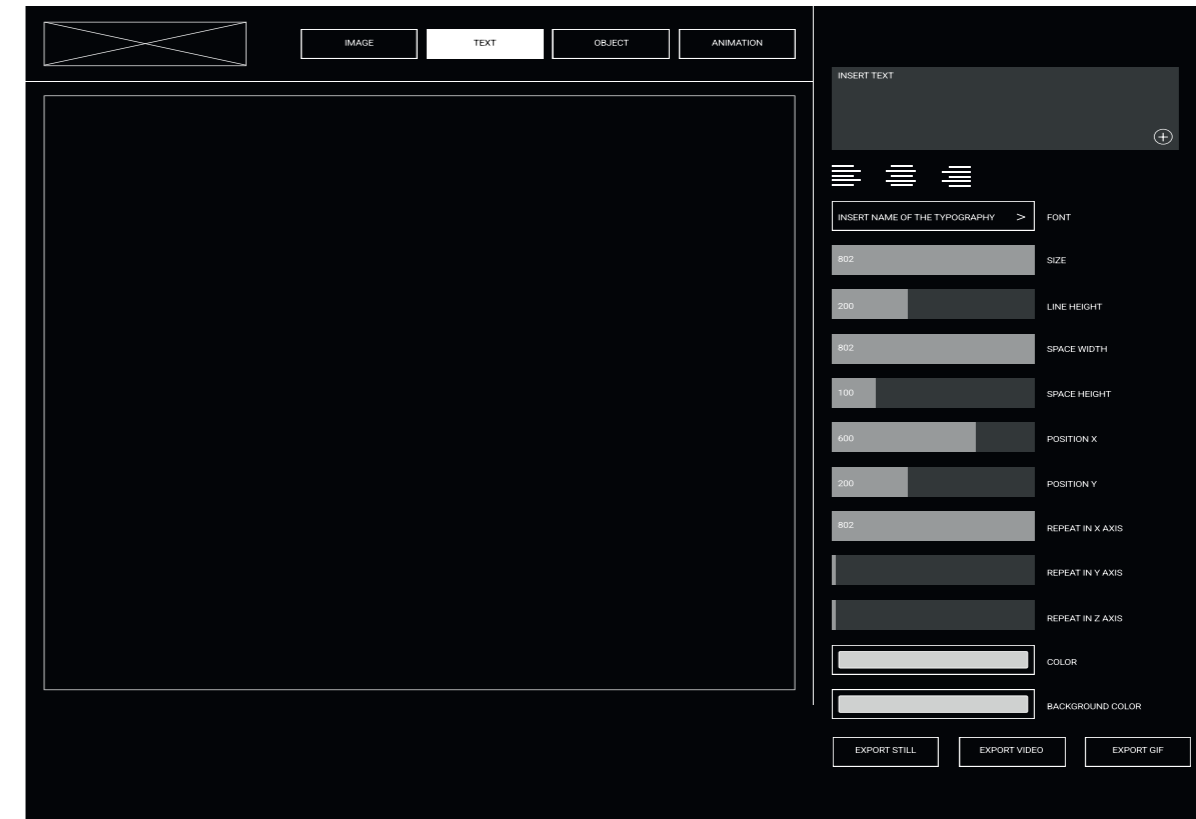


Fig 52 - Wireframes em papel (sketching)



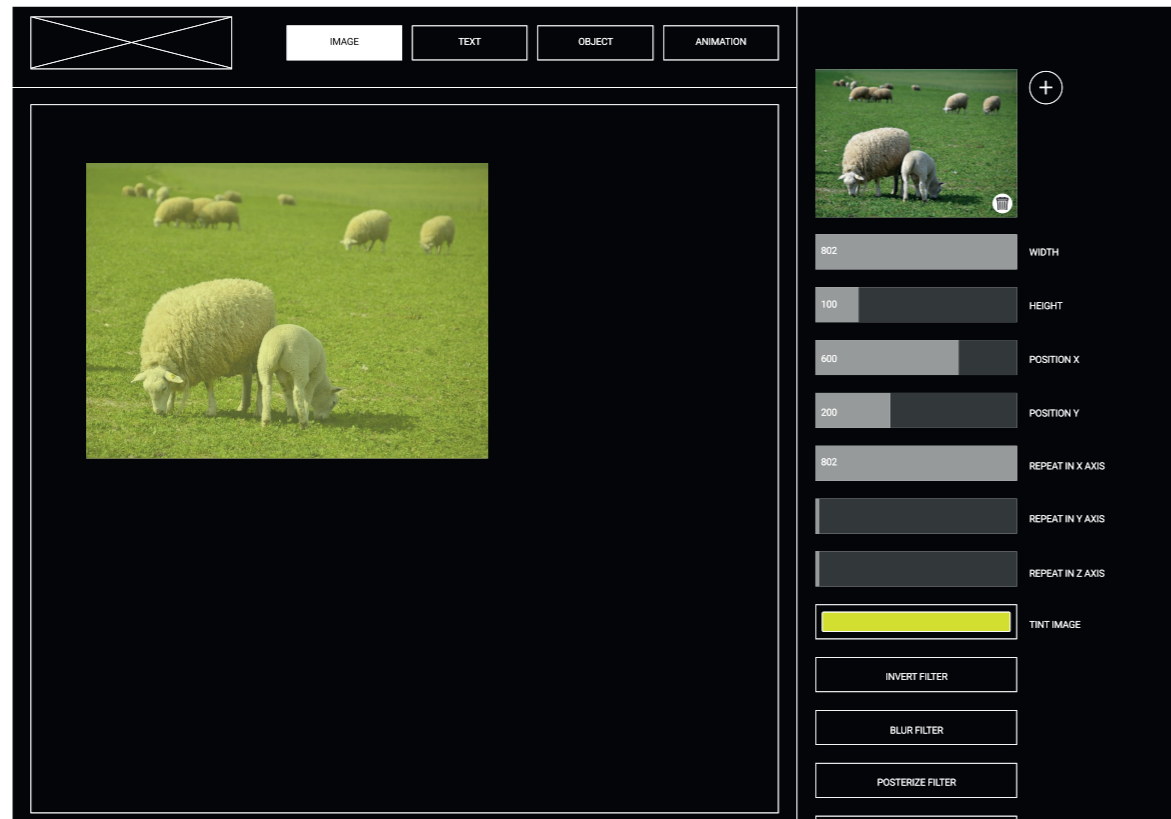


Fig 53 - Primeiros wireframes da ferramenta desenhados em Figma

Também ficou estabelecido que a ferramenta poderia desenhar 2 tipos de soluções 600 x 600 (px, pixéis), que permite um tamanho quadrangular semelhante ao desenho de conteúdo para social media, e 800 x 600 (px, pixéis), um tamanho mais vertical, que se assemelha a um desenho de cartaz. (Estes dois tamanhos também já estavam a ser utilizados nos ensaios visuais dos componentes.)

O posicionamento dos objetos, textos e imagens, passou a ser controlado livremente pelo utilizador, com um espectro onde o mesmo poderia colocar onde desejasse, sendo que tanto o valor de x e y alterariam em conjunto e não em separado como nos protótipos anteriores.

O tamanho dos objetos, texto e imagens passaria a funcionar com 3 opções: SMALL, MEDIUM e BIG, com valores estipulados por forma ao utilizador deixar de ter tanto controlo e poder aplicar-se aqui um dos princípios da parametrização.

Aplicou-se também um botão em todas as páginas de RANDOMIZE ALL, que iria colocar valores aleatórios (random) em todos os campos que fossem alteráveis. Inseriu-se nestas fase do protótipo também uma tab com as ferramentas de exportação em formato de imagem, vídeo (que já existia nas versões anteriores), bem como um botão de “clear” para apagar todo o desenho feito até então. Se nestas primeiras versões do protótipo havia uma liberdade para se colocar os elementos exatamente onde o utilizador desejava, nas versão seguinte limitaram-se os posicionamentos para o centro, topo e baixo, esquerda e direita. Mesmo estes valores não necessitariam de ser lineares, iriam estar inseridos num espectro de parâmetros selecionados, estando de qualquer forma sujeitos a uma aleatoriedade de valores (random values). O mesmo aconteceu com o tamanho das imagens, do texto e dos objetos. Definiram-se níveis de inserção dos mesmos, pequeno, médio e grande, full screen e full screen with margins (estes últimos dois casos aplicam-se apenas para as imagens), de forma a promover a aleatoriedade e a parametrização, duas das componentes que mais se pretendia explorar com este projeto.

Na secção dos objectos, houve uma redução das opções, passando o utilizador a poder desenhar formas primitivas como um círculo, um quadrado, uma esfera e um cubo, devido à complexidade das outras formas pensadas inicialmente. Quanto à secção do texto foi necessário definir tipologias do mesmo, Headline (título), parágrafo e caption (legendas/notas de rodapé), de forma a reduzir a liberdade de escolha do utilizado.

Depois de afinados todos estes pormenores, com ainda cinco versões realizadas do protótipo, era necessário definir estilos de texto e cor e a identidade gráfica do projeto, que só veio a revelar-se numa fase posterior. Todas as páginas foram desenhadas e pensadas com a exceção das animações devido à limitação do Figma. Na parte da animação o objetivo era enaltecer primeiramente a generatividade e a parametrização. Apesar destes dois termos parecerem em tudo opostos para o utilizador, o que transitaria seria a generatividade mas assente numa parametrização pré-selecionada.

Sempre num constante processo entre o código e o Figma, para começar a realizar essa mesma aplicação definiram-se no código algumas cores, ainda sem estarem assentes em nenhuma explicação ou com alguma base. Dado que a parte da programação estava a avançar, era necessário, tal como noutras fases, dar um passo atrás e perceber, mesmo que estaticamente, como é que seriam os possíveis resultados da ferramenta, bem como perceber melhor o que se queria realizar a nível de animações.

Foi nesta fase que se definiu a importância de desenvolver esta ferramenta a nível de mobile, pois é um dispositivo que cada vez mais utilizamos como elemento primário no nosso dia-a-dia e para alguma interação. Aqui também se percebeu a potencialidade, ainda que teórica, de mobile, pois mesmo sendo mais limitadora a nível de espaço, comparativamente a desktop, conseguimos aqui ter outro tipo de inputs e outputs por acrescentarmos uma via sensorial distinta de desktop.

Com a adição do desenvolvimento de mobile foi necessário fazer um reajuste de desktop para viver em harmonia com o de mobile, e vice-versa.

Tab - Uma tab é um elemento gráfico que tem como função fazer uma separação de componentes, normalmente uma separação de alguma opção de menu.

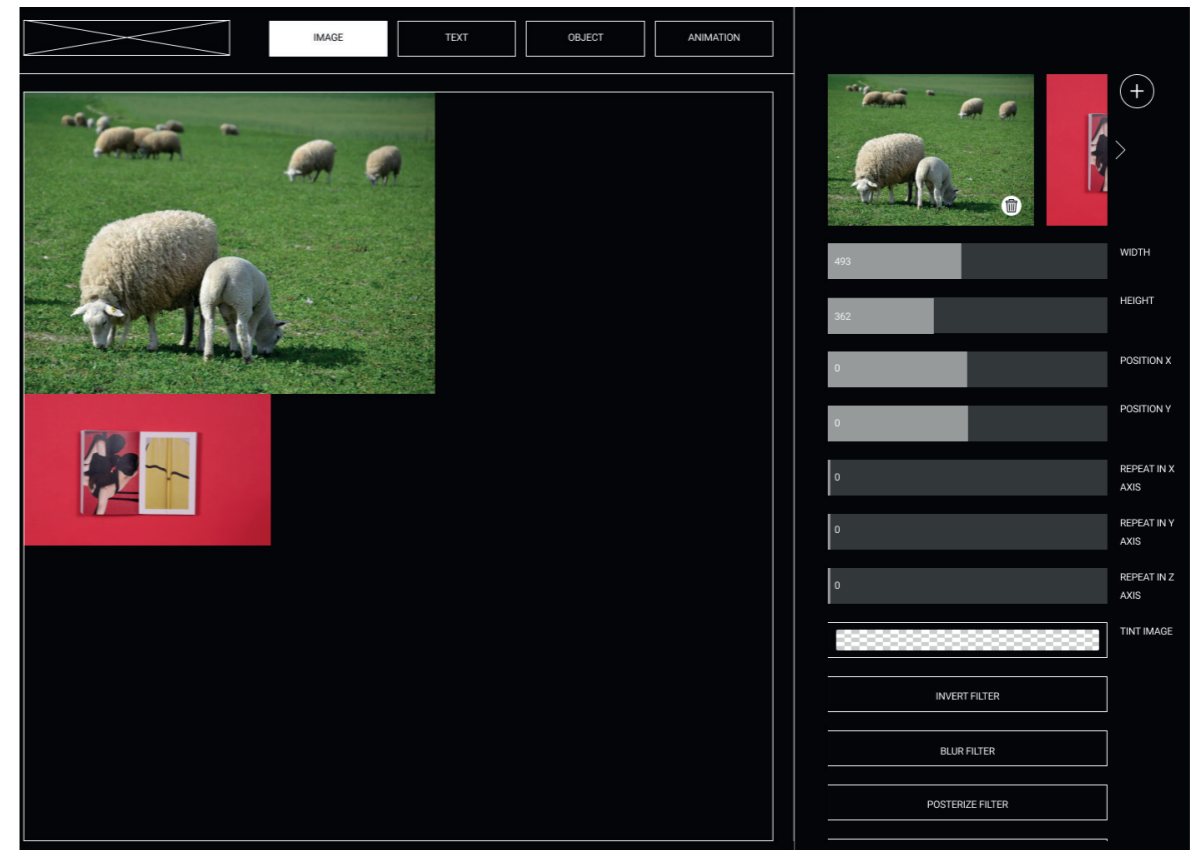
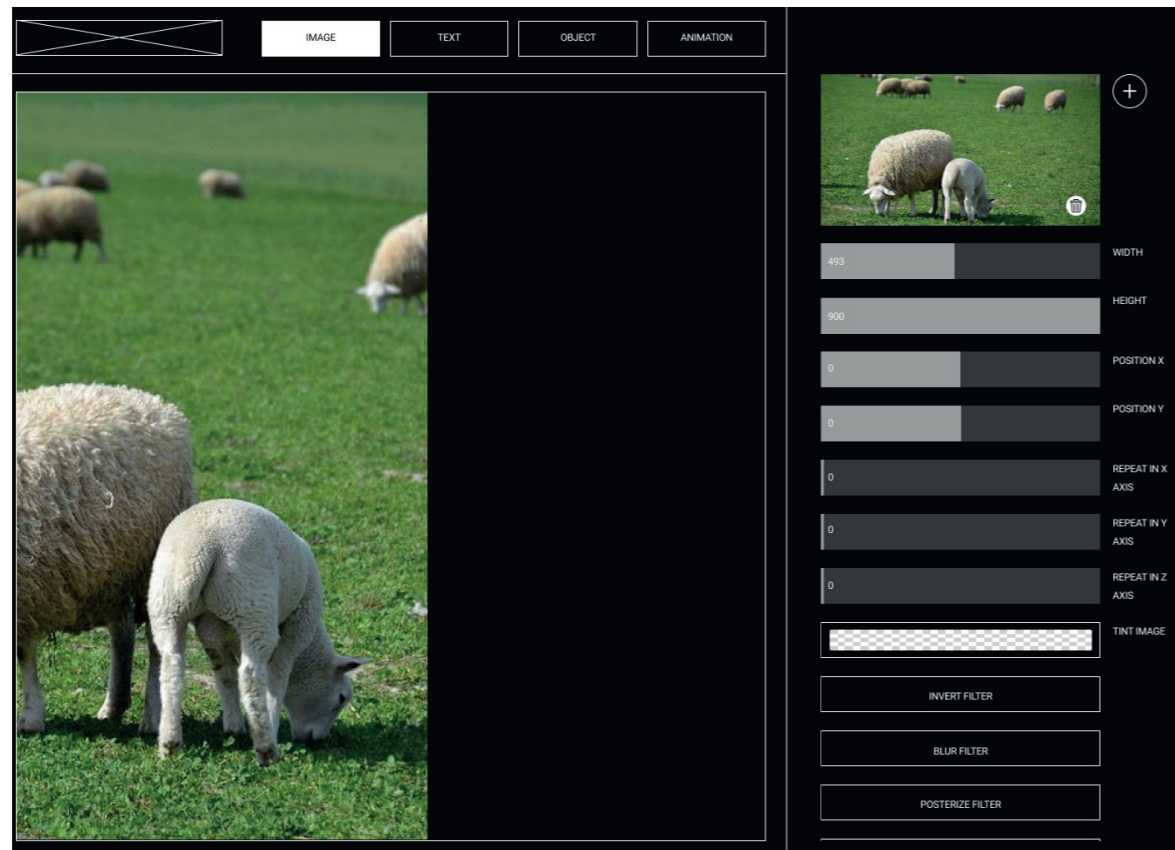


Fig 54 - Segunda versão dos wireframes da secção das imagens

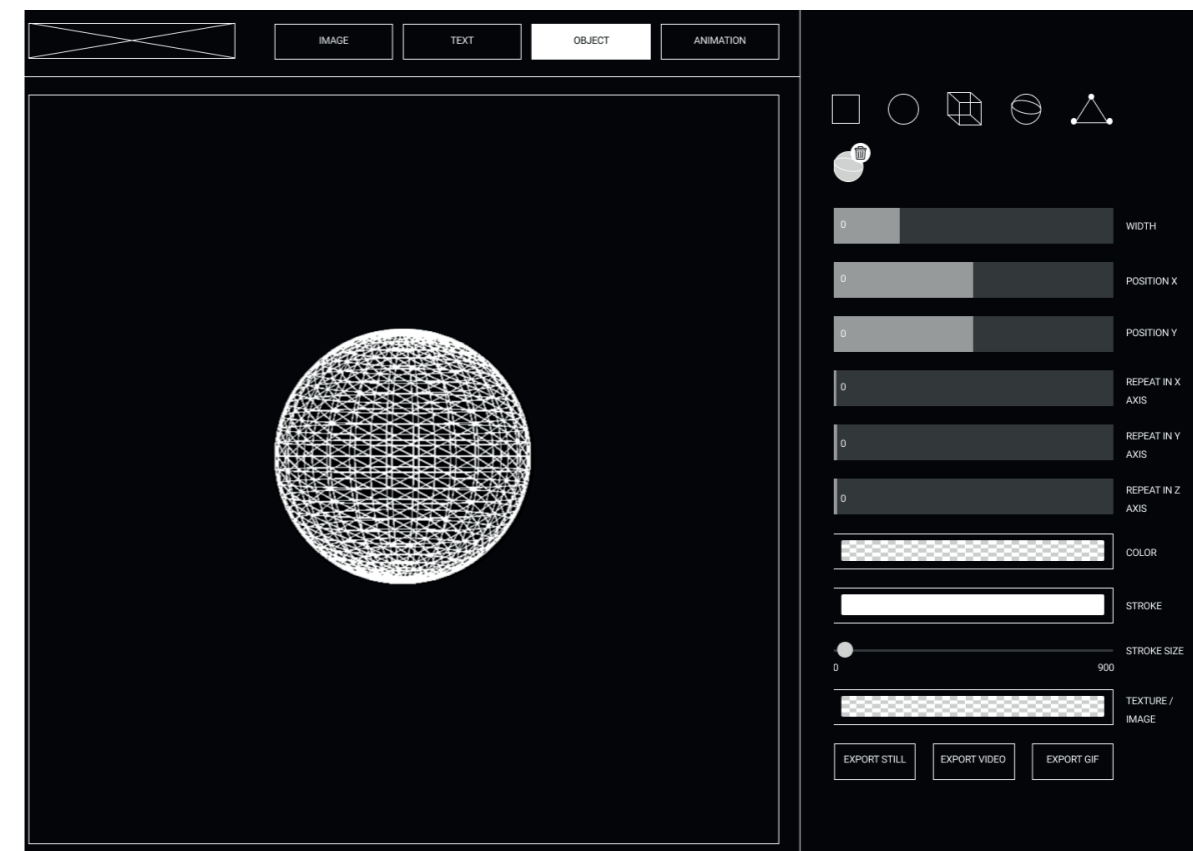
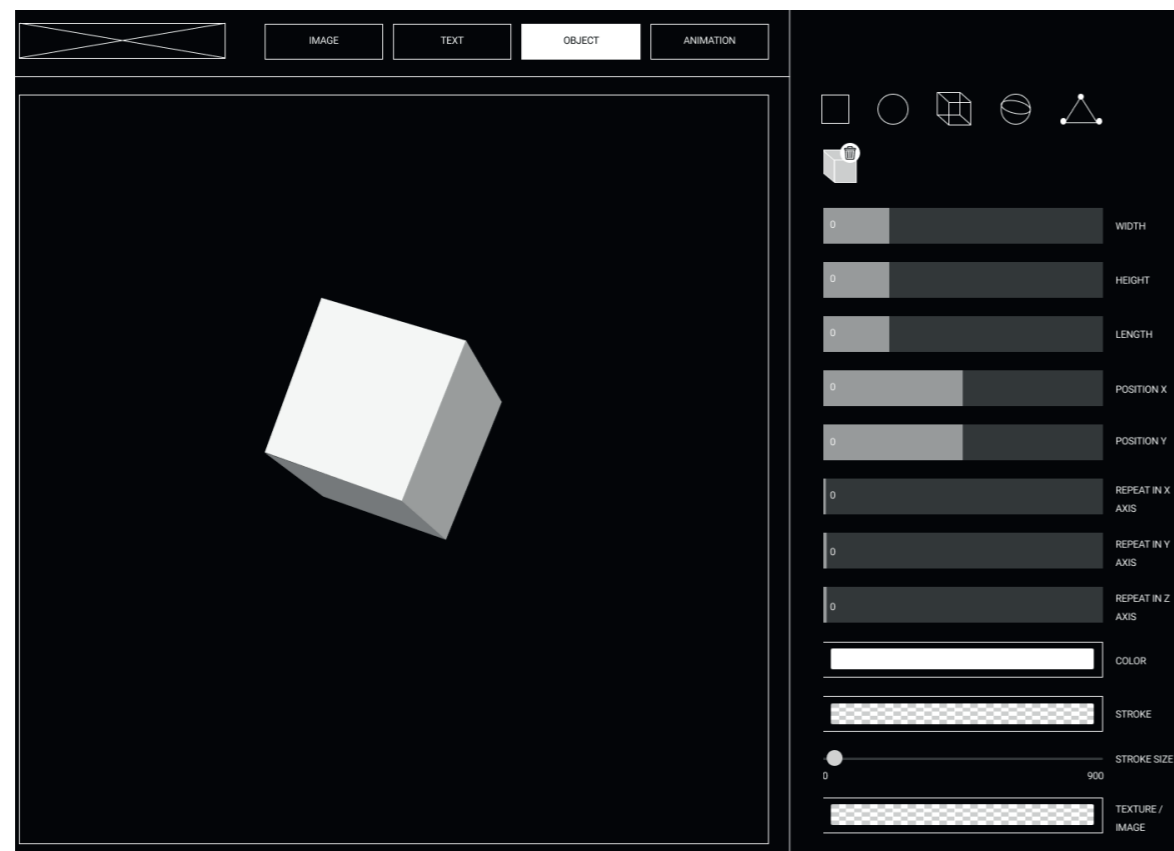
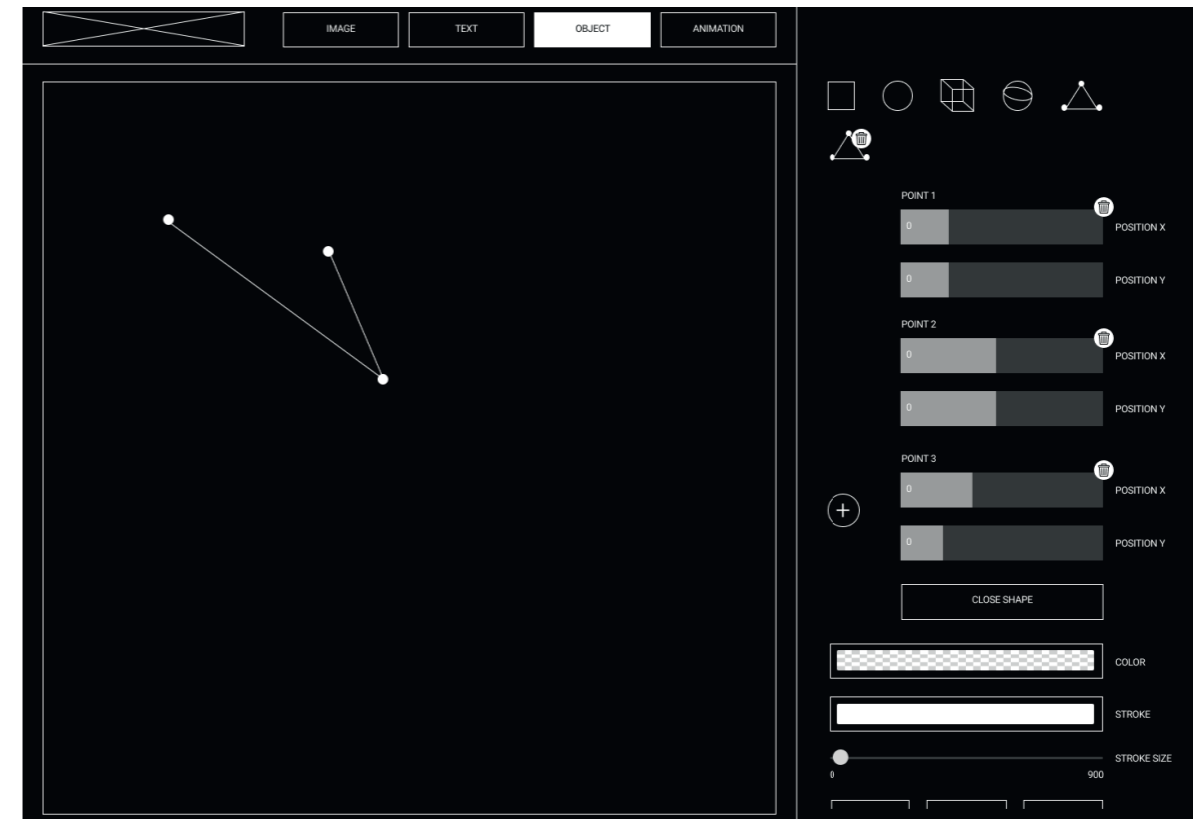


Fig 55- Segunda versão dos wireframes da secção dos objetos

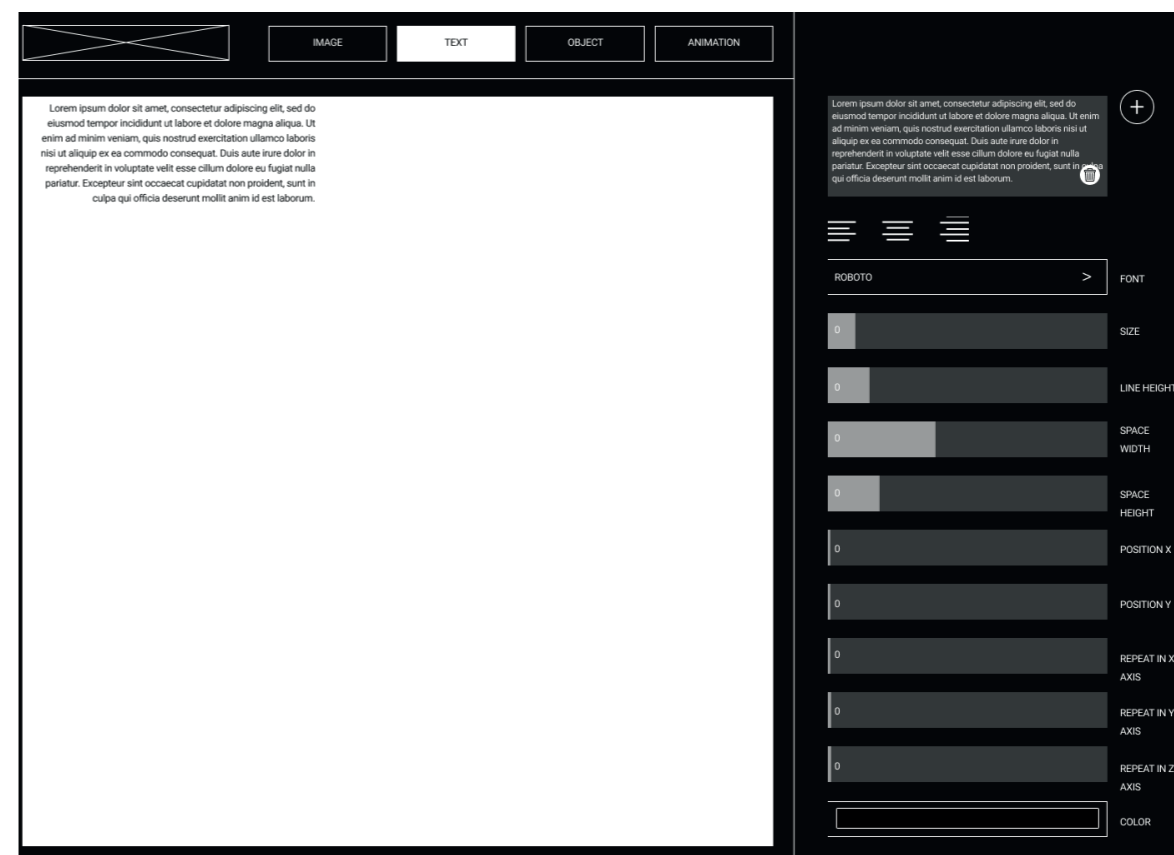
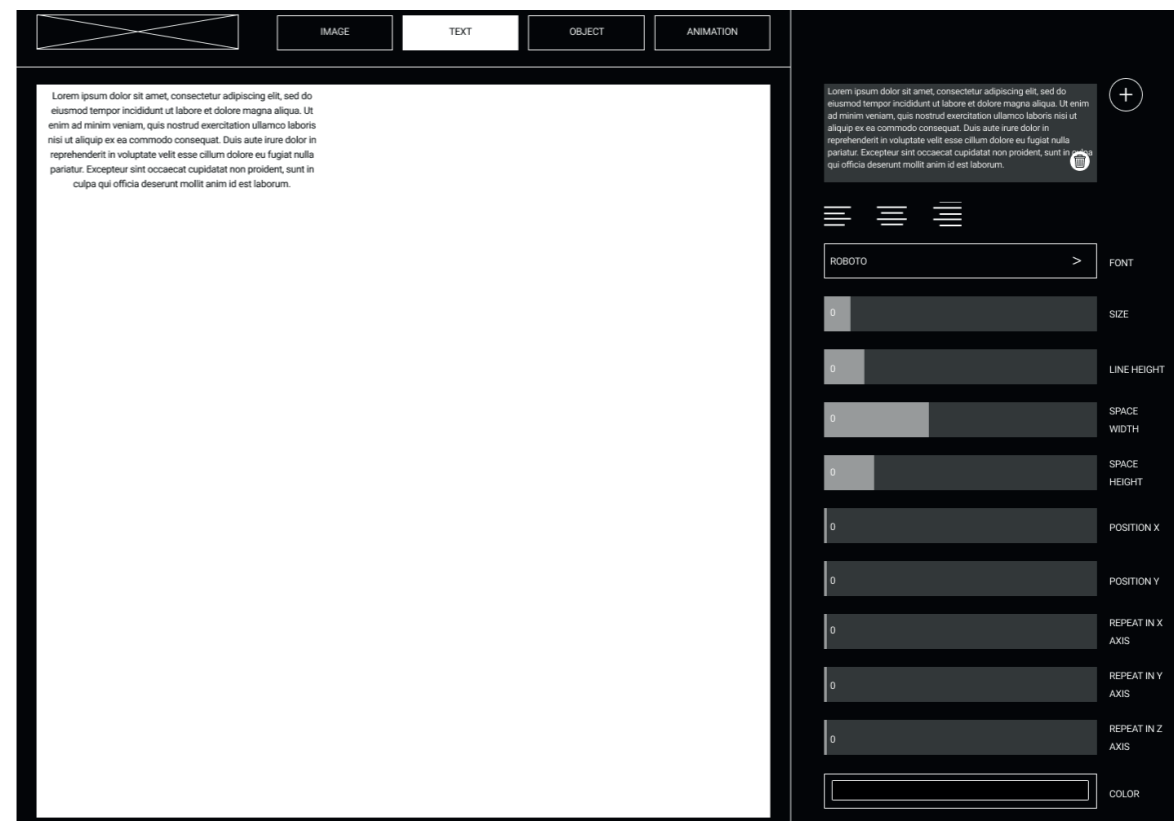
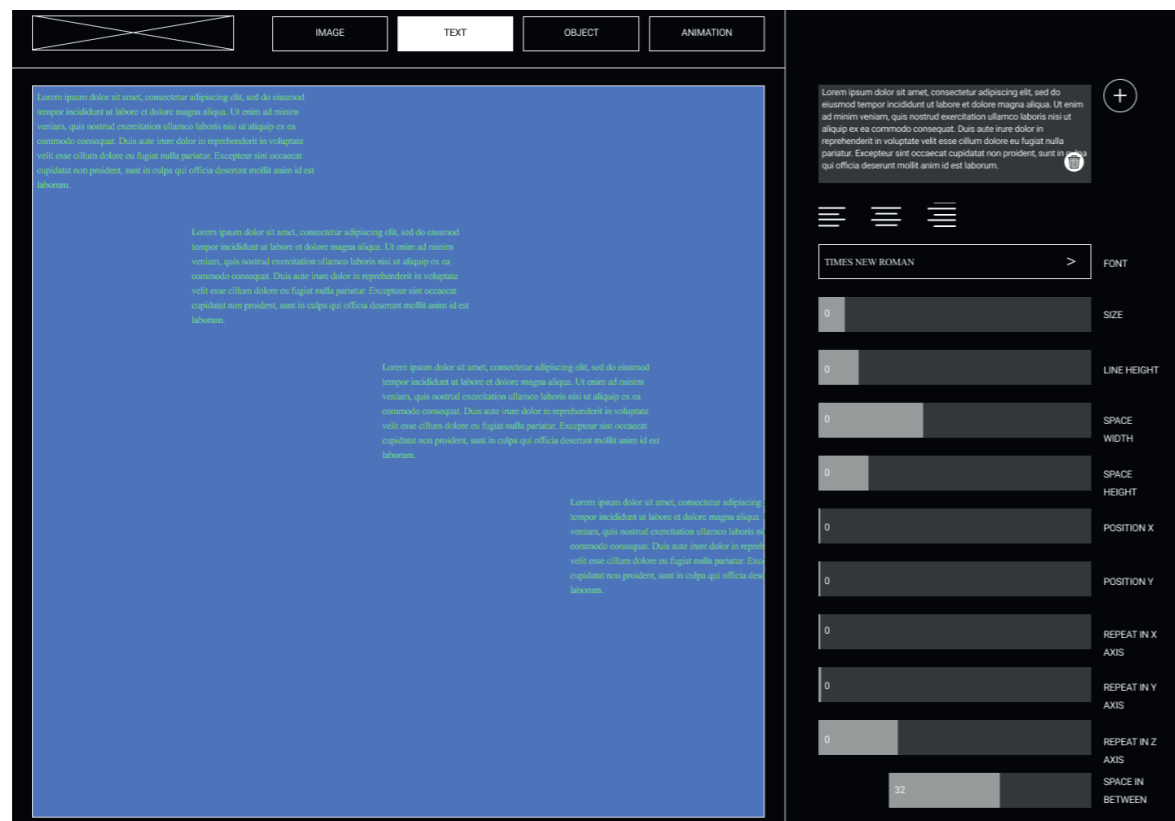


Fig 56 - Segunda versão dos wireframes da secção do texto

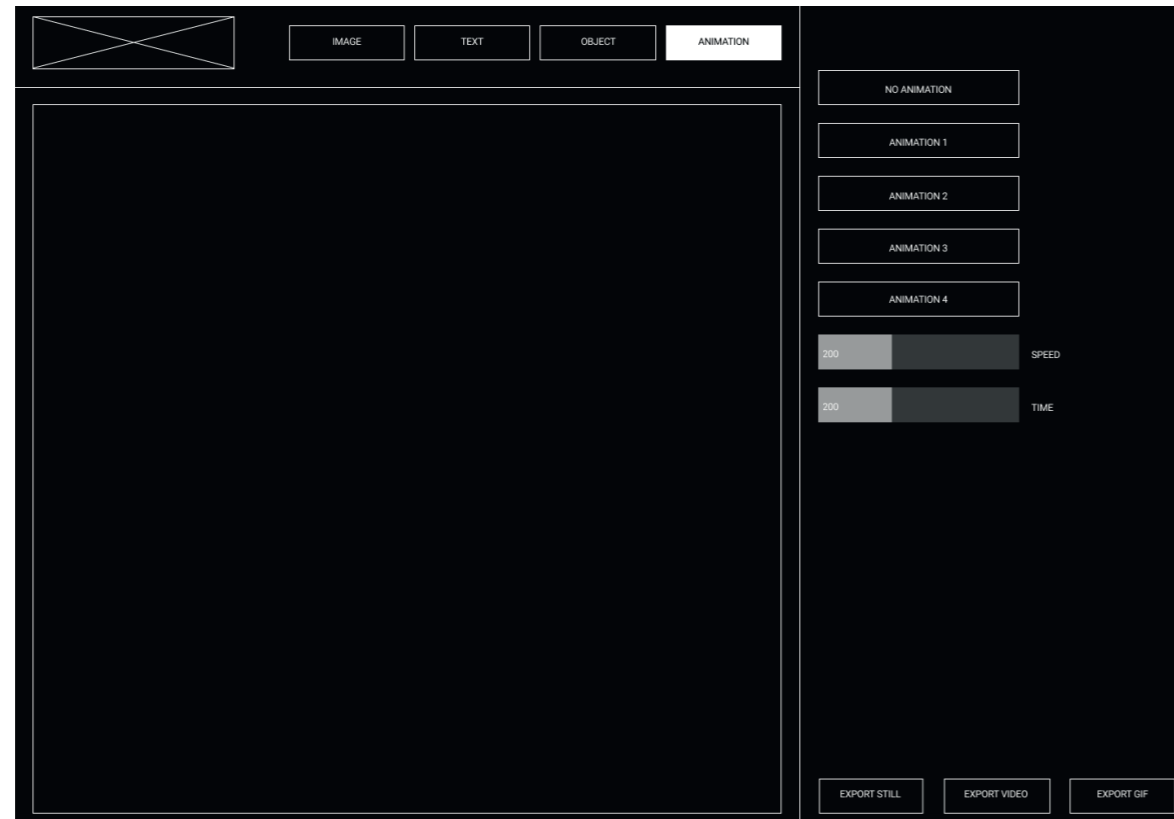
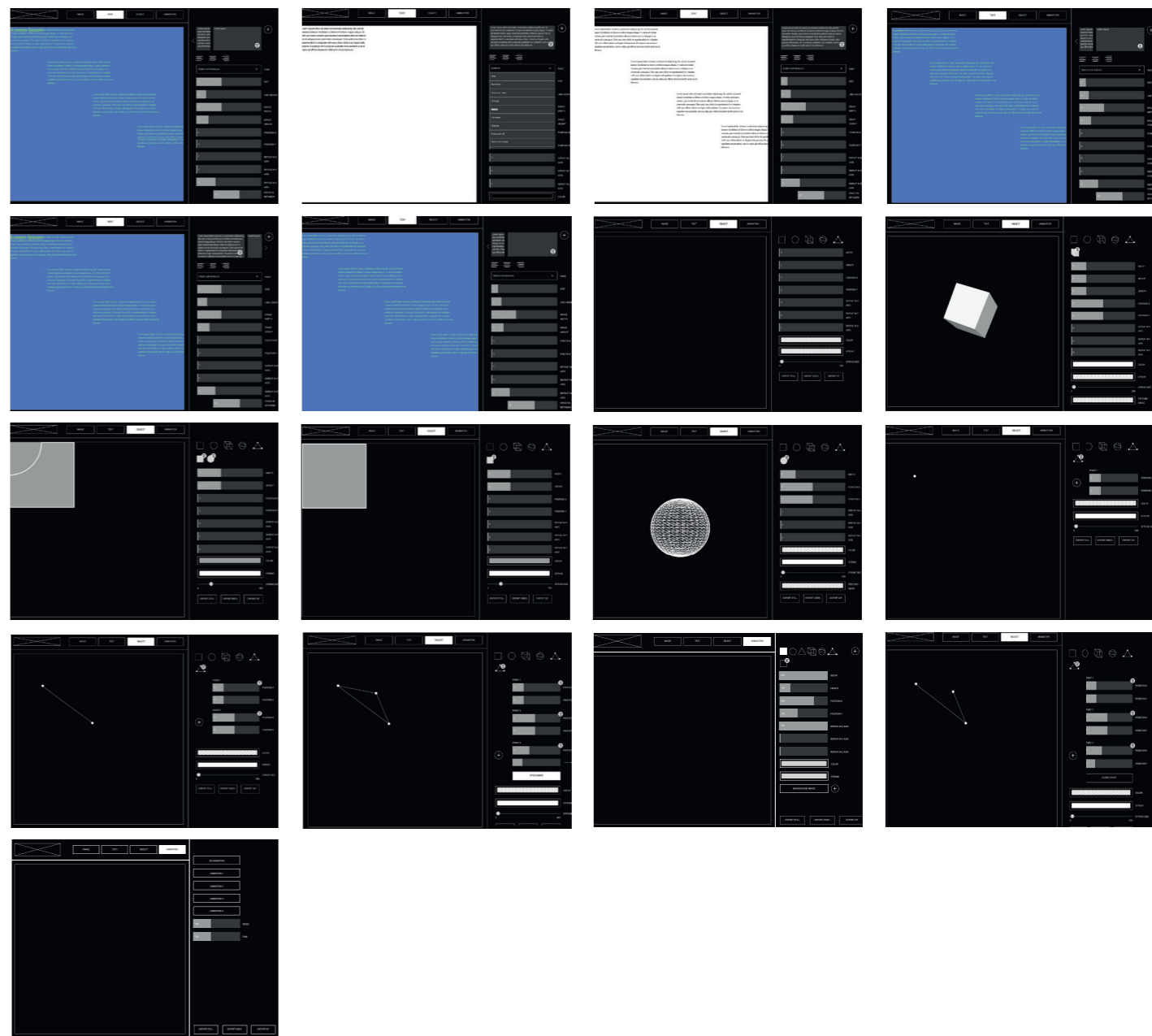
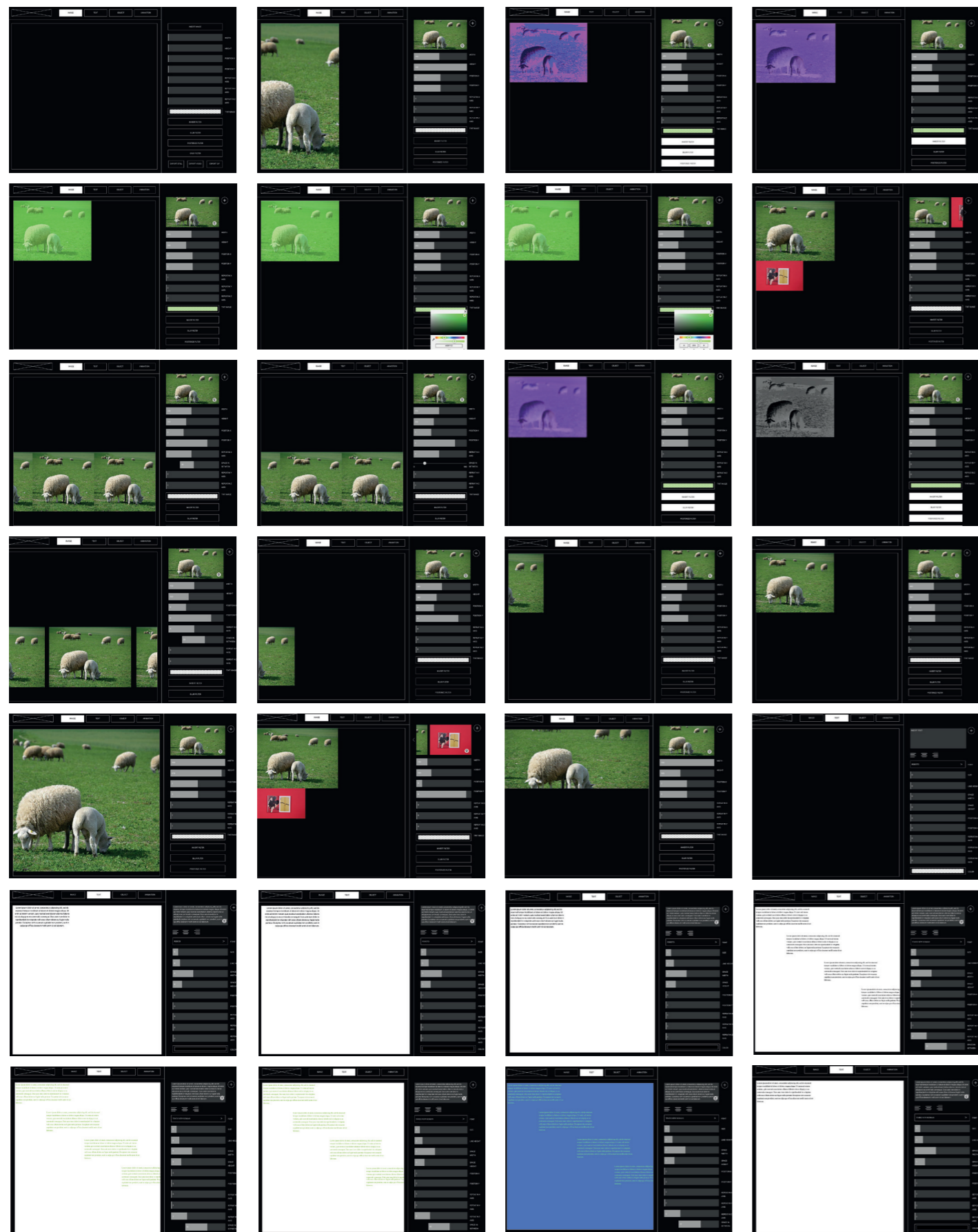


Fig 57 - Segunda versão dos wireframes da secção das animações



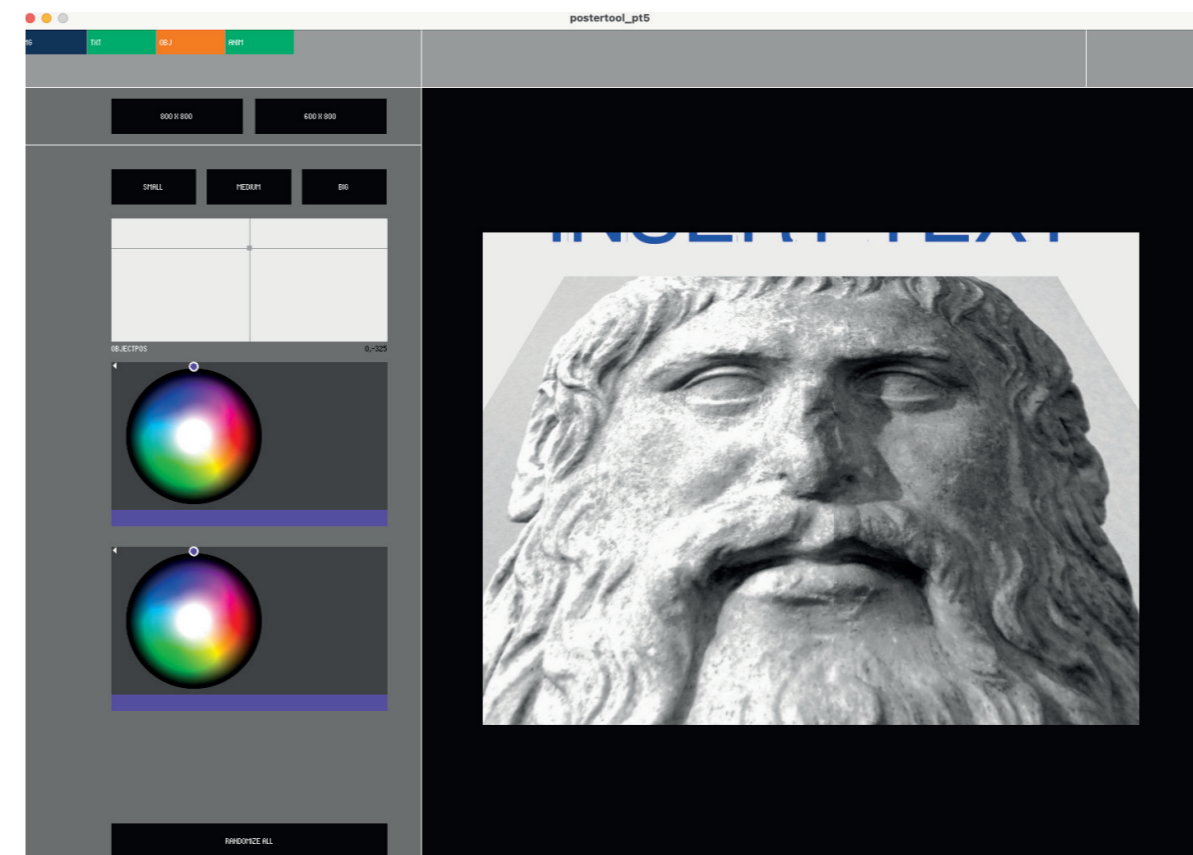
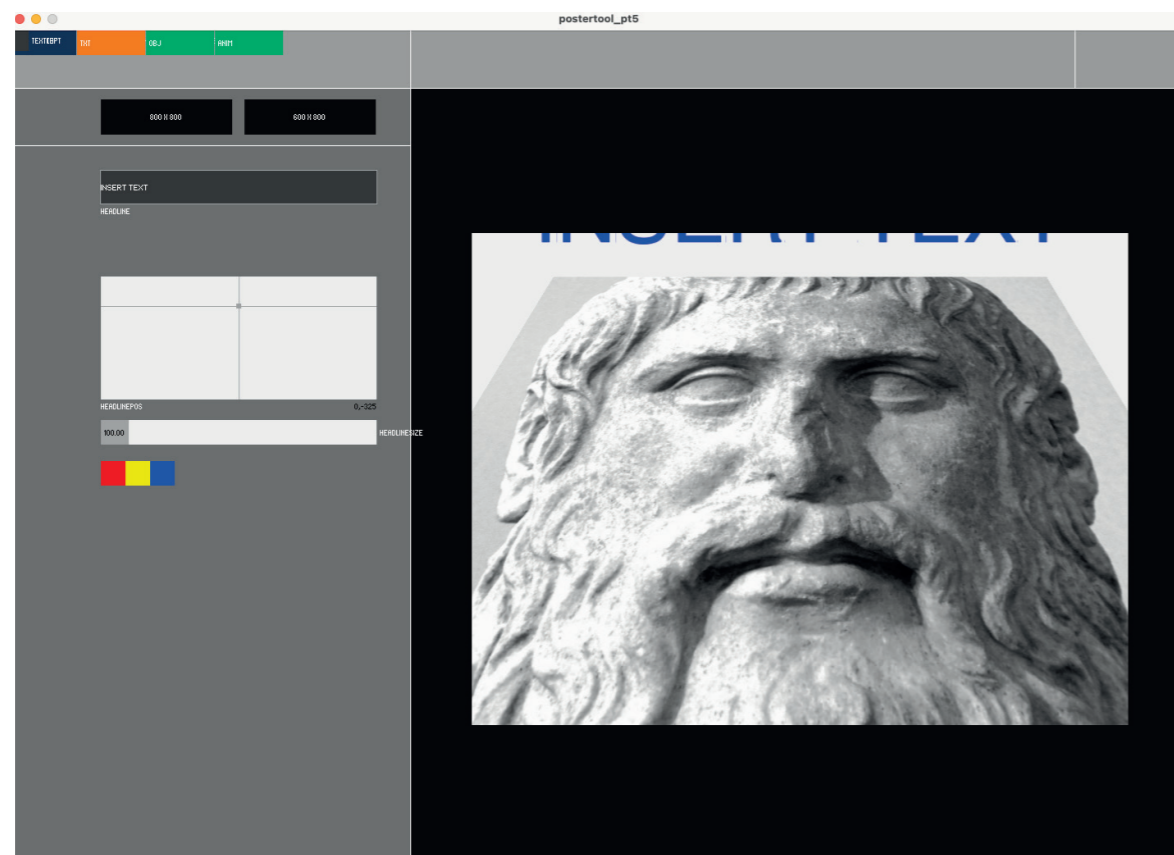
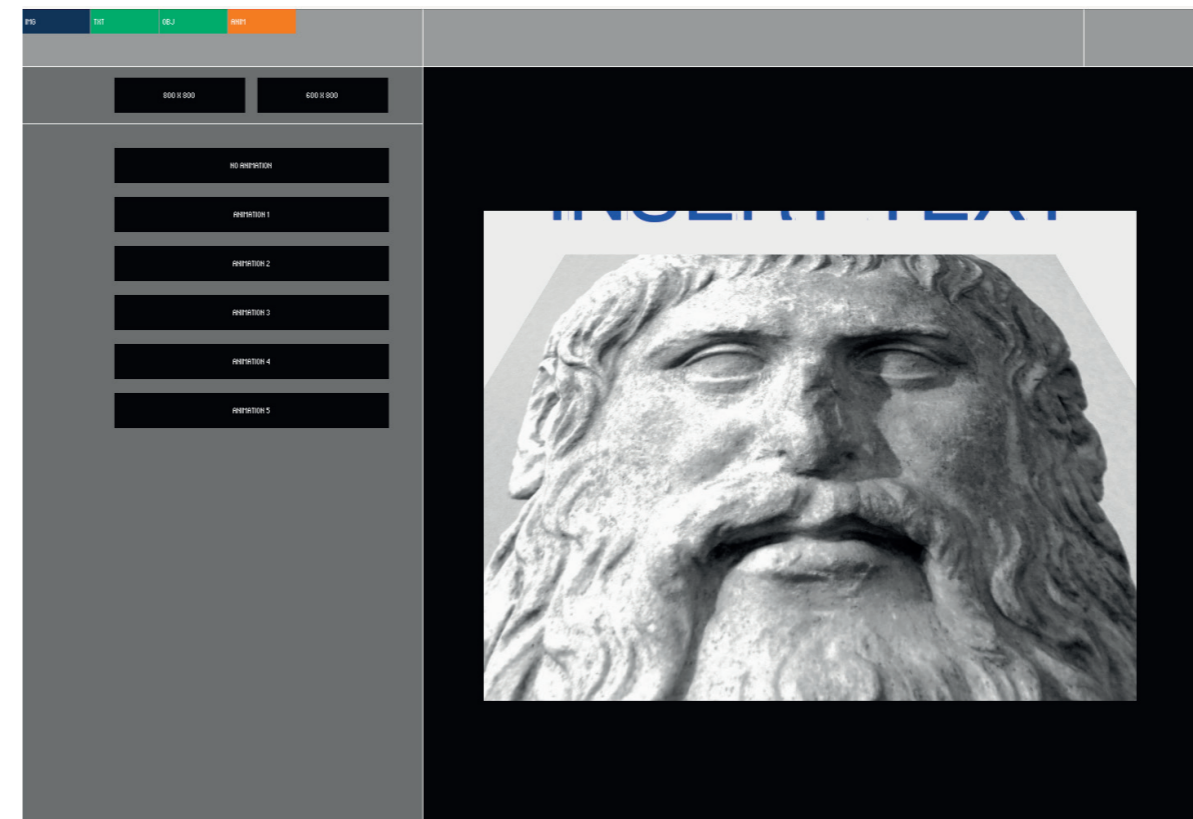
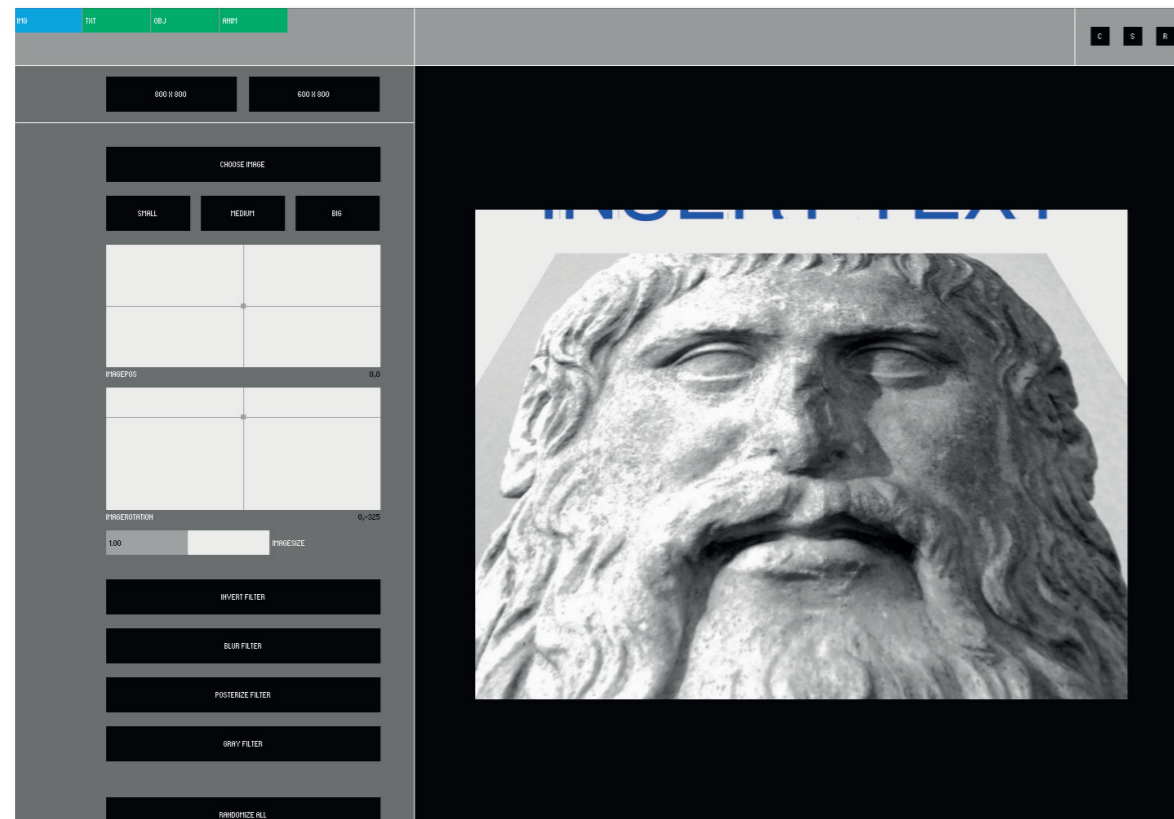


Fig 58 - MVP realizada no Processing

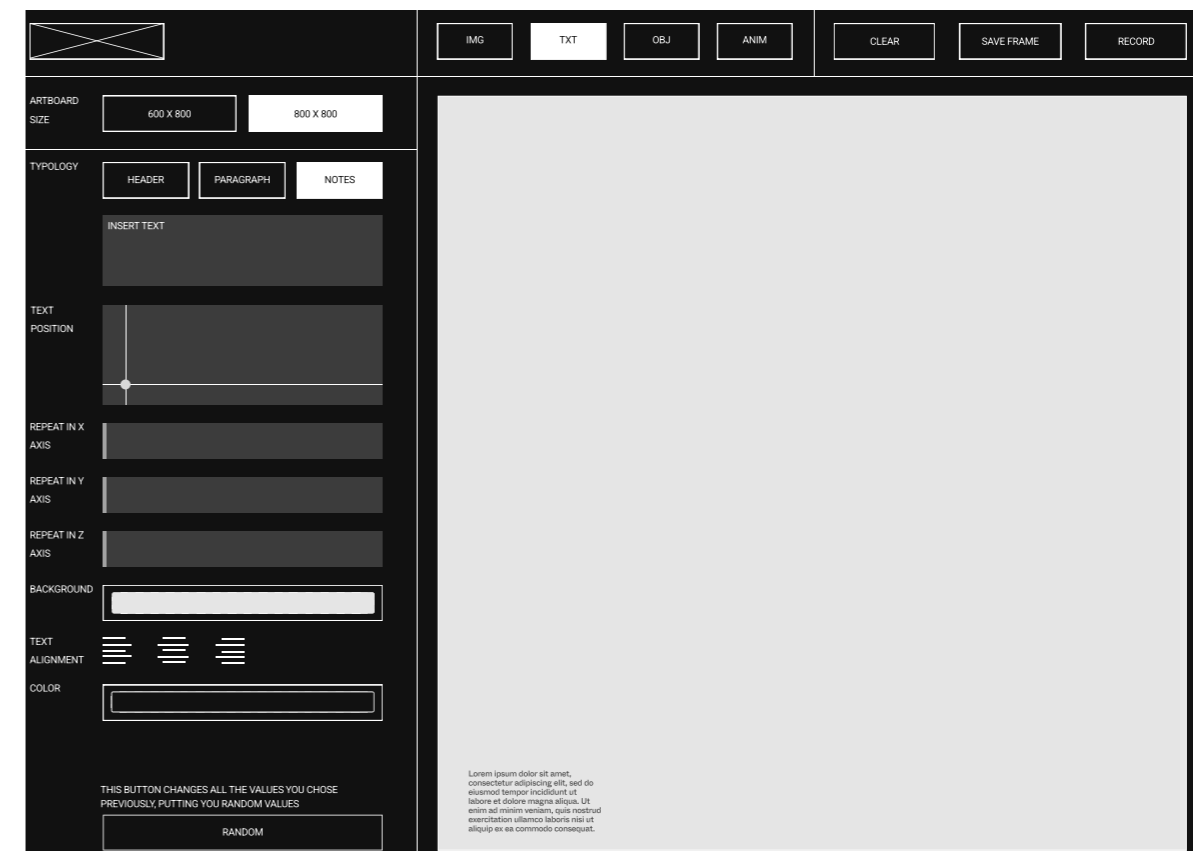
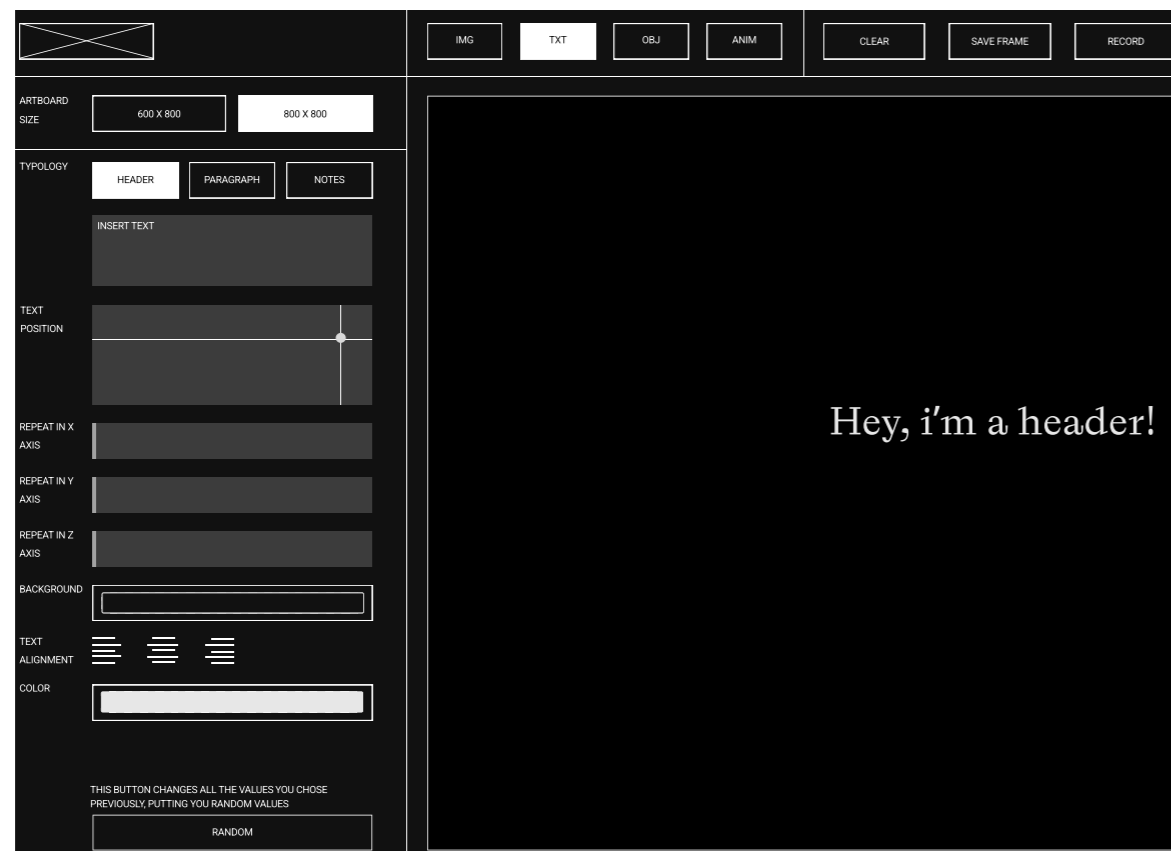
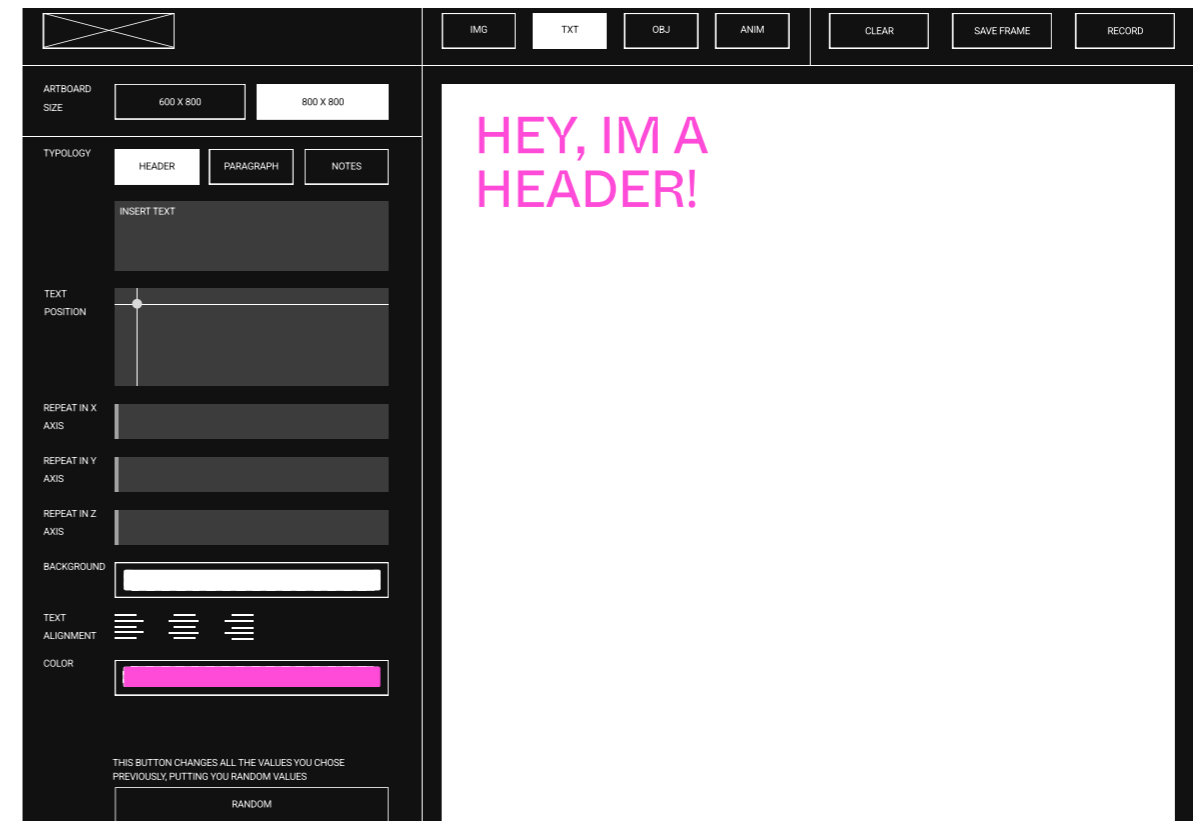
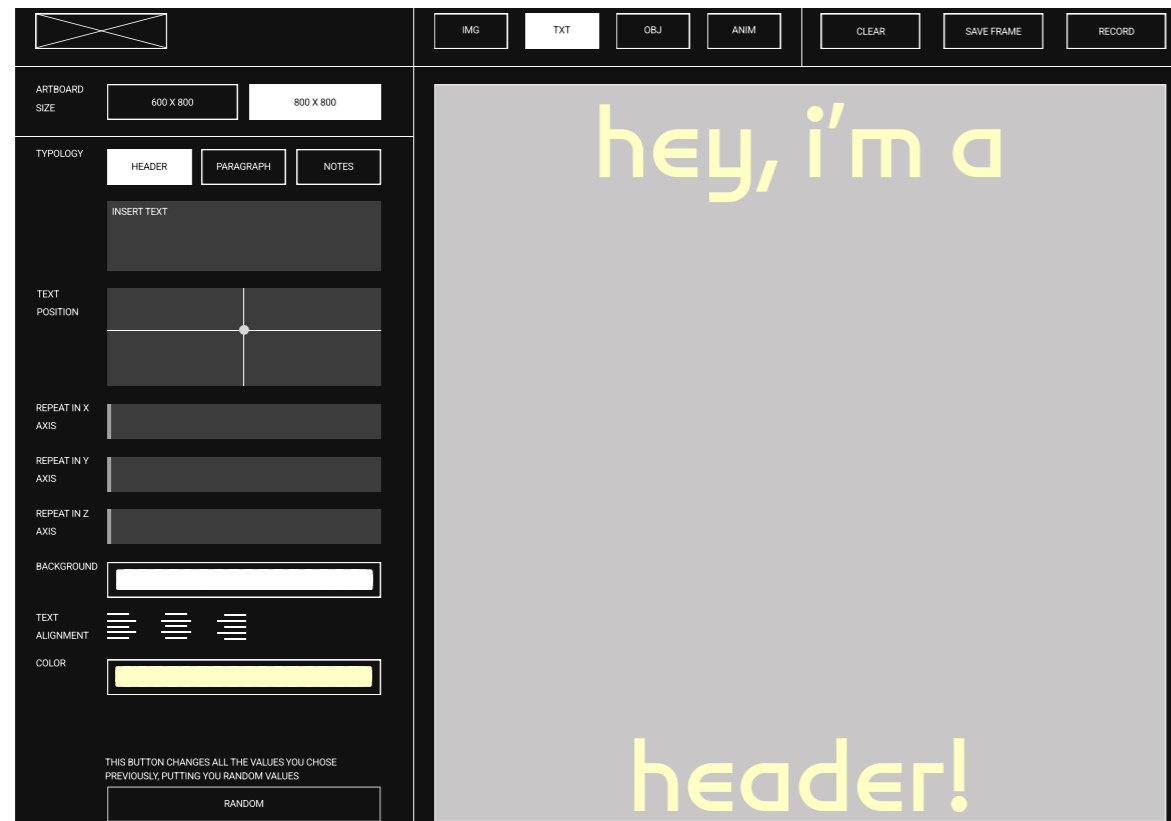


Fig 59 - Terceira versão dos wireframes da secção do texto

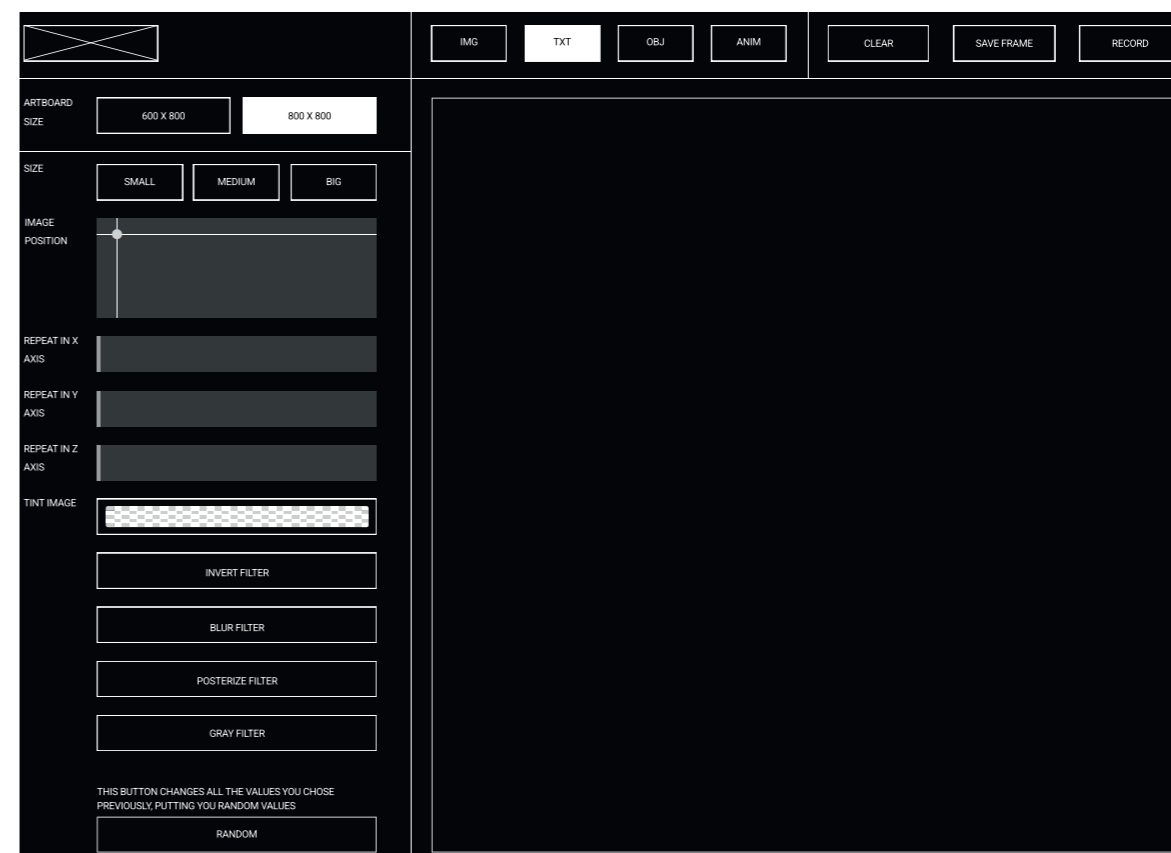
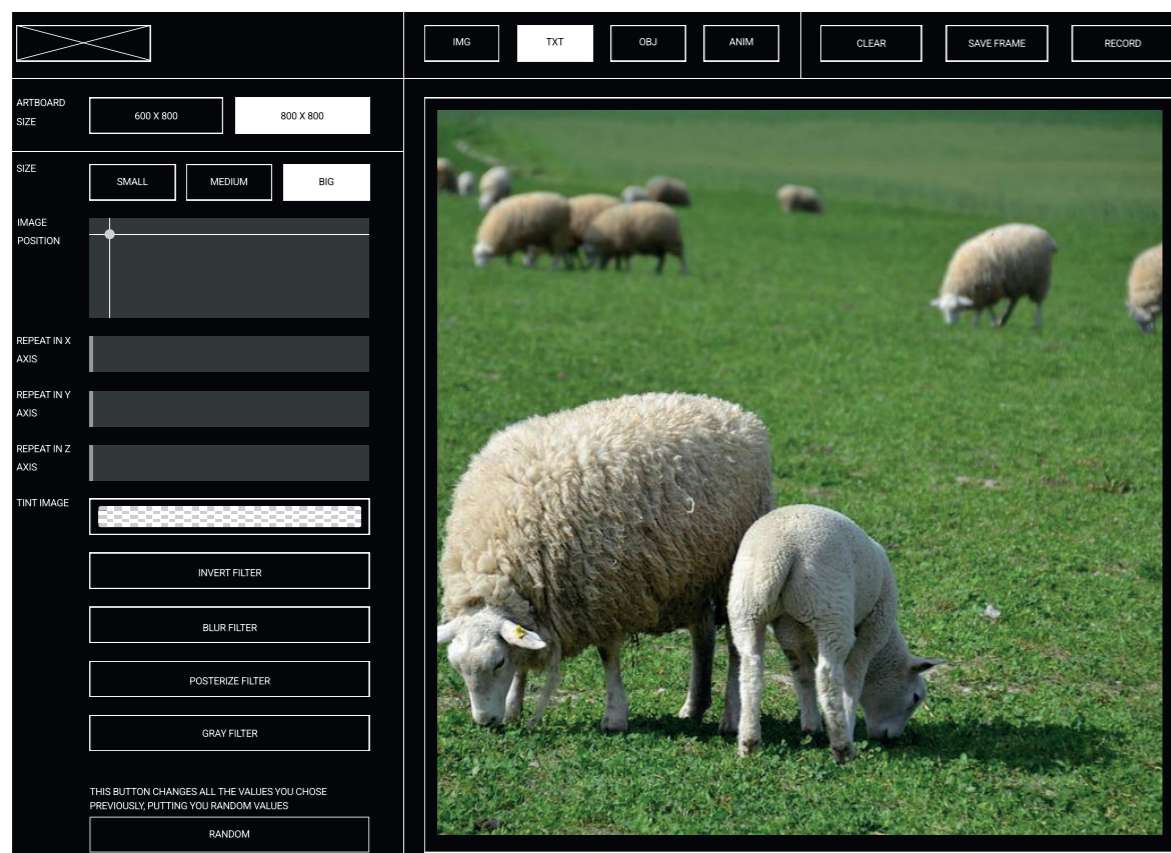
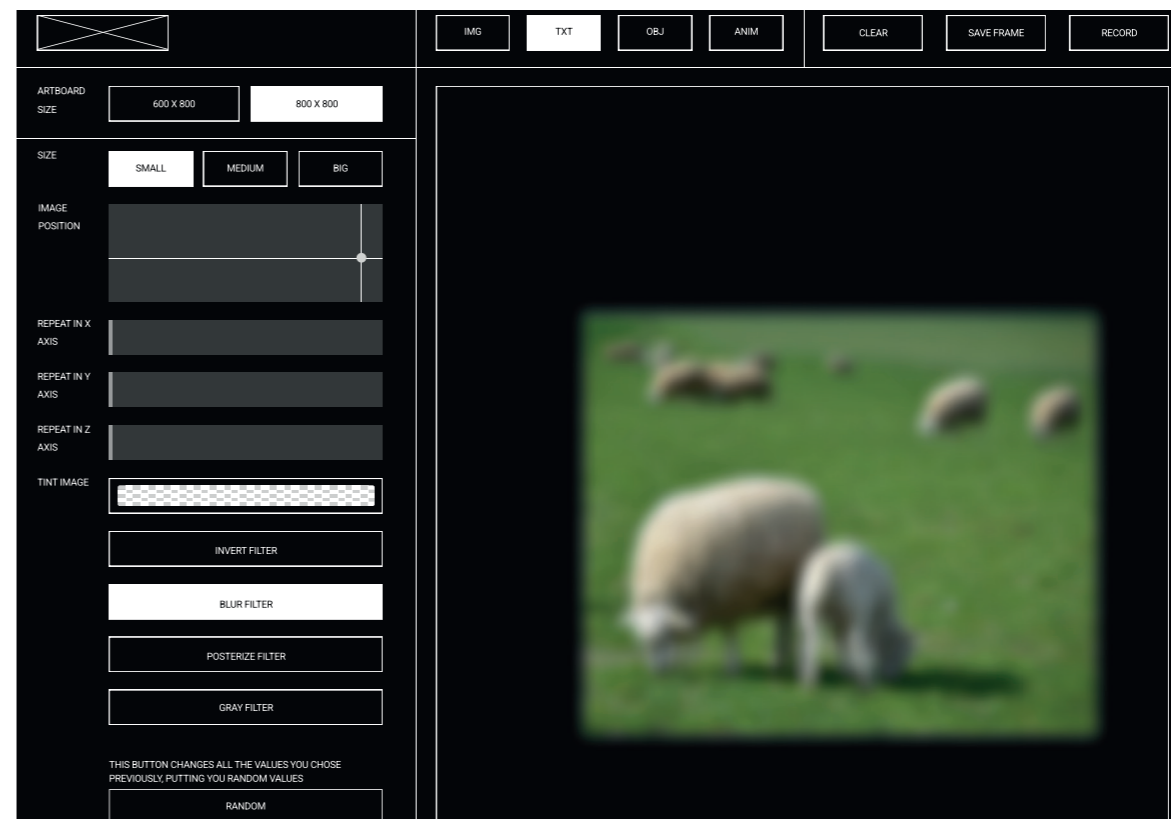
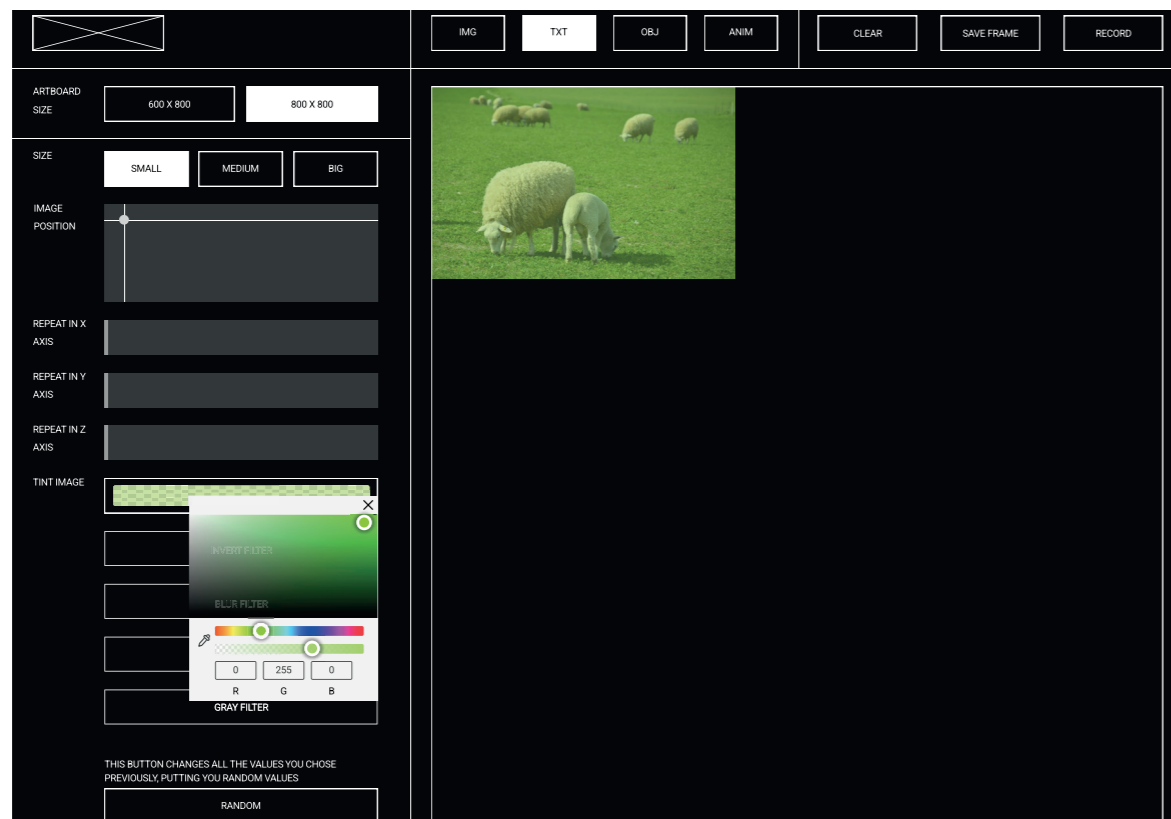


Fig 60 - Terceira versão dos wireframes da secção das imagens

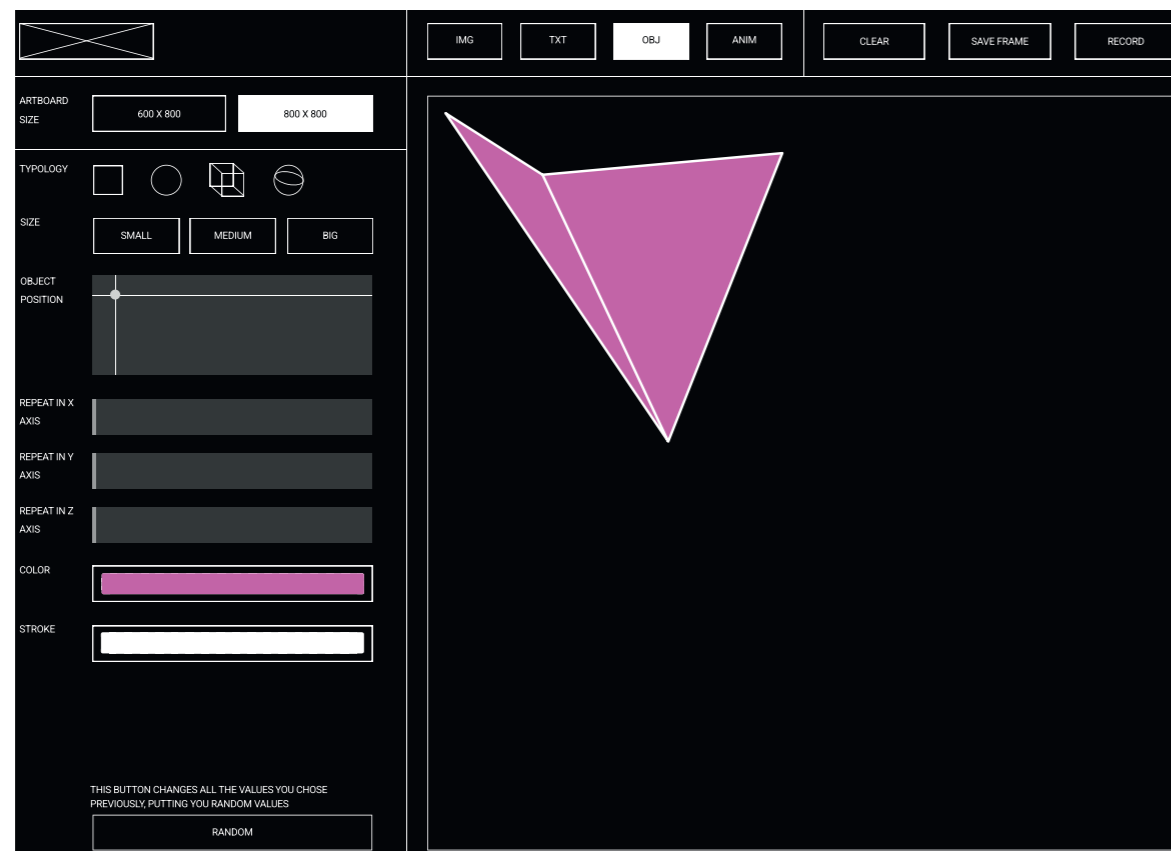
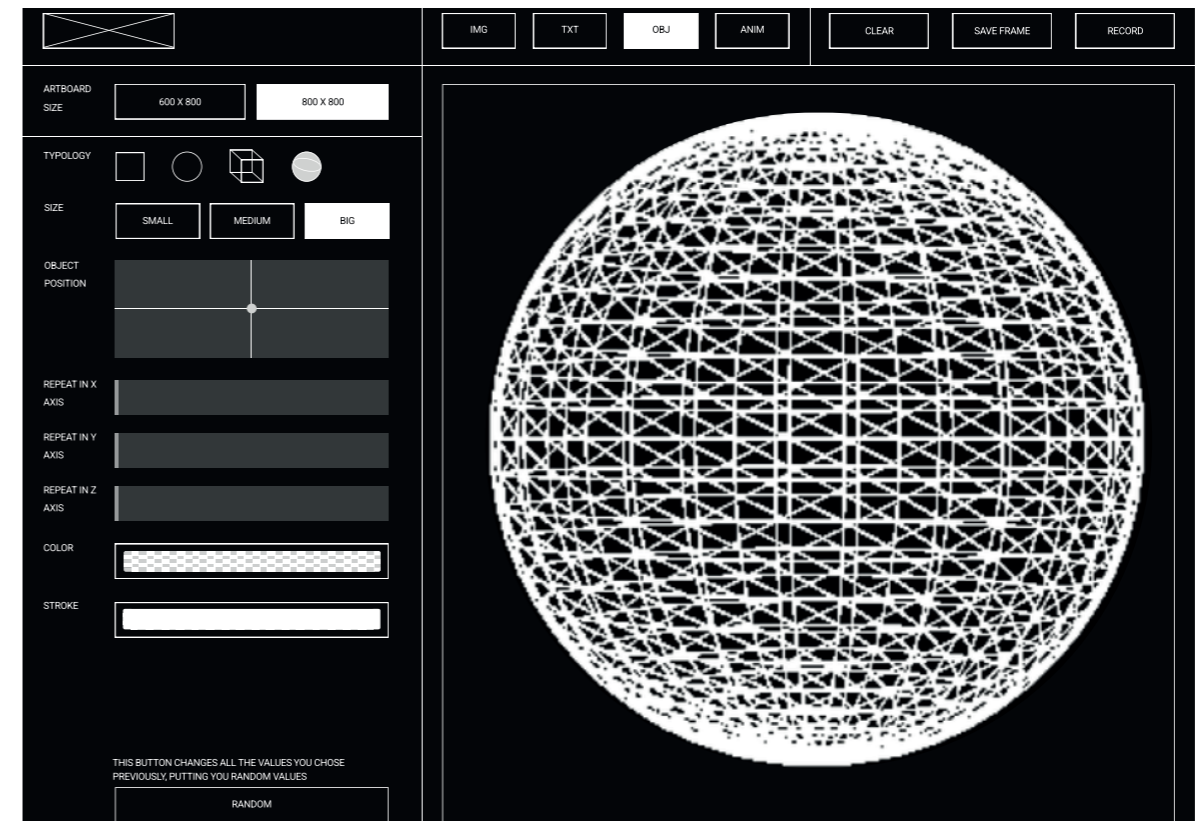
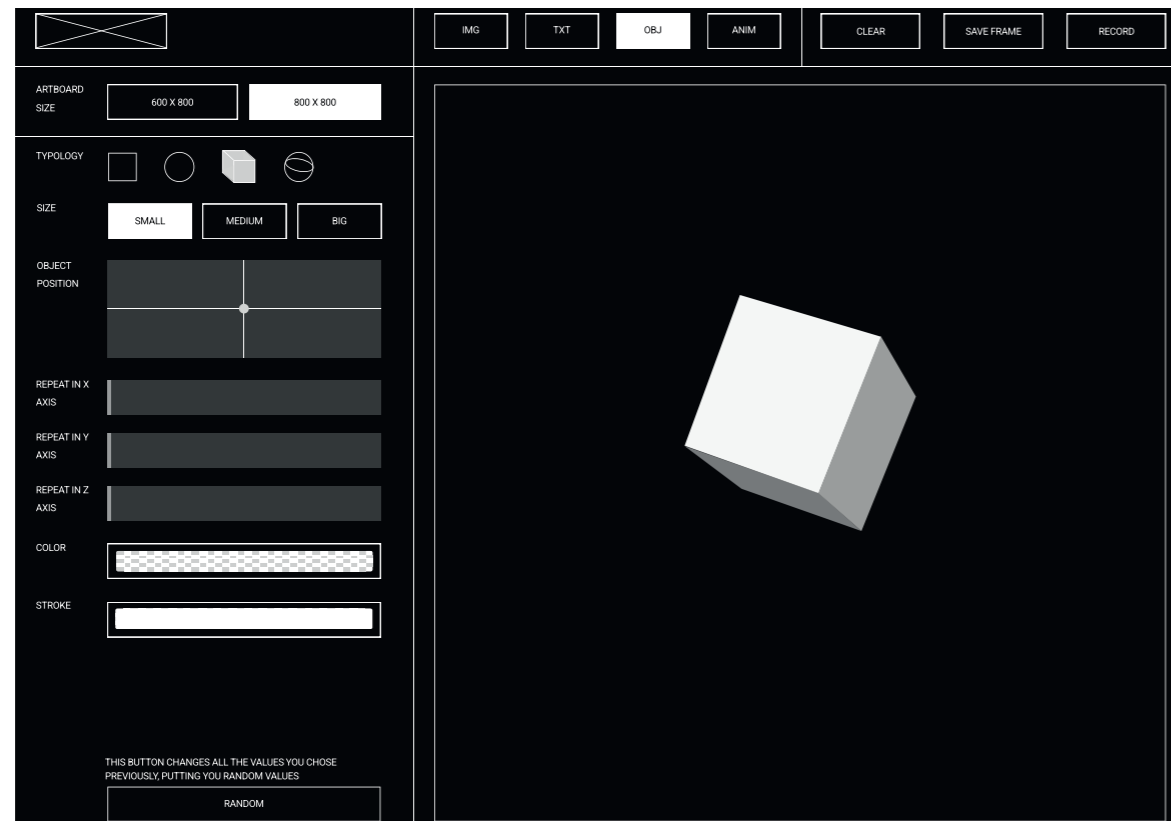


Fig 61 - Terceira versão dos wireframes da secção dos objectos

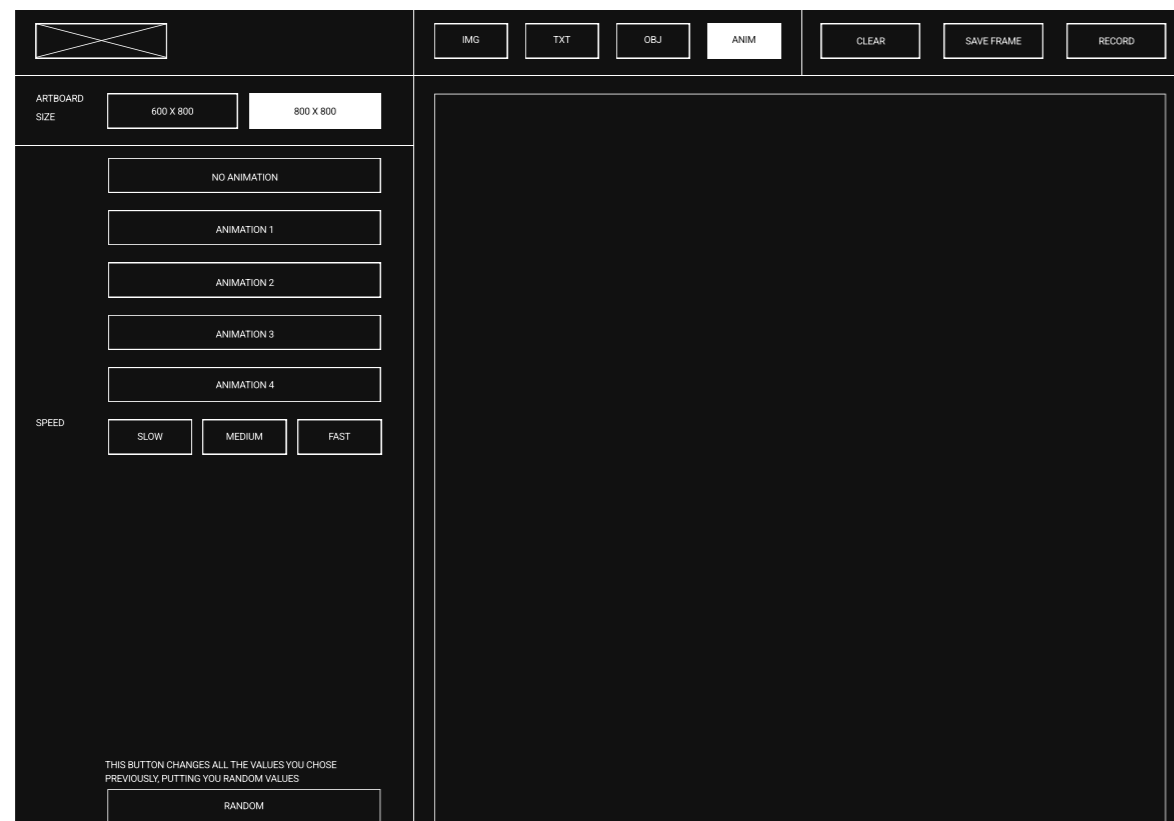
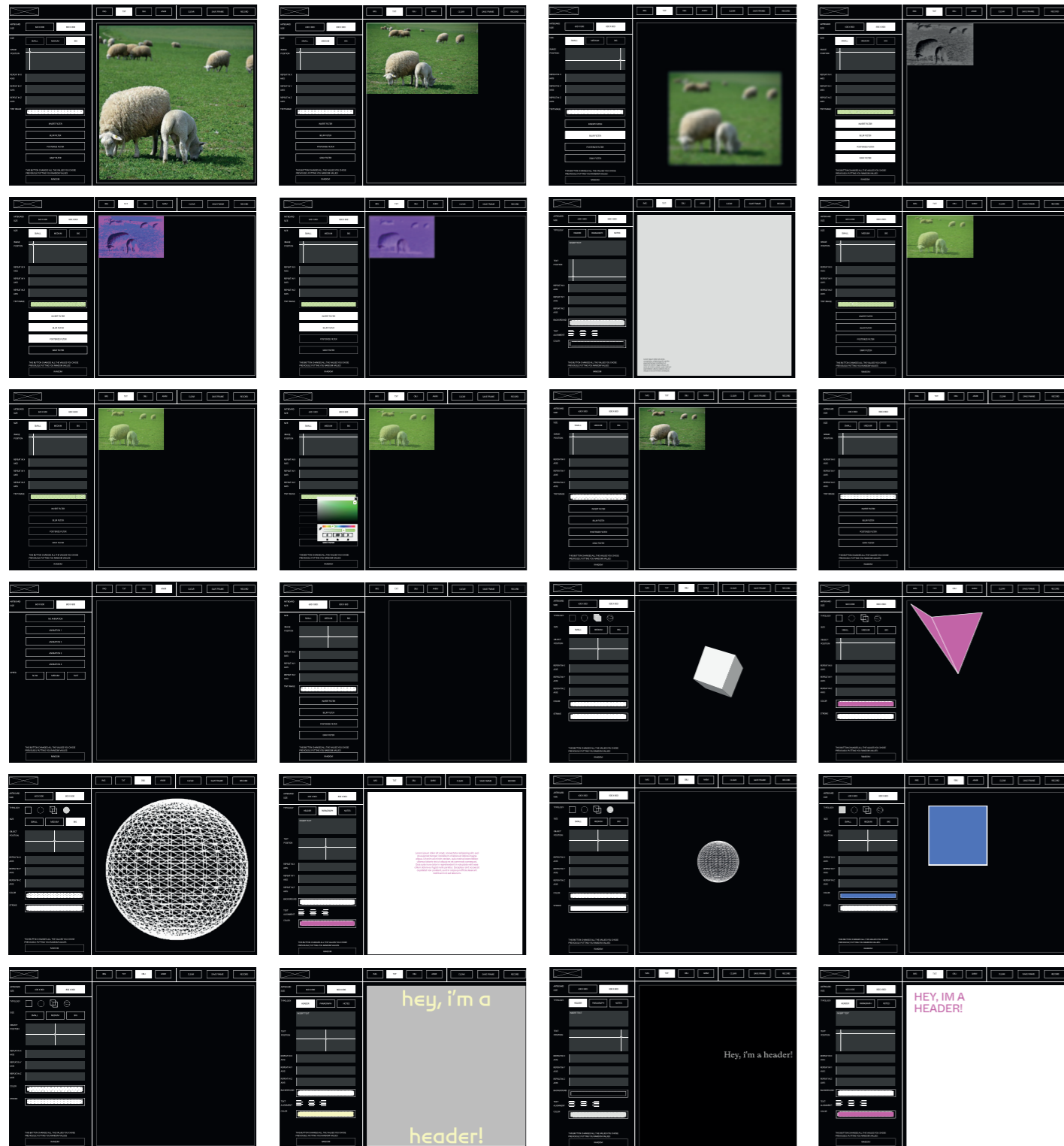


Fig 62 - Terceira versão dos wireframes da secção das animações



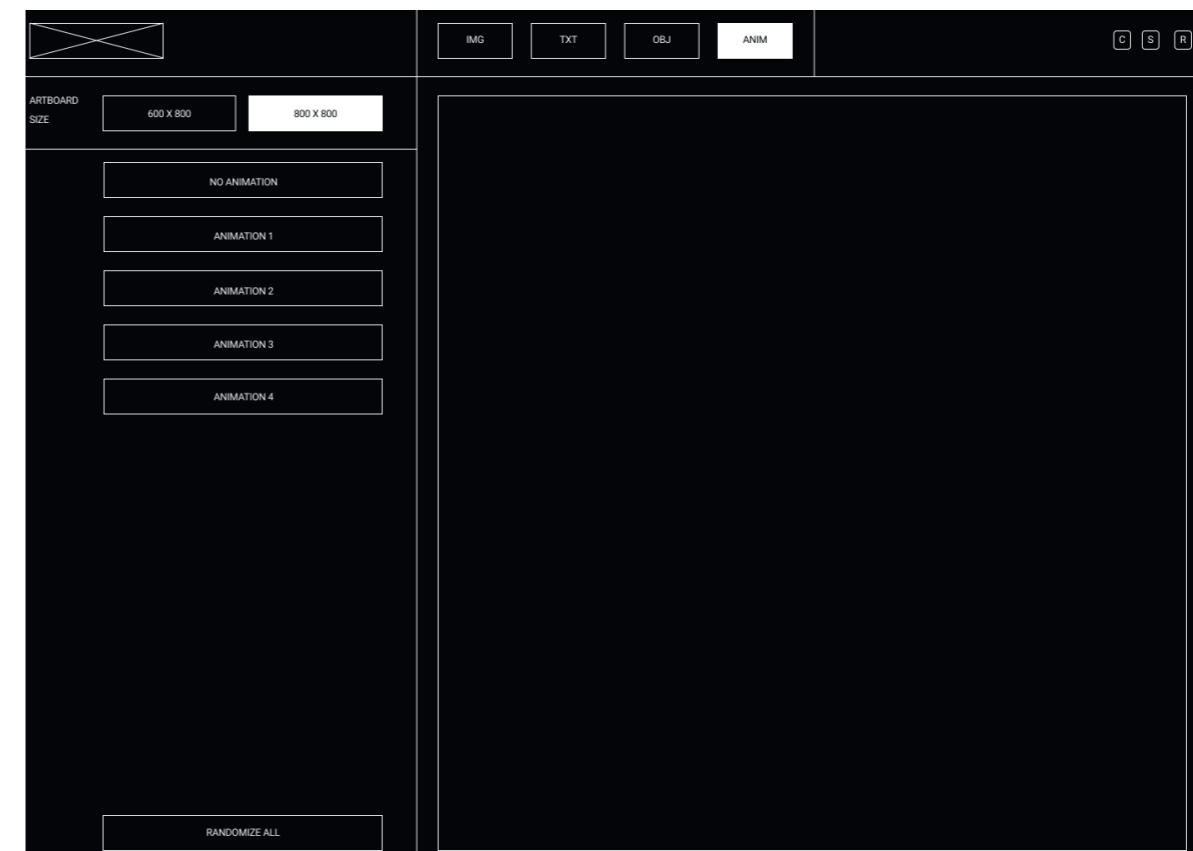
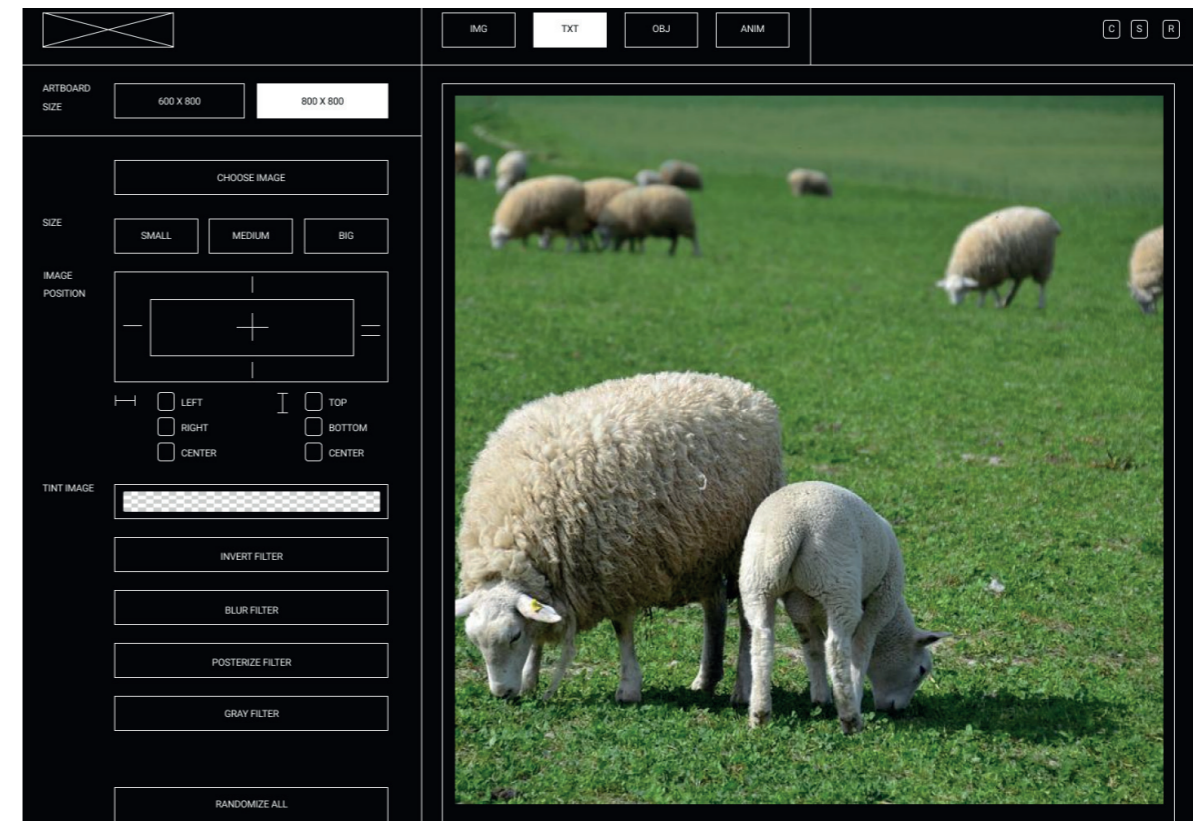
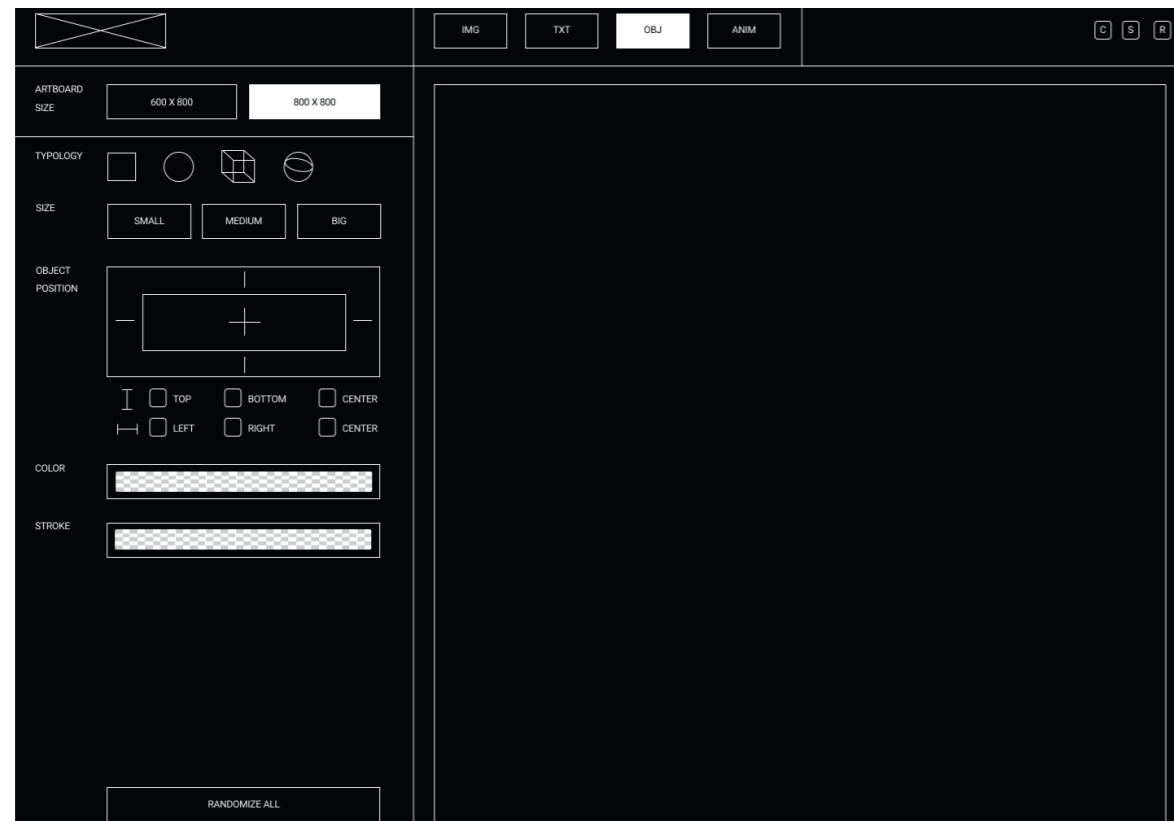
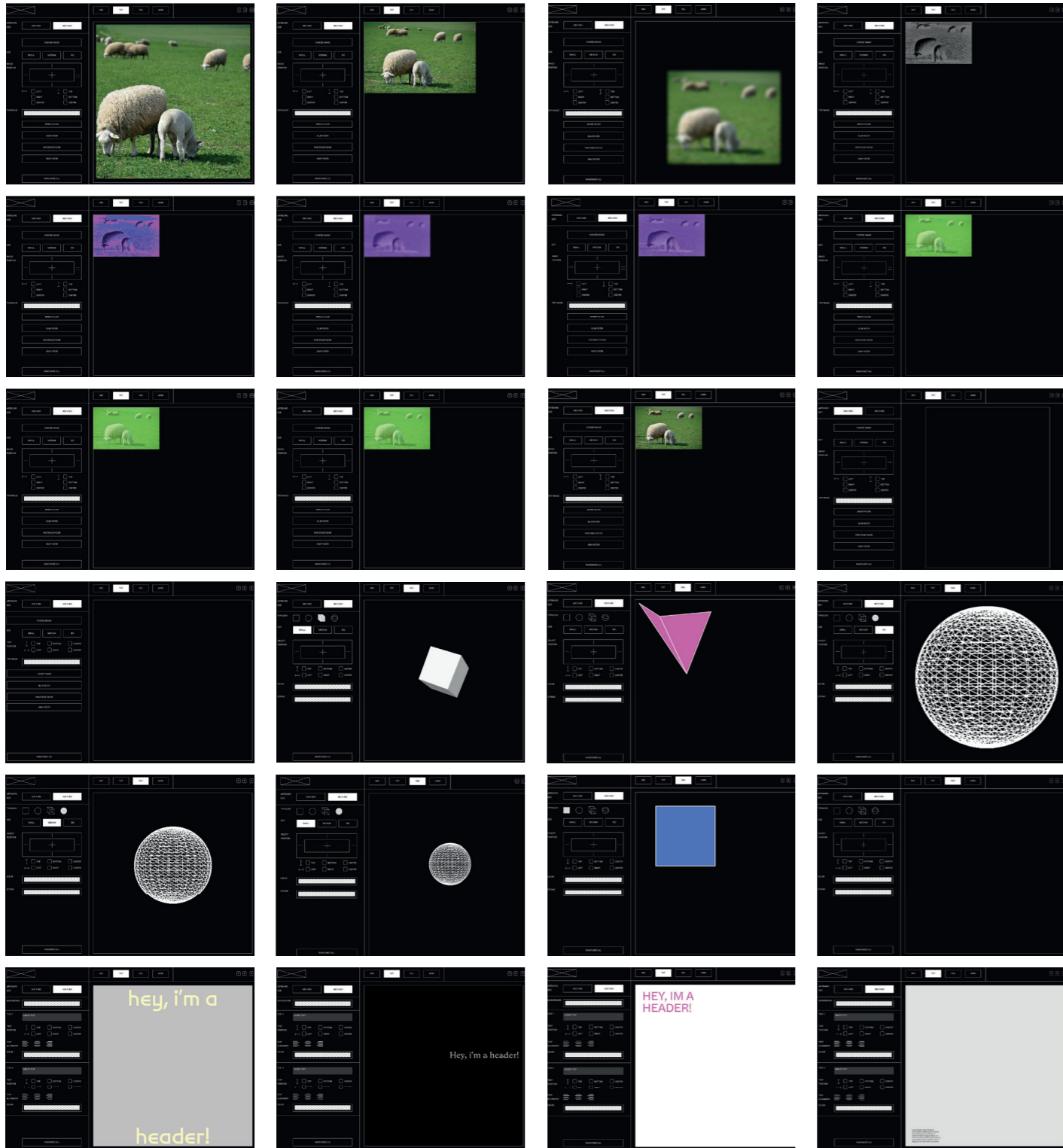


Fig 63 - Quarta versão dos wireframes



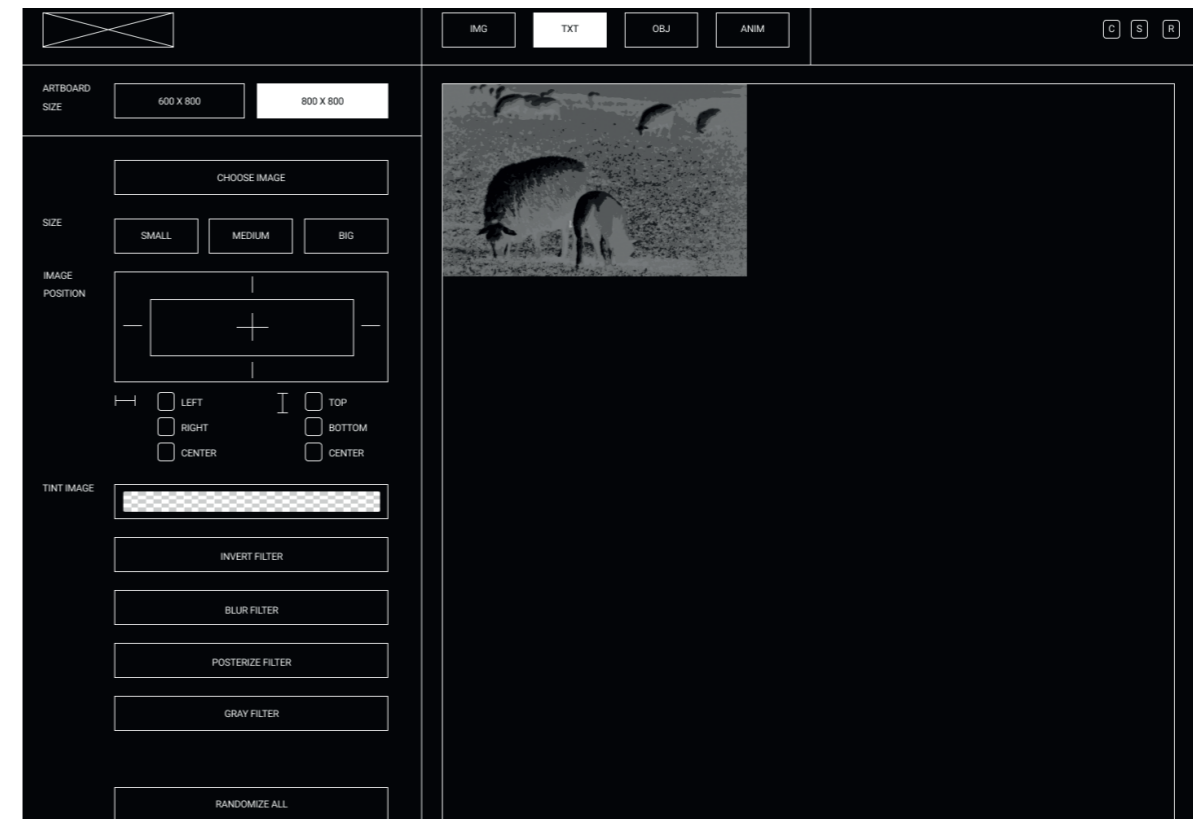
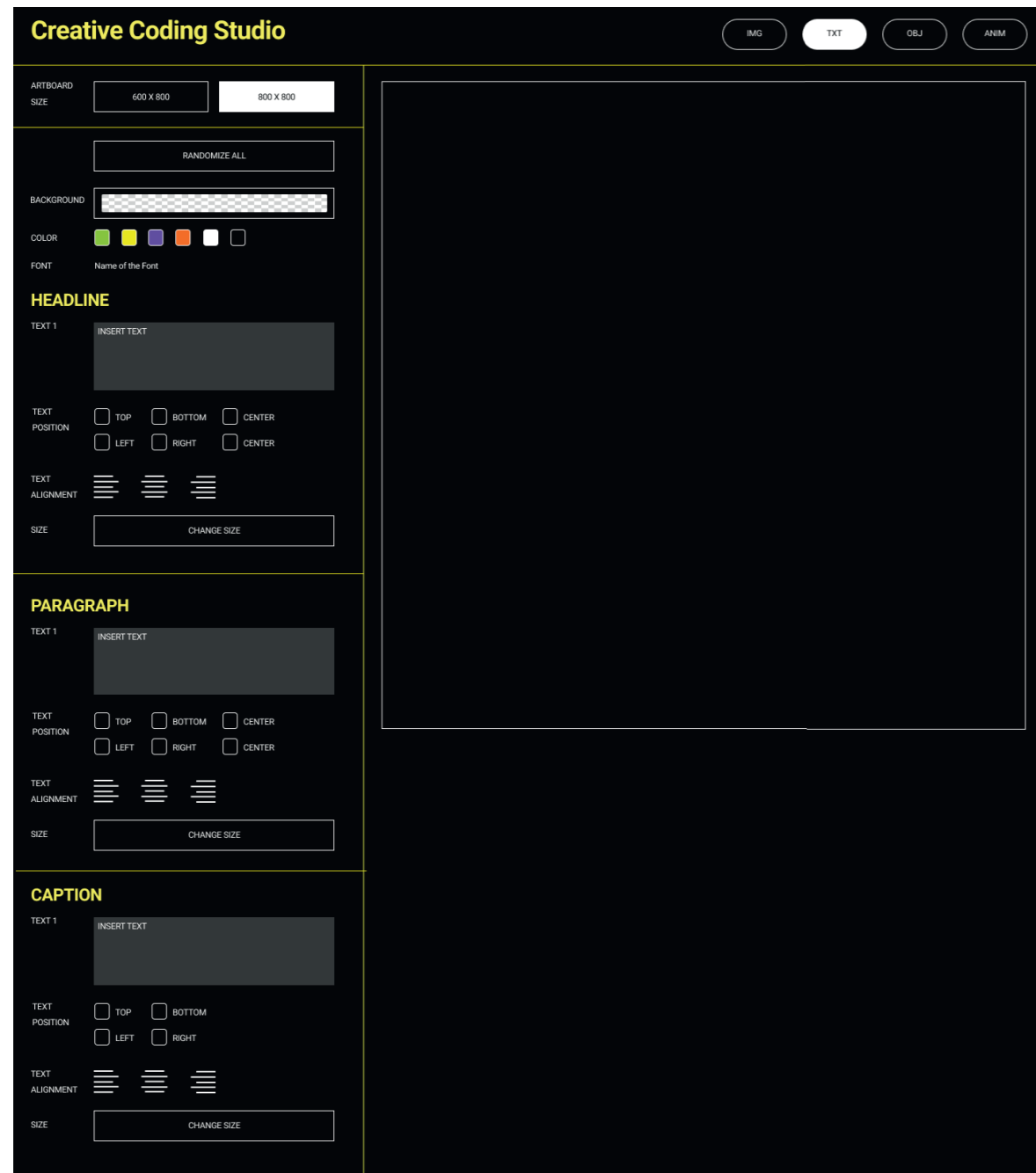


Fig 64 - Quarta versão dos wireframes

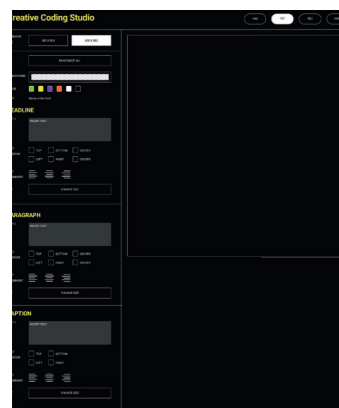
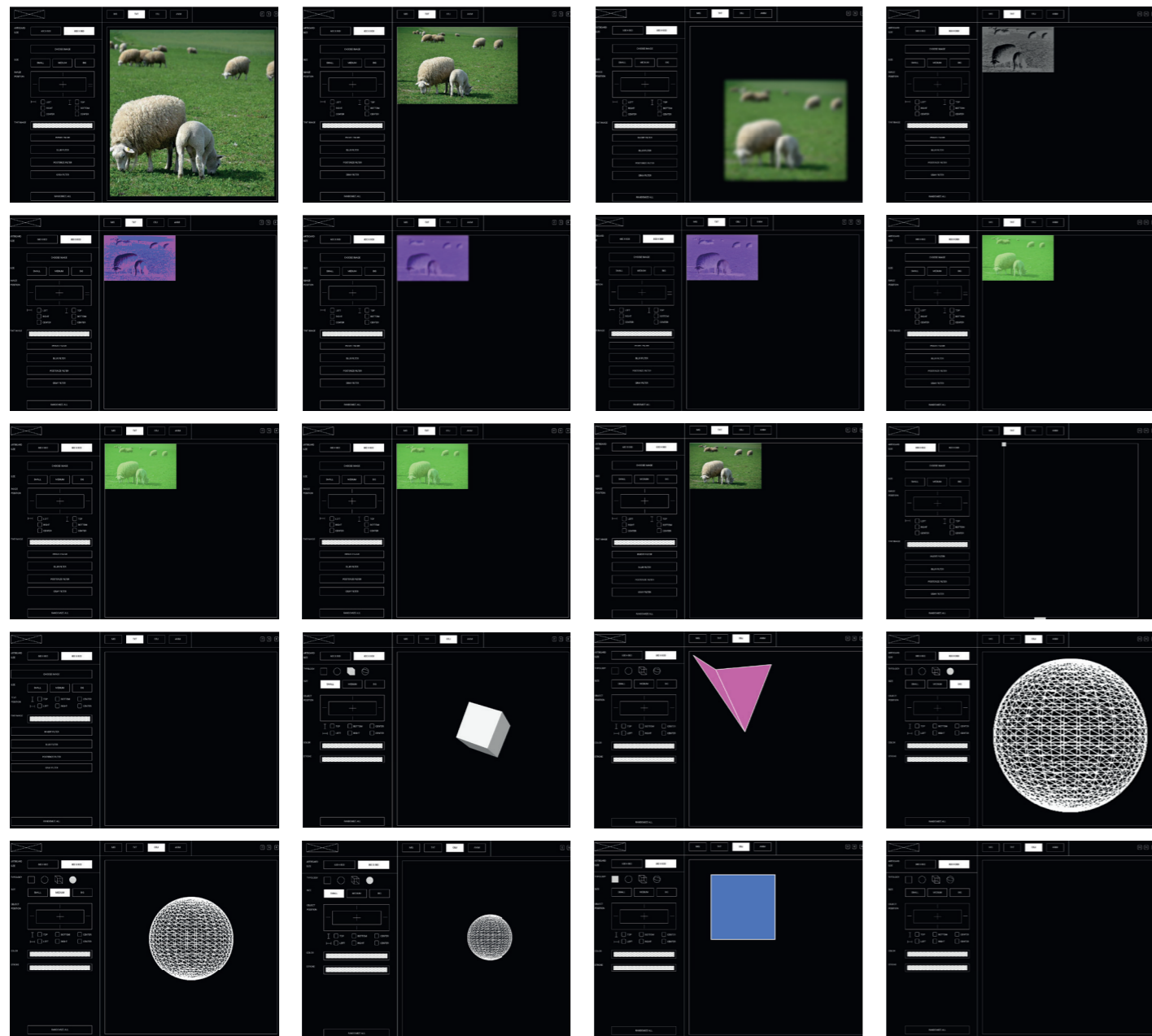




Fig 65 - Resultado planeado estaticamente no Figma (800 x 800)



Fig 66 - Resultado planeado estaticamente no Figma (800 x 800)

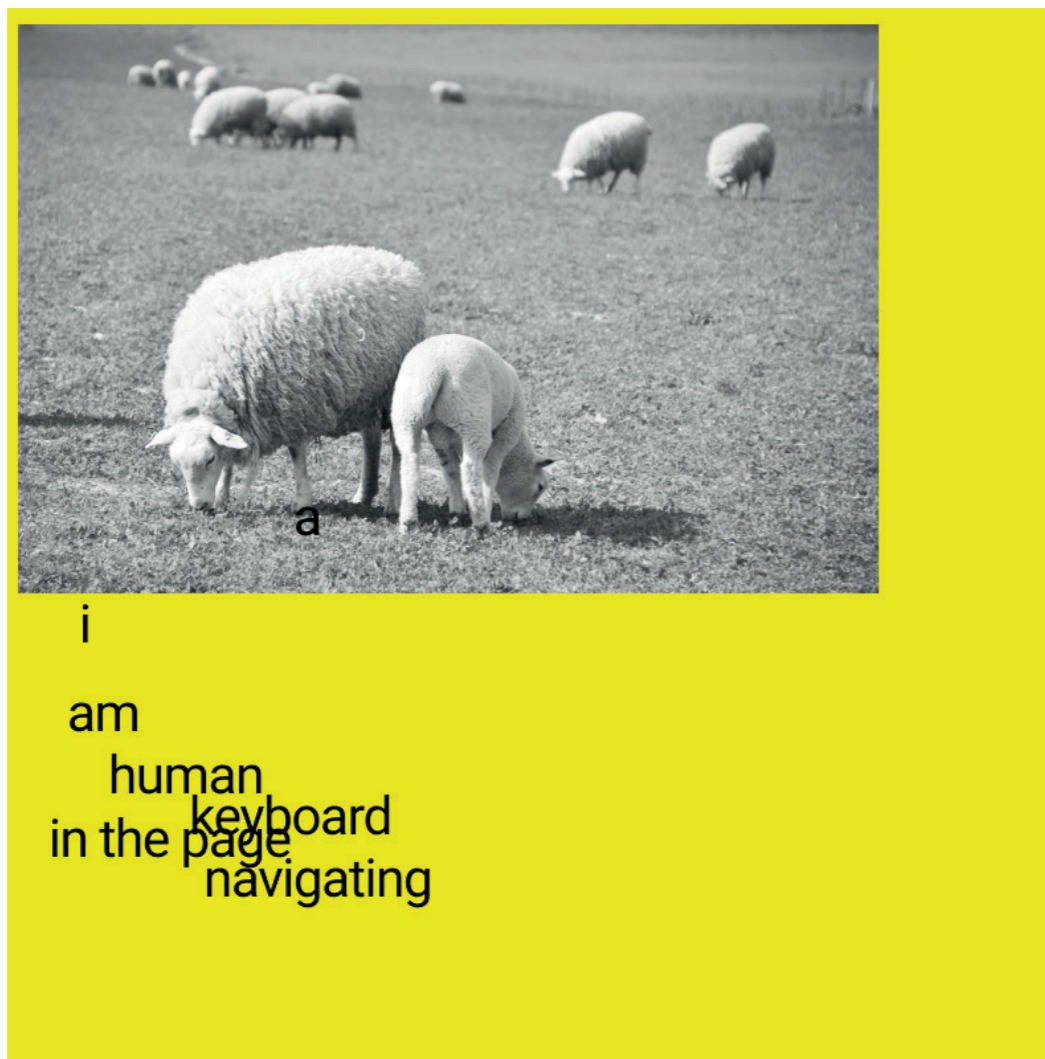


Fig 67 - Resultado planeados estaticamente no Figma (800 x 800)



Fig 68 - Resultado planeados estaticamente no Figma (600 x 800)

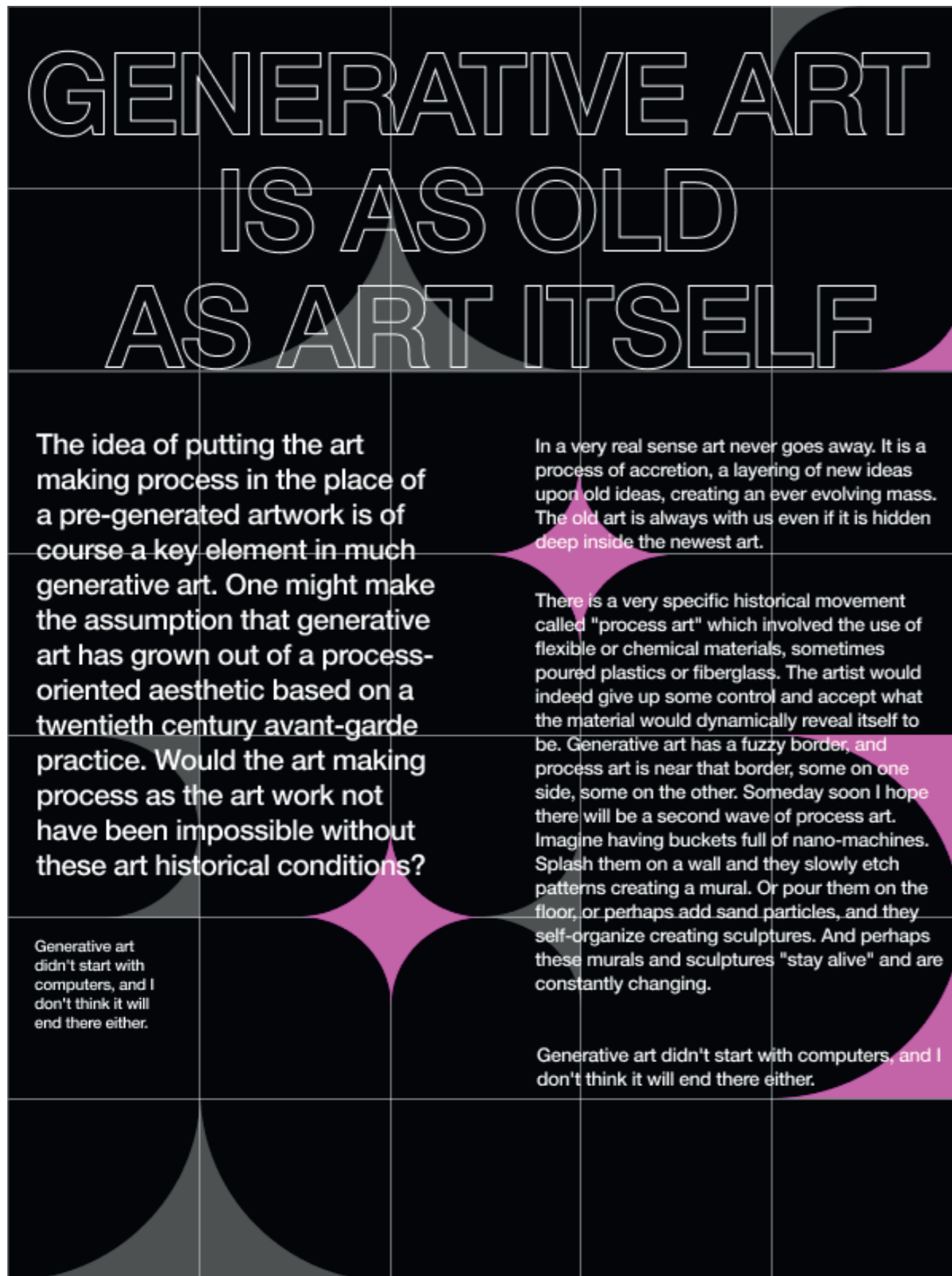
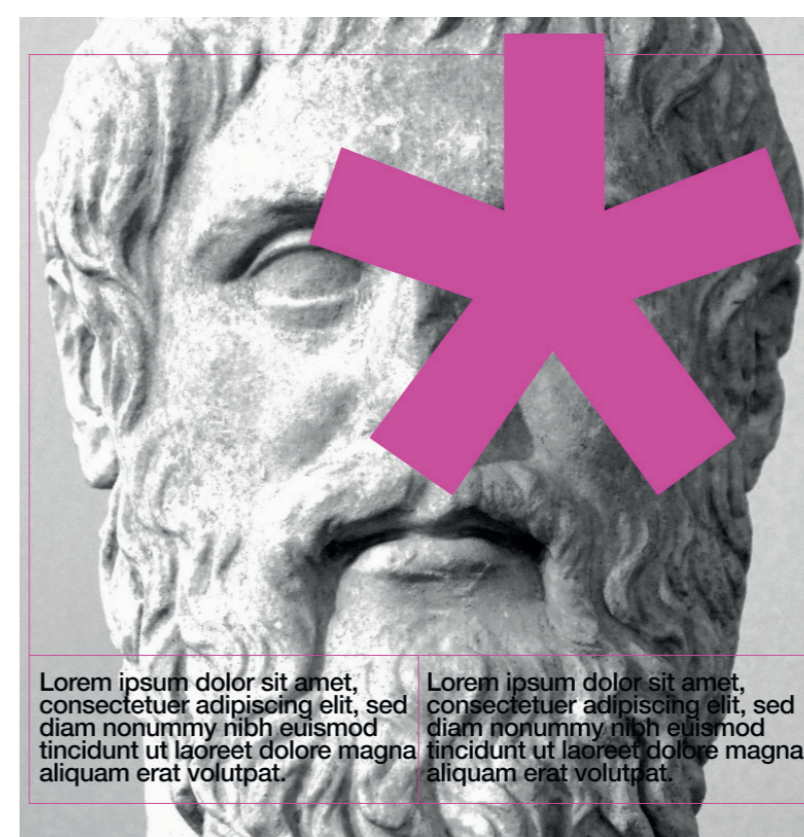
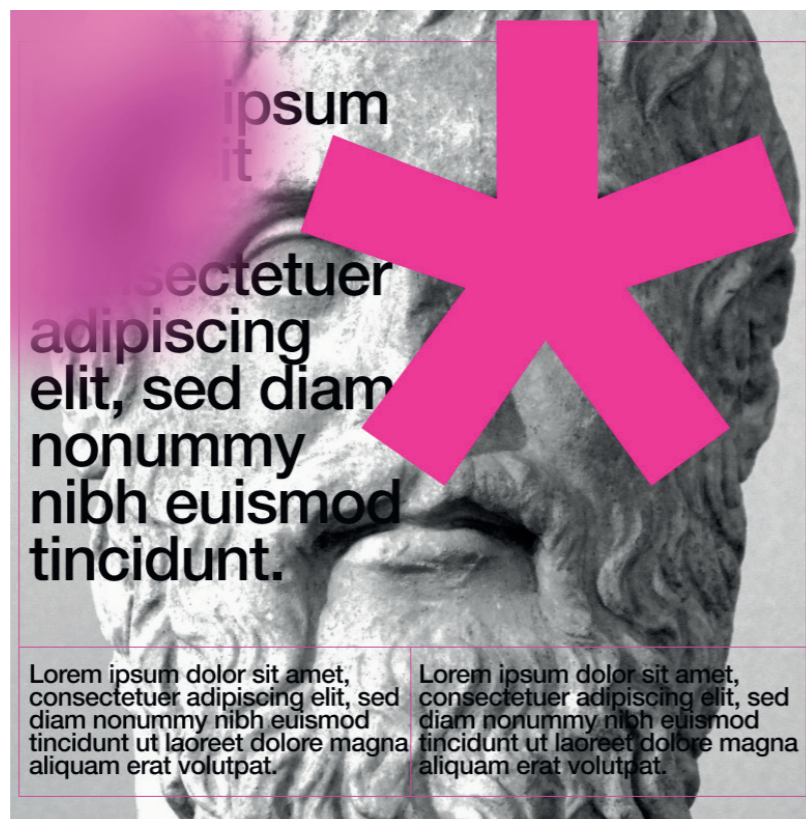


Fig 69 - Resultados planeados estaticamente no Figma (600 x 800)



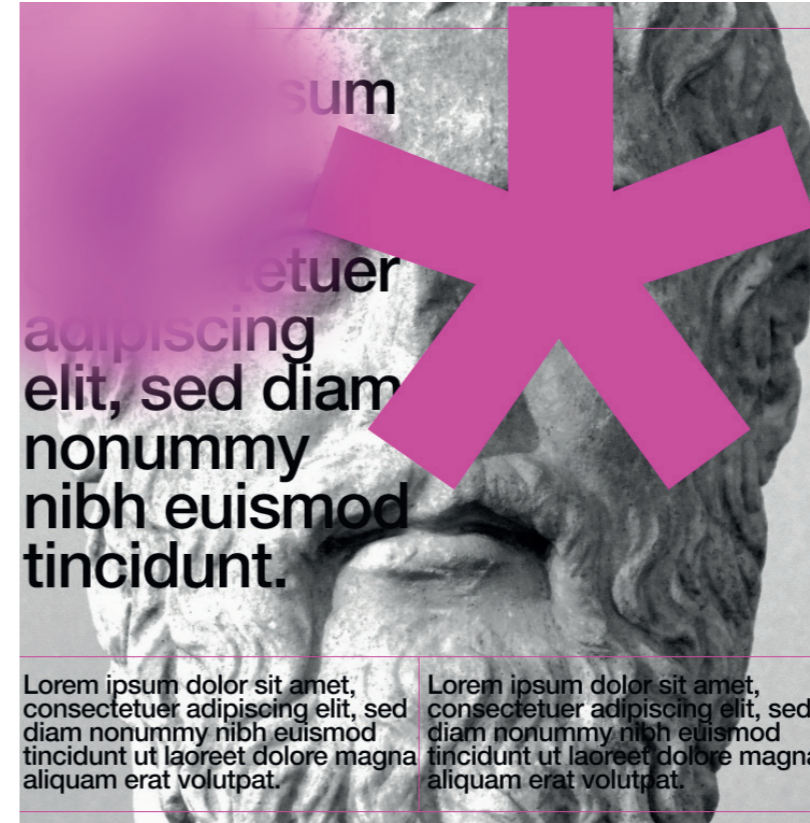
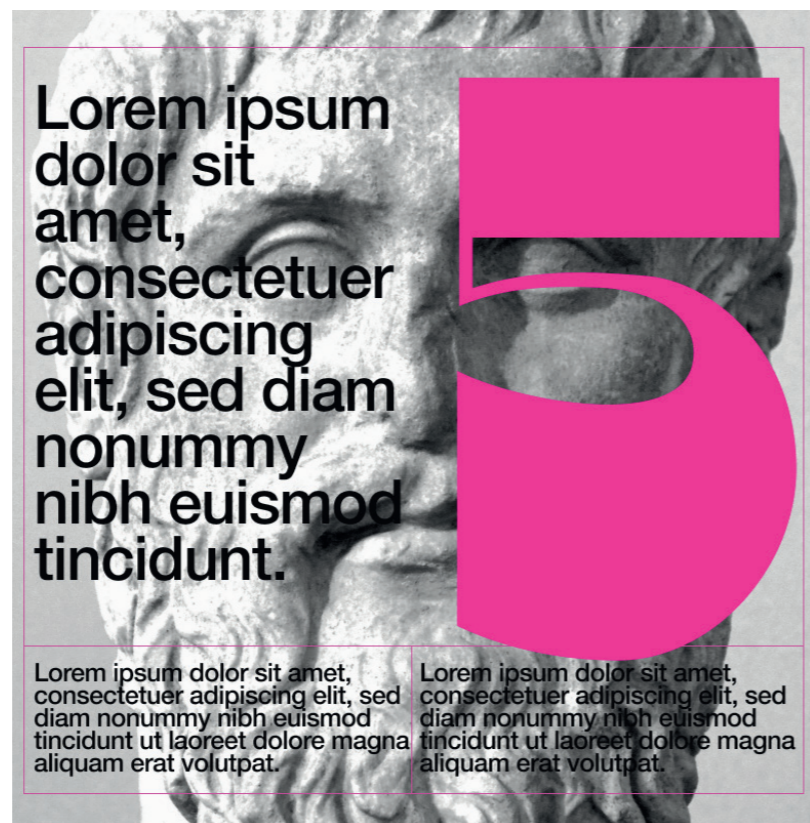


Fig 70 - Resultados planeados estaticamente no Figma (800 x 800)

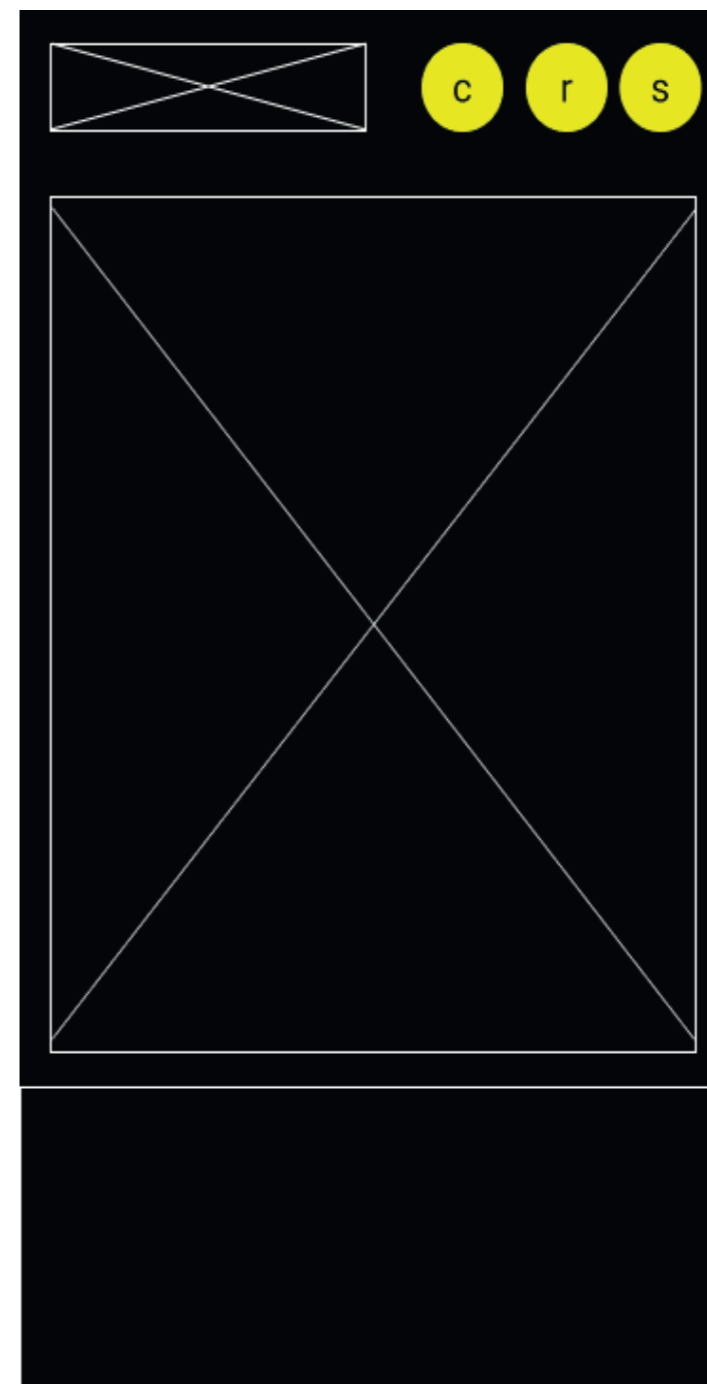


Fig 71 - Resultados planejados estaticamente no Figma da versão mobile

#### (4.3.4) Solução

A solução final resultou numa ferramenta com 4 divisões, imagens, texto, animações e experiências. Numa primeira instância entramos numa página onde não estamos em nenhuma secção, mas podemos alterar o tamanho da nossa artboard para 600x600 e 800x600 (esta medida manteve-se desde o início da proposta). Temos também ao nosso dispor os valores das cores, o canvas colour que representa a cor do fundo da artboard, que por default se encontra a branco, a fill colour que representa a cor de elementos como o texto, que por default se encontra a preto, e a stroke colour, que constituiu a cor do stroke do texto, que por default se encontra a branco.

A nível de cores, as cores escolhidas foram o branco (#FFFFFF), o preto (#030303), o amarelo (#FFFF00), o azul (#2525FF) e o rosa (#FF48B9). O branco e o preto foram escolhidos por serem as cores predominantes em interfaces. O azul por ter sido das primeiras cores a serem colocadas em webpages, principalmente no primeiro website, a World Wide Web, para representar as hiperligações. Por sua vez, o amarelo, foi uma cor selecionada por ser uma cor presente no website Space Jam, o primeiro website com animações em 1996. O rosa foi uma cor predominante no primeiro website www, a conter imagens. Este website pertence às Les Horribles Cernettes, uma banda feminina suíça que nasceu em 1990. Selecionando a secção das imagens é possível colocar uma imagem e alterar o seu tamanho para os seguintes parâmetros - small, medium, big e fullscreen. Cada um destes tamanhos tem um intervalo de tamanho pré-estabelecido, que quando se carrega no botão o mesmo vai alterando o tamanho. Houve a necessidade de colocar um aviso inicial “every time you press the same button the values change”, para alertar o utilizador dos valores variáveis/random que cada botão contém. Ainda nesta secção temos a possibilidade de alterar a posição da imagem, com a mesma lógica do tamanho, em que sempre que o mesmo é clicado altera a posição da mesma. Por fim, dispomos duma quantidade de filtros que podemos aplicar, desde colocar um tint, inverter as cores da mesma, colocar a preto e branco, aplicar um blur e posterizar.

Passando para a secção de texto, aqui dispomos de três hierarquias, Headline, Paragraph e Caption. Em qualquer uma delas temos a liberdade de estabelecer qual o conteúdo textual que podemos colocar, bem como selecionar o seu tamanho. À semelhança das imagens o botão do tamanho, apesar de só conter um parâmetro, e não diferentes níveis como na secção anterior, sempre que é clicado altera o seu valor, dentro dum espectro de parâmetros definidos. Os valores do Headline, que se entende como o tamanho maior, nunca chegam a ser os do parágrafo, e assim sucessivamente. Também nos é permitido alterar a posição do mesmo, funcionando este botão da mesma forma que os das imagens, sempre que carregamos nele os valores de x e y (sistema de coordenadas) é alterado. Para além disto, dispomos também da possibilidade de alterar o alinhamento do texto, para um alinhamento à esquerda, direita e ao centro. Aqui também dispomos dum botão para alterar a tipografia, sendo que sempre que carregamos nele, a tipografia altera. Esta alteração acontece nos três níveis de texto, Headline, Paragraph e Caption. Estas tipografias fazem parte dum bundle de fontes da Pangram Pangram Foundry, estúdio de tipografia criado em 2015 pelo design Mathieu Desjardins, disponibilizados pelos mesmos.

Passando para a secção ANIM, animações, conseguimos animar todos os elementos que estabelecemos anteriormente. A primeira animação Wavy, possibilita animarmos o texto sobre a forma de uma onda, sendo que aqui temos a opção Random e Mouse. Na Random a onda adquire valores aleatórios e uma posição aleatória tendo como base a posição onde o mesmo texto se encontrava. No botão de Mouse, através dos movimentos do Mouse conseguiríamos definir onde a mesma se encontra mediante o nosso movimento. Apesar de o tipo de animação ser igual, os valores da mesma diferem dando um aspeto de uma onda mais vertical, distinta da que encontramos no botão Random.

Na animação Cloudy, dispomos também das opções Random e Sound. Em qualquer uma das variantes temos a aparição de circunferências com um efeito de blur na página. Enquanto que no Random as mesmas vão aparecendo aleatoriamente, no botão de Sound as mesmas vão aparecendo mediante o som que captam do exterior, sendo que quanto maior for o volume e quanto mais ruído houver, mais circunferências vão aparecendo. Estas circunferências à medida que vão sendo desenhadas vão aplicando um efeito de blur sobre elas mesmas, fazendo com que os primeiros elementos estejam com um blur mais recente do que os últimos a serem colocados no ecrã.

Na animação Drawing, com a opção de Mouse e Camera, conseguimos desenhar no ecrã um efeito semelhante a desenhar com o lápis numa folha de papel, onde o lápis é o nosso mouse e o papel é o nosso ecrã/canvas. No botão de Mouse conseguimos desenhar como referido anteriormente através do nosso mouse. No botão Camera é pedida autorização ao utilizador para estabelecer ligação com a câmara do seu dispositivo e o desenho é formado através do movimento do nosso nariz. Aqui foi necessário ligar a câmara para estabilizar a forma da face e reconhecimento do nariz, onde o movimento dele vai sendo guardado sobre a forma de ponto, com a coordenada x e y, e estabelecida a ligação dos mesmos sobre a forma de uma linha, formando o nosso desenho.

Na animação Line Growth temos a opção de Random. Esta animação simula o crescimento, progressiva e aleatoriamente, de uma linha no ecrã. No botão de Bubbling temos apenas a opção Random. Aqui tudo o que é texto assume uma aparência de uma bolha, cada ponto que a tipografia assumir é convertido para uma circunferência. Estas circunferências podem ser mais ou menos numerosas mediante o afastamento ou aproximação do cursor. Spinning é uma animação que faz alterações apenas na Headline. Na posição de Random o texto assume uma rotação e dispõe-se sobre uma circunferência, rodando constantemente. Na posição de Mouse o texto assume uma perspectiva de espiral sobre todo o canvas, aumentando ou diminuindo o seu centro e a complexidade mediante a posição de x e y do cursor.

Nesta secção temos ainda uma animação de Ping Pong, que é executada em Mobile. Aqui os elementos andam dum lado para o outro do ecrã mediante o ângulo do telemóvel paralelamente ao chão. Por último, existe uma secção intitulada de Experiments, que resulta num conjunto de experiências que serão implementadas futuramente na secção de Anim. Estas animações não podem ser alteradas pelo utilizador, resultam apenas de resultados visuais estáticos, para dar uma previsão de possíveis desenvolvimentos da ferramenta ao utilizador.

Quando terminada a composição, a mesma pode ser apagada, iniciando o processo de criação do início. Nesta fase, tudo o que foi desenhado é apagado, voltando o canvas a ficar a branco, existindo, também, a possibilidade de guardar uma imagem e de a gravar.

Toda esta ferramenta, que foi pensada para desktop, foi também transferida para um contexto de mobile, dada a pertinência e uso constante deste dispositivo no nosso dia-a-dia e a facilidade que nos propõe de acesso a qualquer tipo de website.

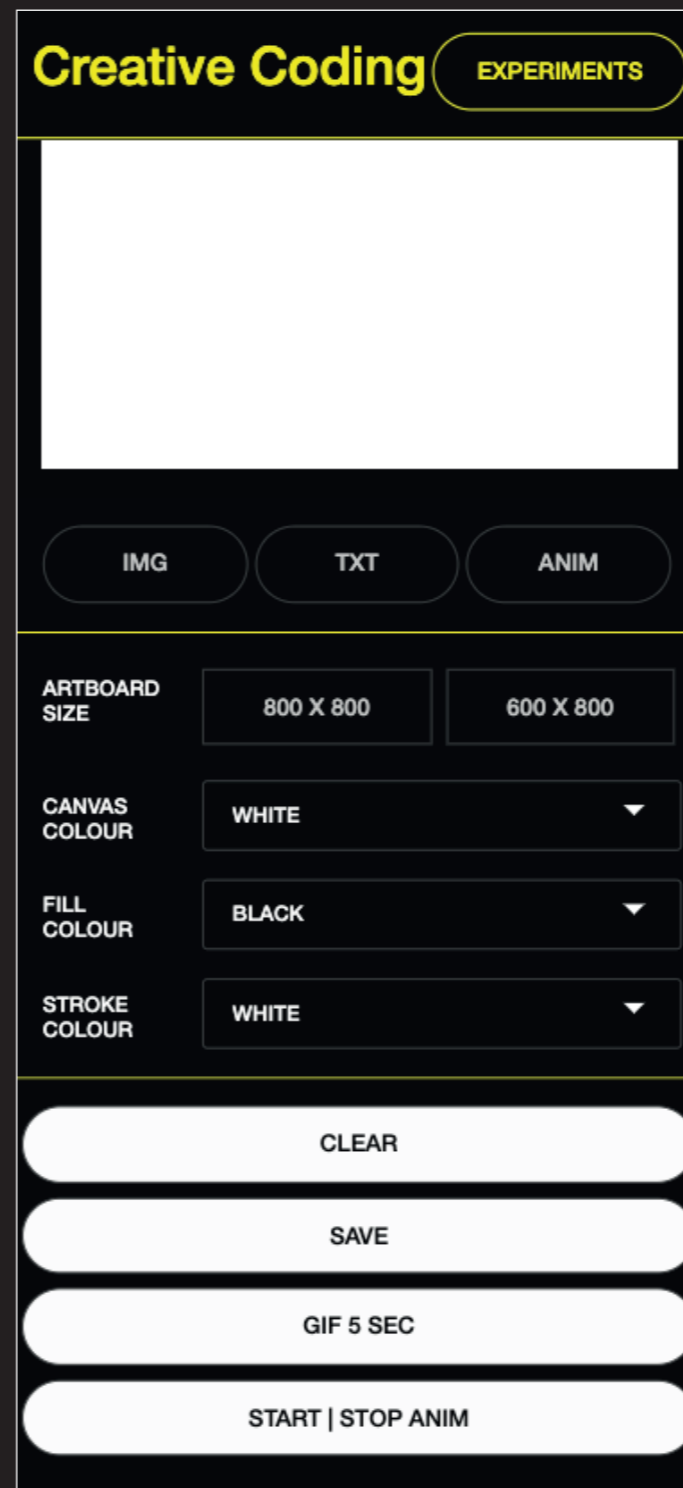


Fig 72 - Versão Final da Ferramenta, secção de abertura (mobile)

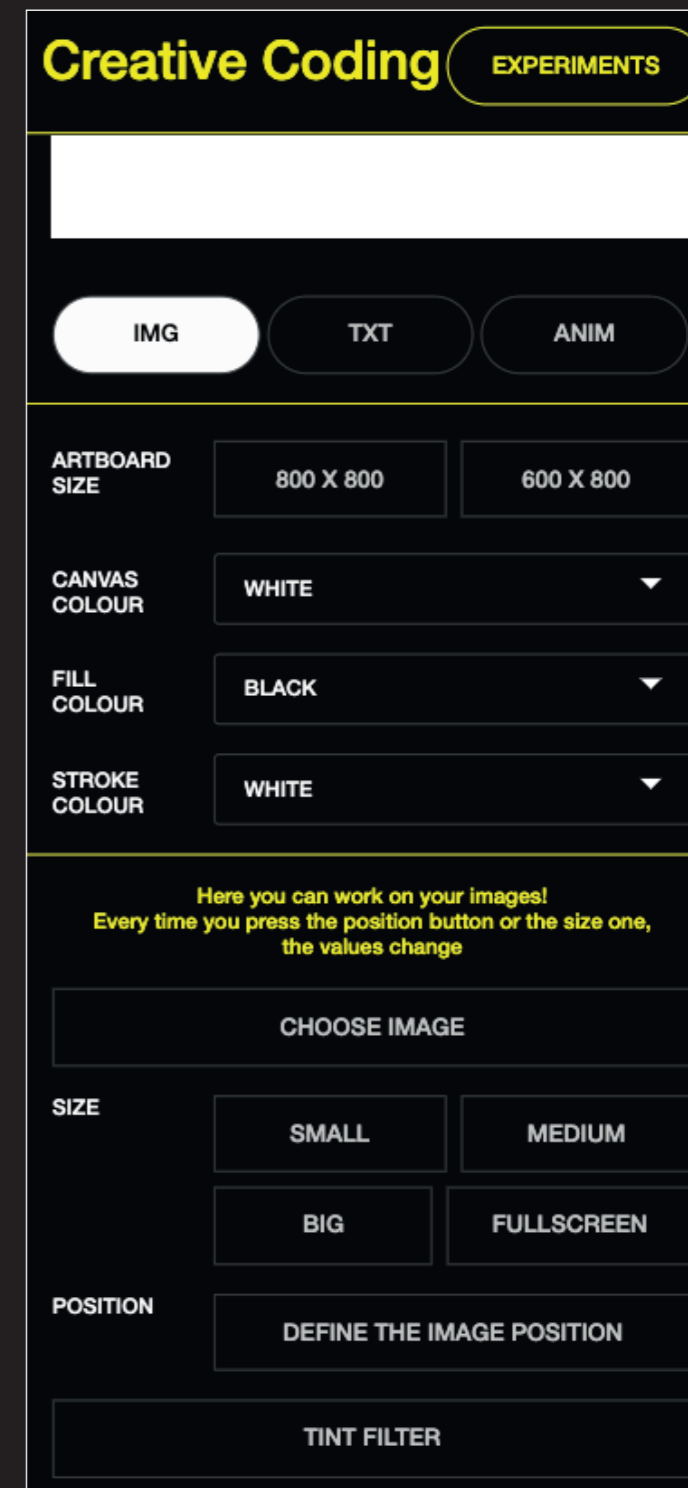


Fig 73 - Versão Final da Ferramenta, secção das imagens (mobile)

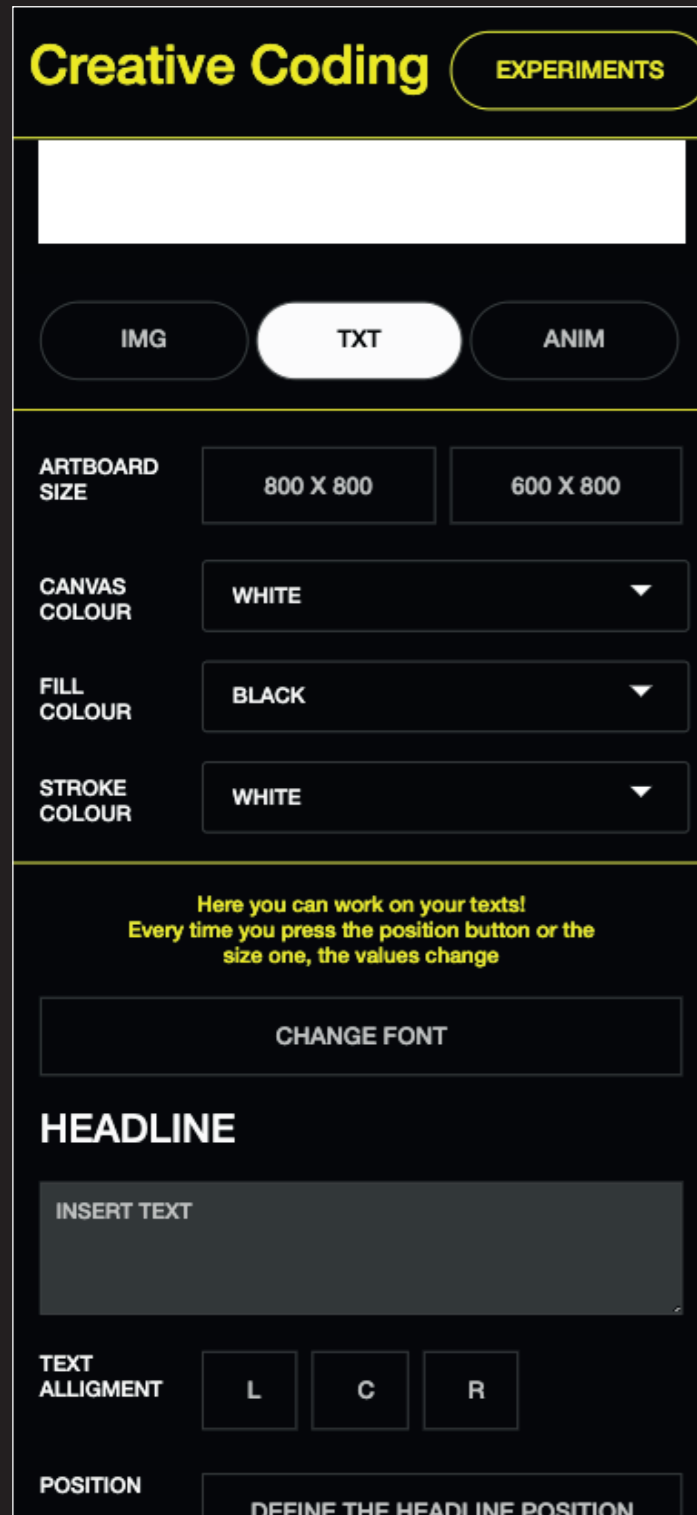


Fig 74 - Versão Final da Ferramenta, secção do texto (mobile)

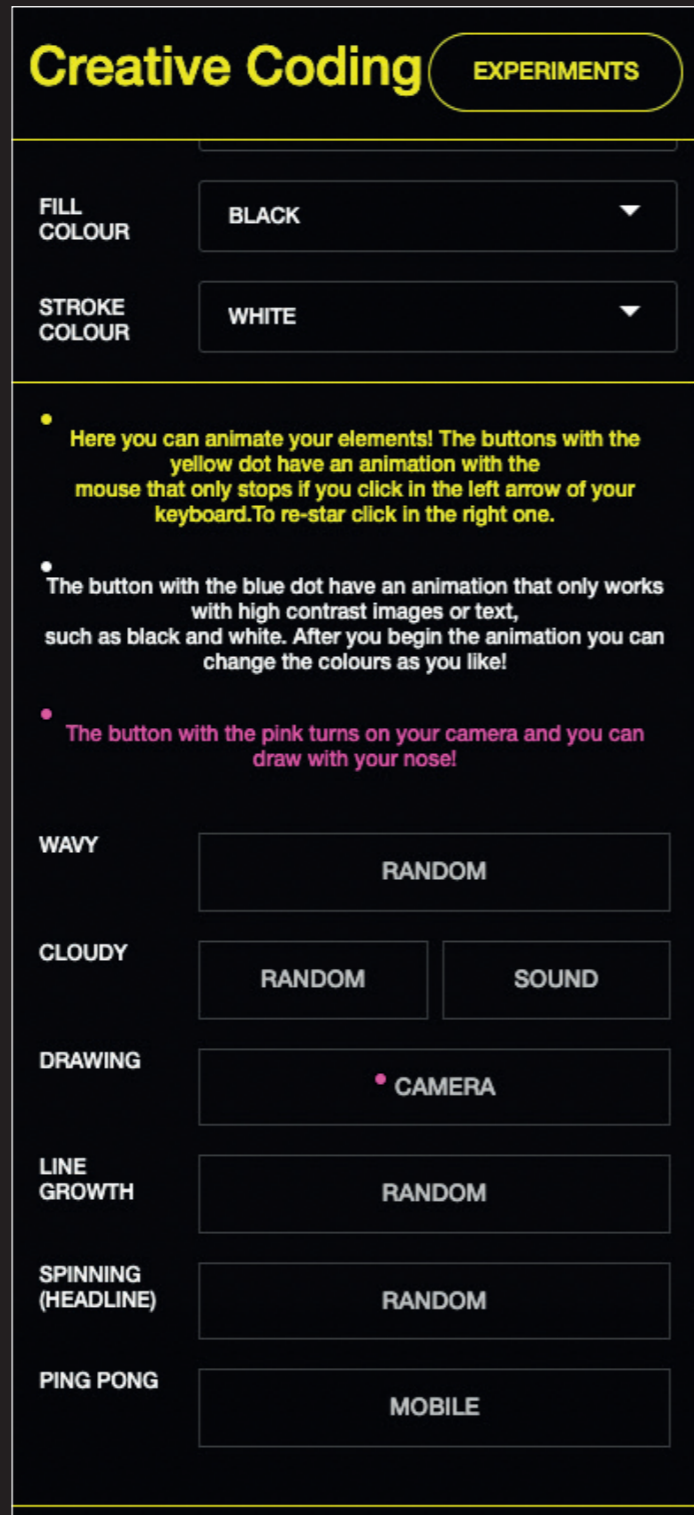


Fig 75 - Versão Final da Ferramenta, secção das animações (mobile)

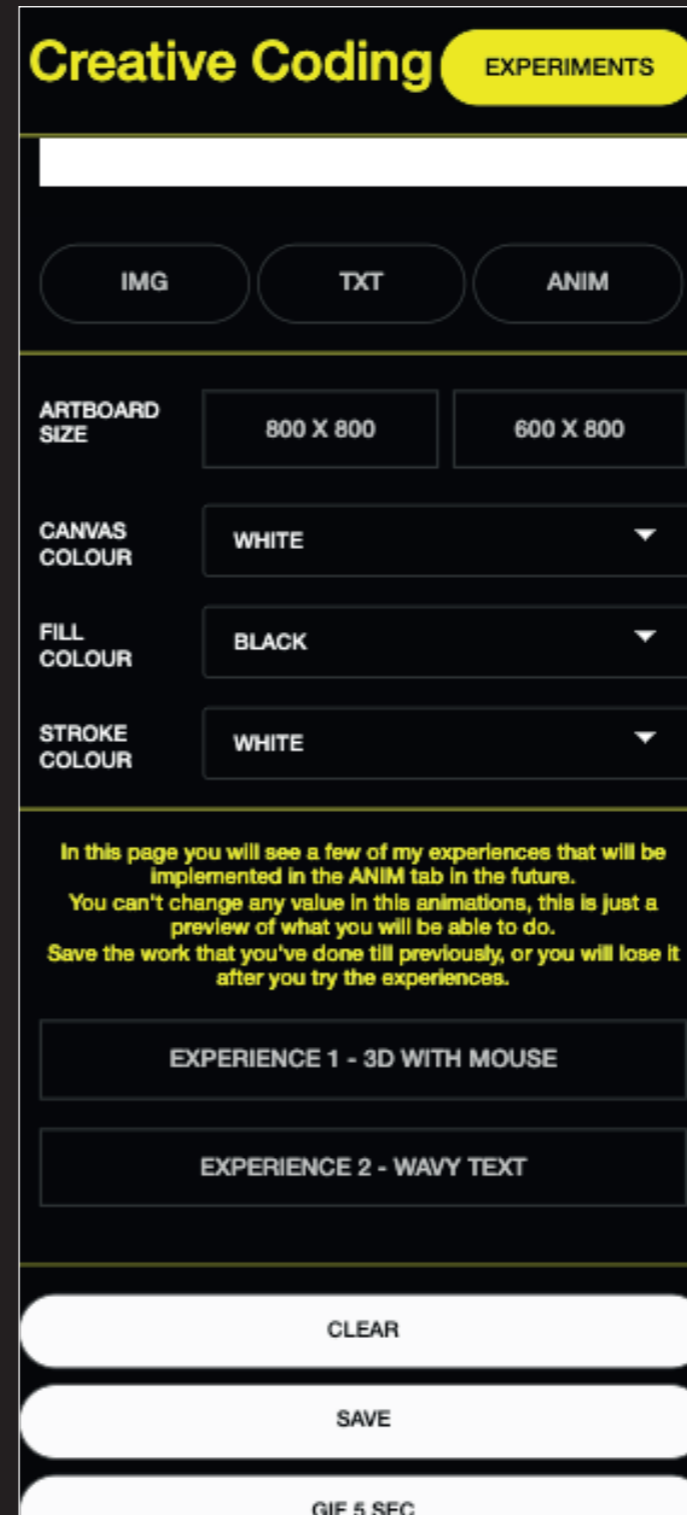


Fig 76 - Versão Final da Ferramenta, secção das experiências (mobile)

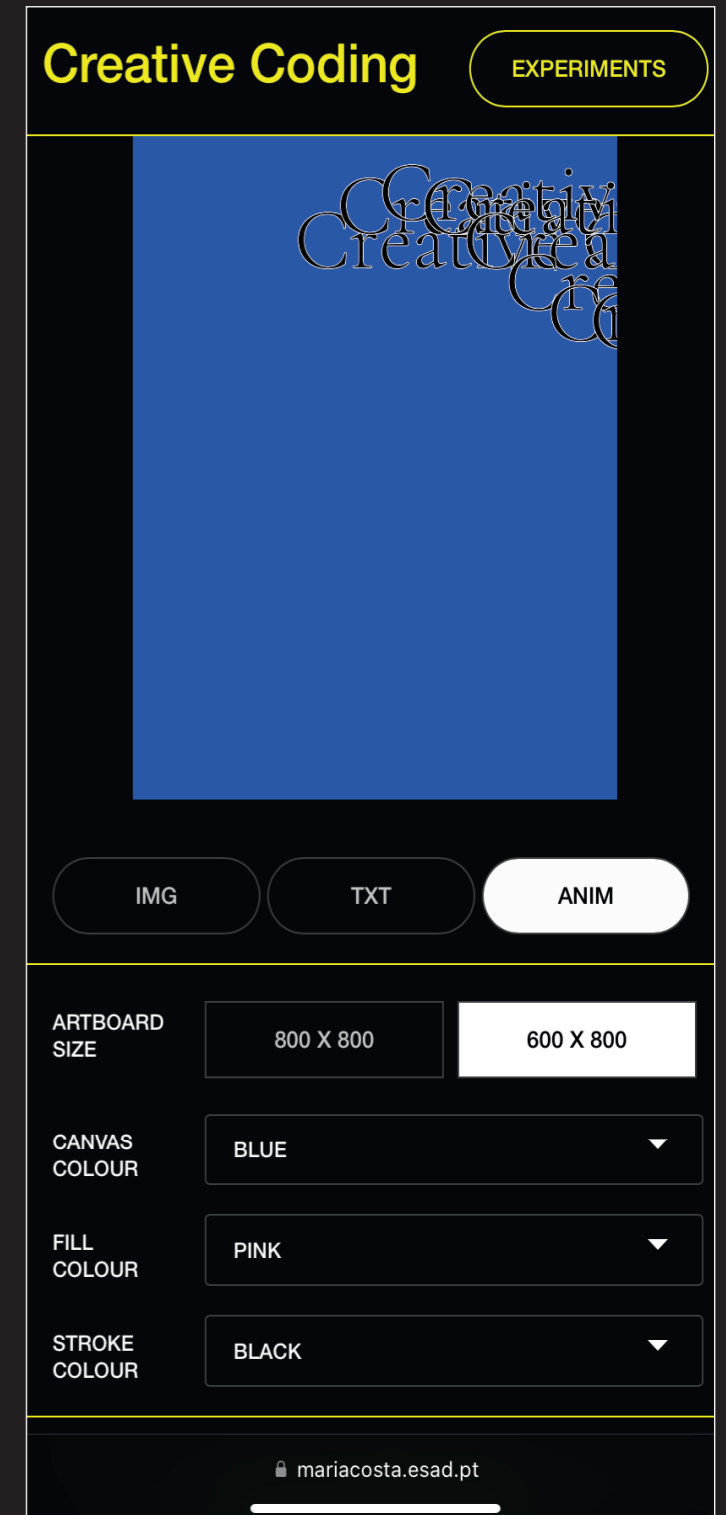


Fig 77 - Versão Final da Ferramenta, canvas 600x800 com texto aplicado (mobile)





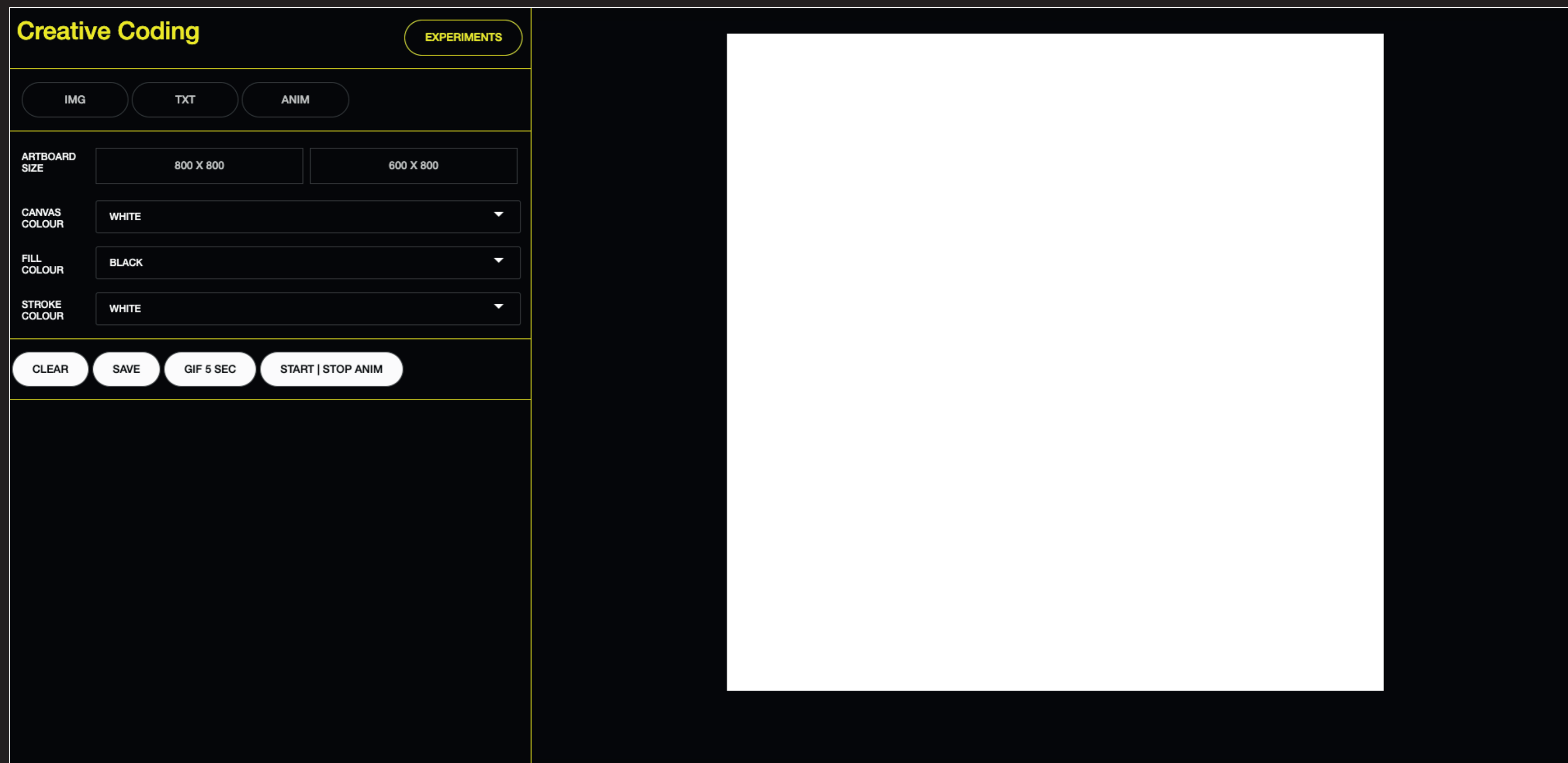


Fig 78 - Versão Final da Ferramenta , secção de abertura (desktop)

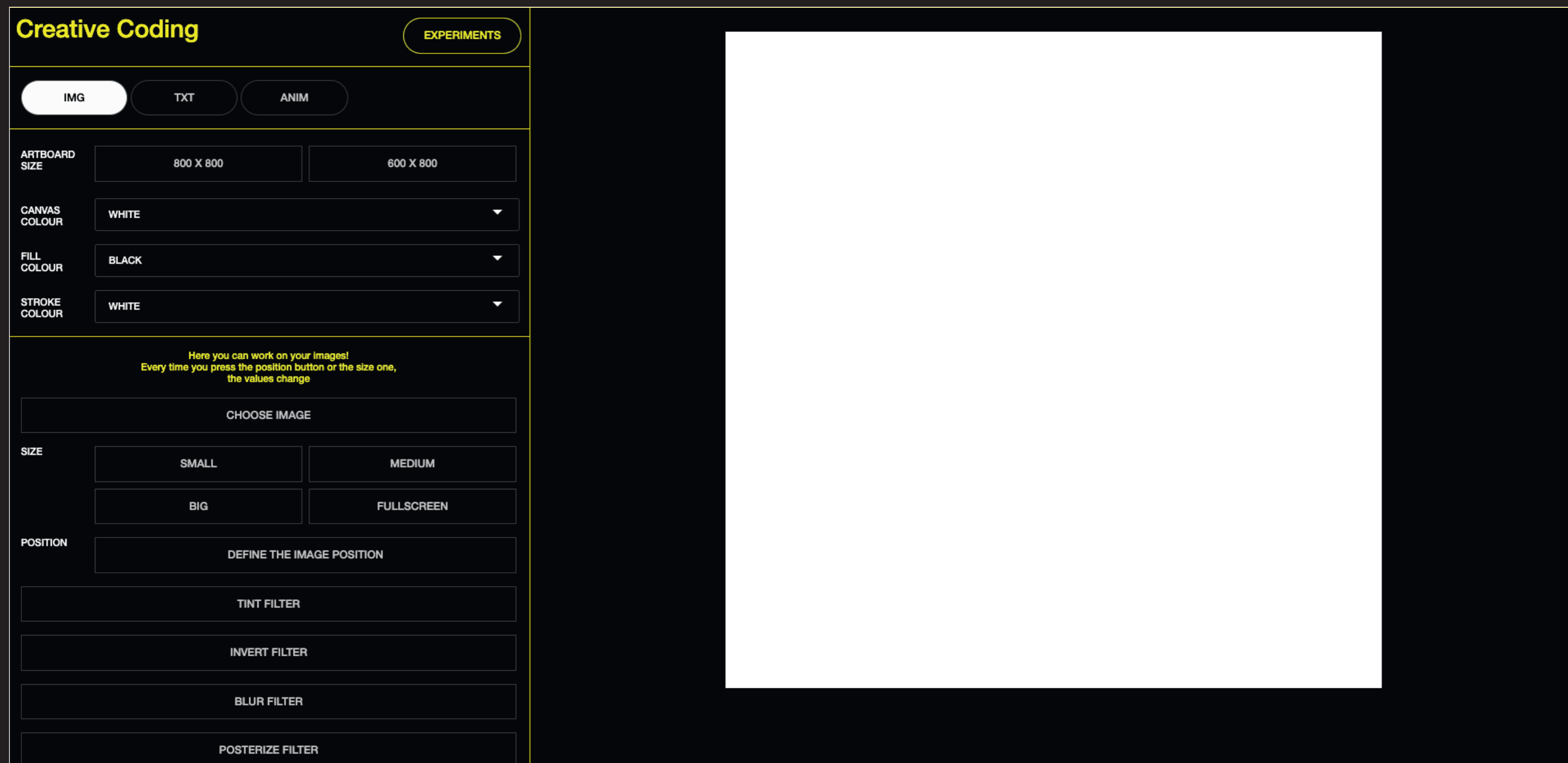


Fig 79 - Versão Final da Ferramenta, secção das imagens (desktop)

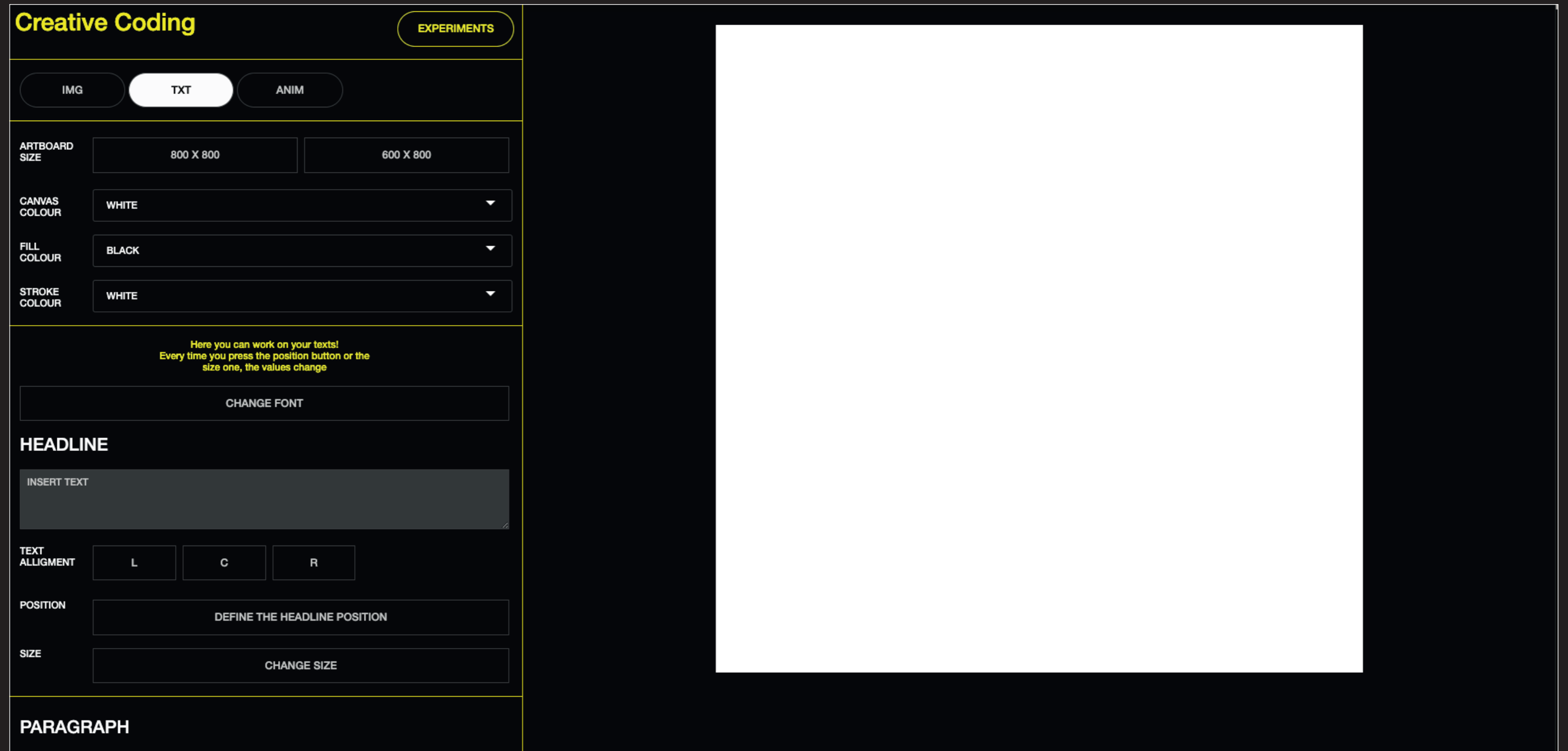


Fig 80 - Versão Final da Ferramenta, secção de texto (desktop)

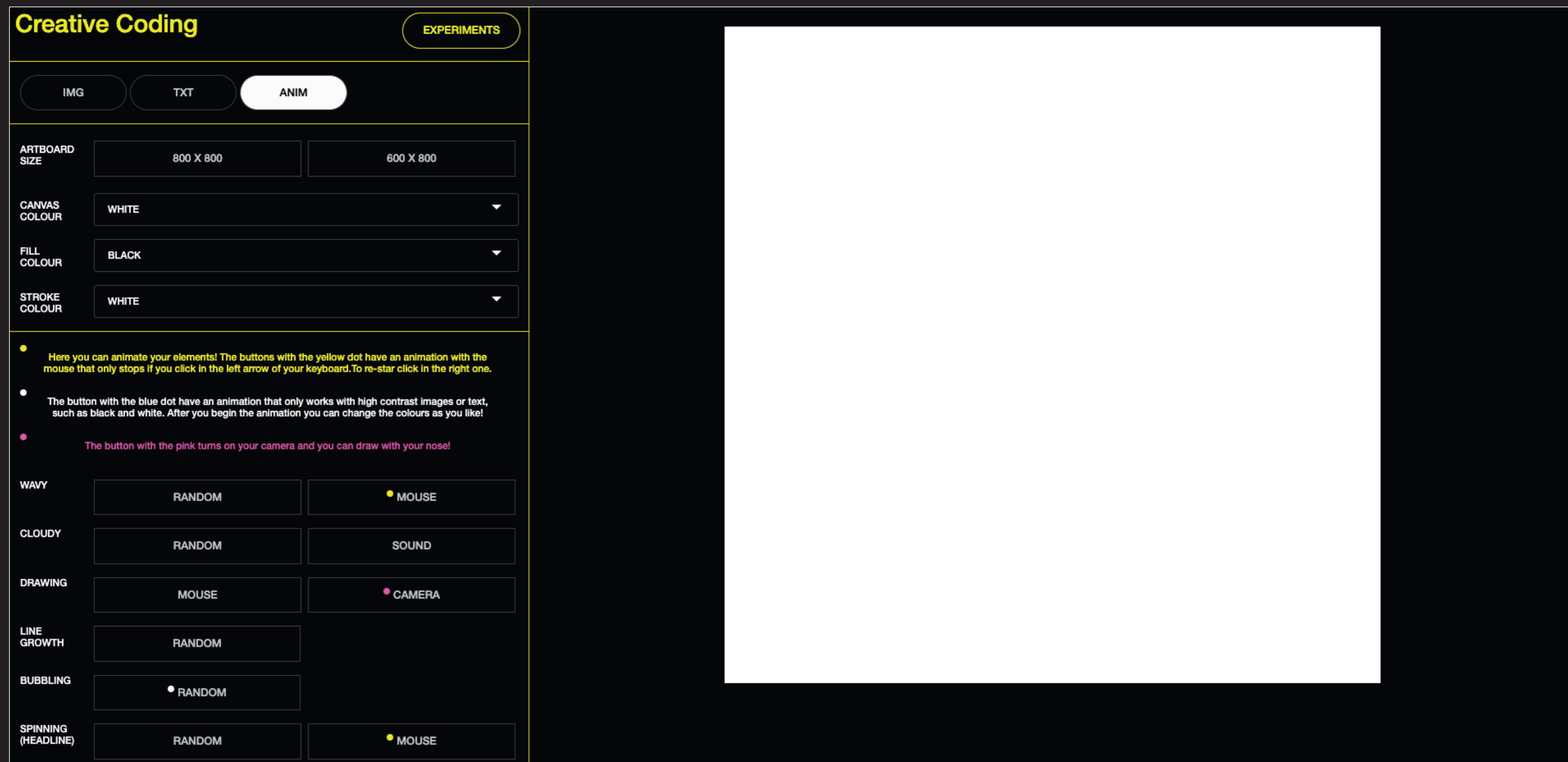


Fig 81 - Versão Final da Ferramenta, secção das animações (desktop)

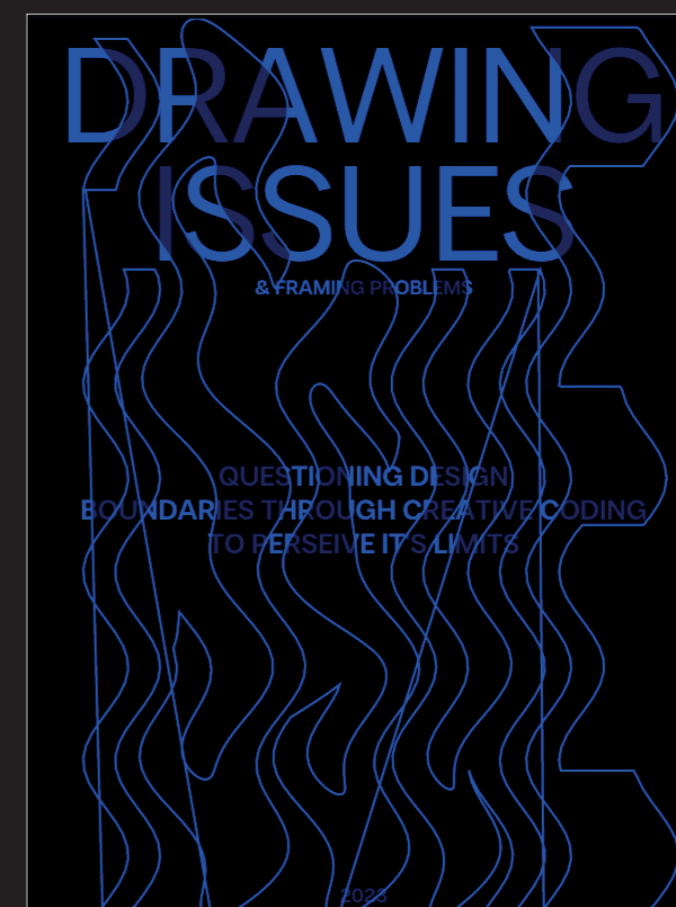
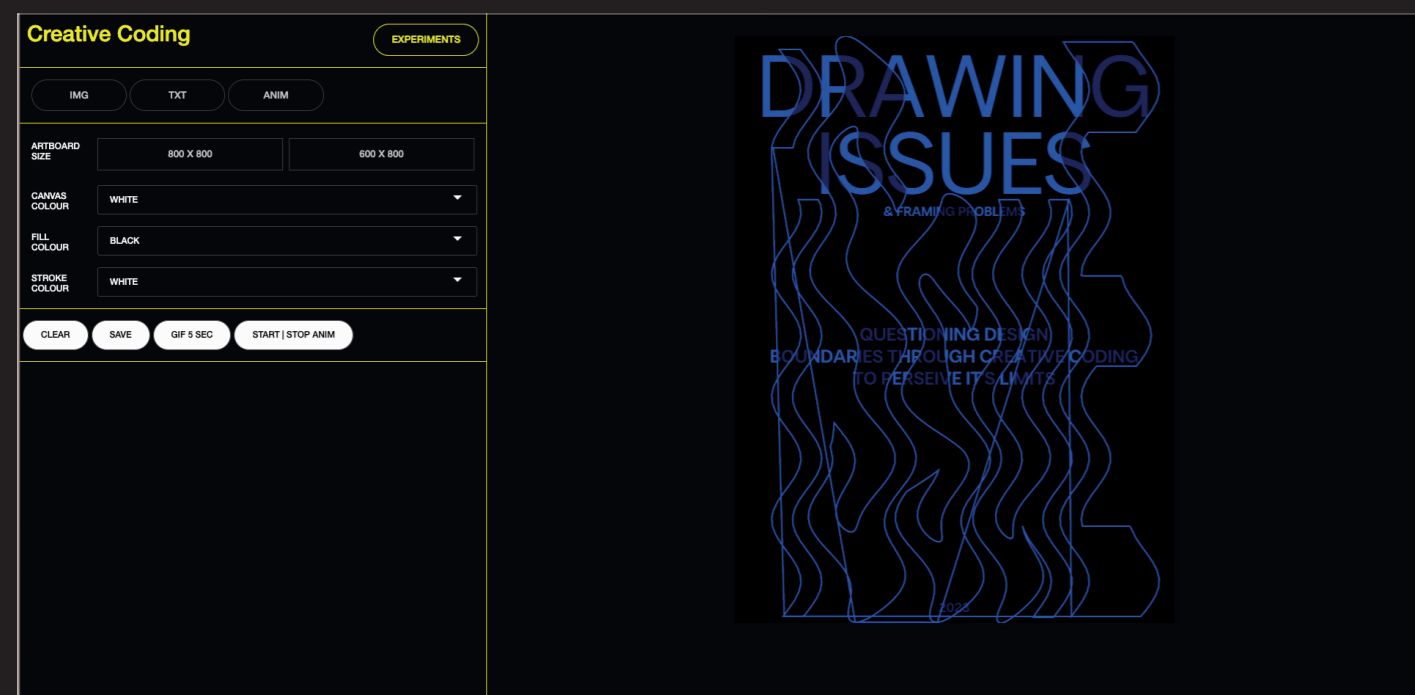
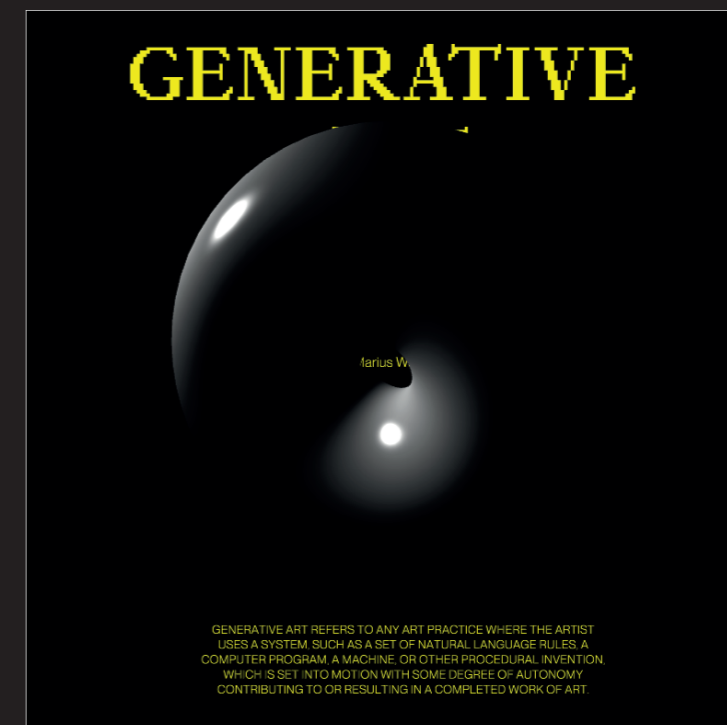
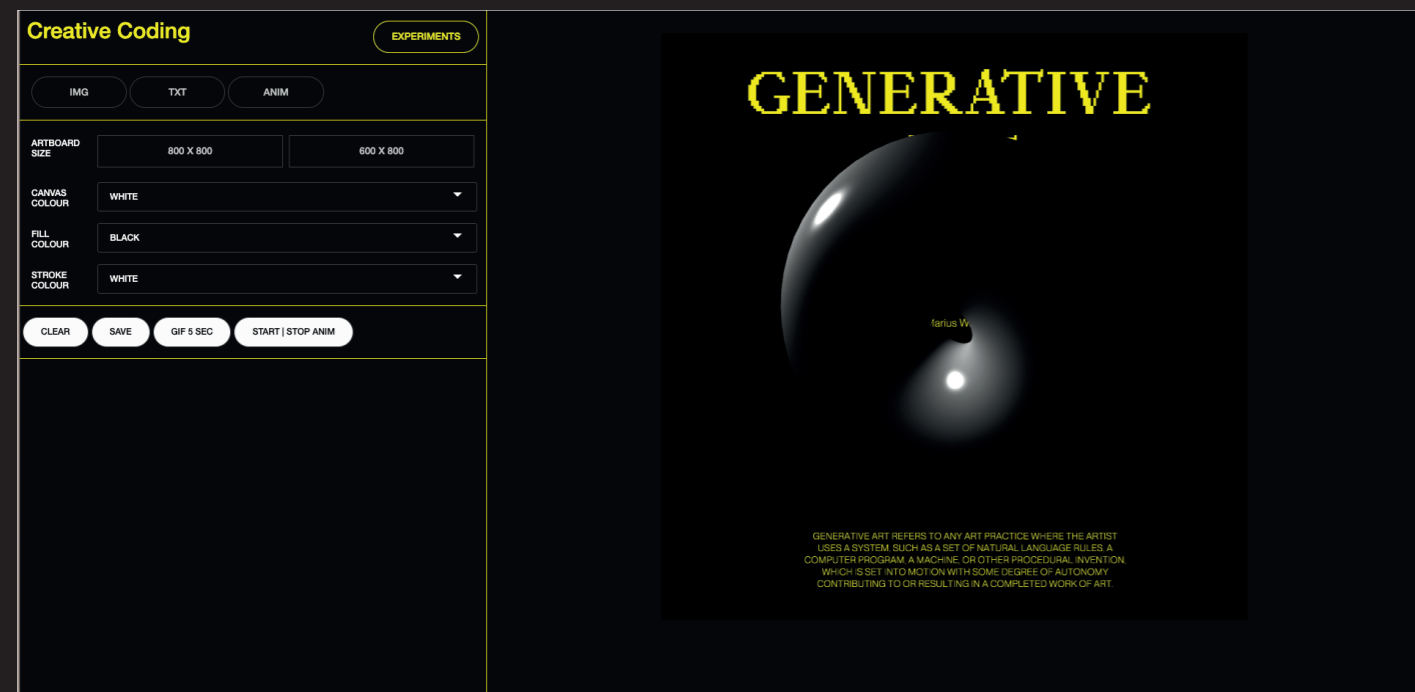


Fig 82 - Versão Final da Ferramenta, secção das experiências com duas das propostas implementadas (desktop)

#### (4.3.5) Resultados

Para perceber a viabilidade da ferramenta foram pedidos a alguns utilizadores para testar a mesma produzindo os resultados que desejassem, sem quaisquer tipo de restrições, de forma a promover a experimentação da ferramenta. Nas páginas seguintes encontram-se alguns exemplos dos resultados obtidos pelos participantes.



Fig 84 - Resultado (800x800) por Ana Moreira

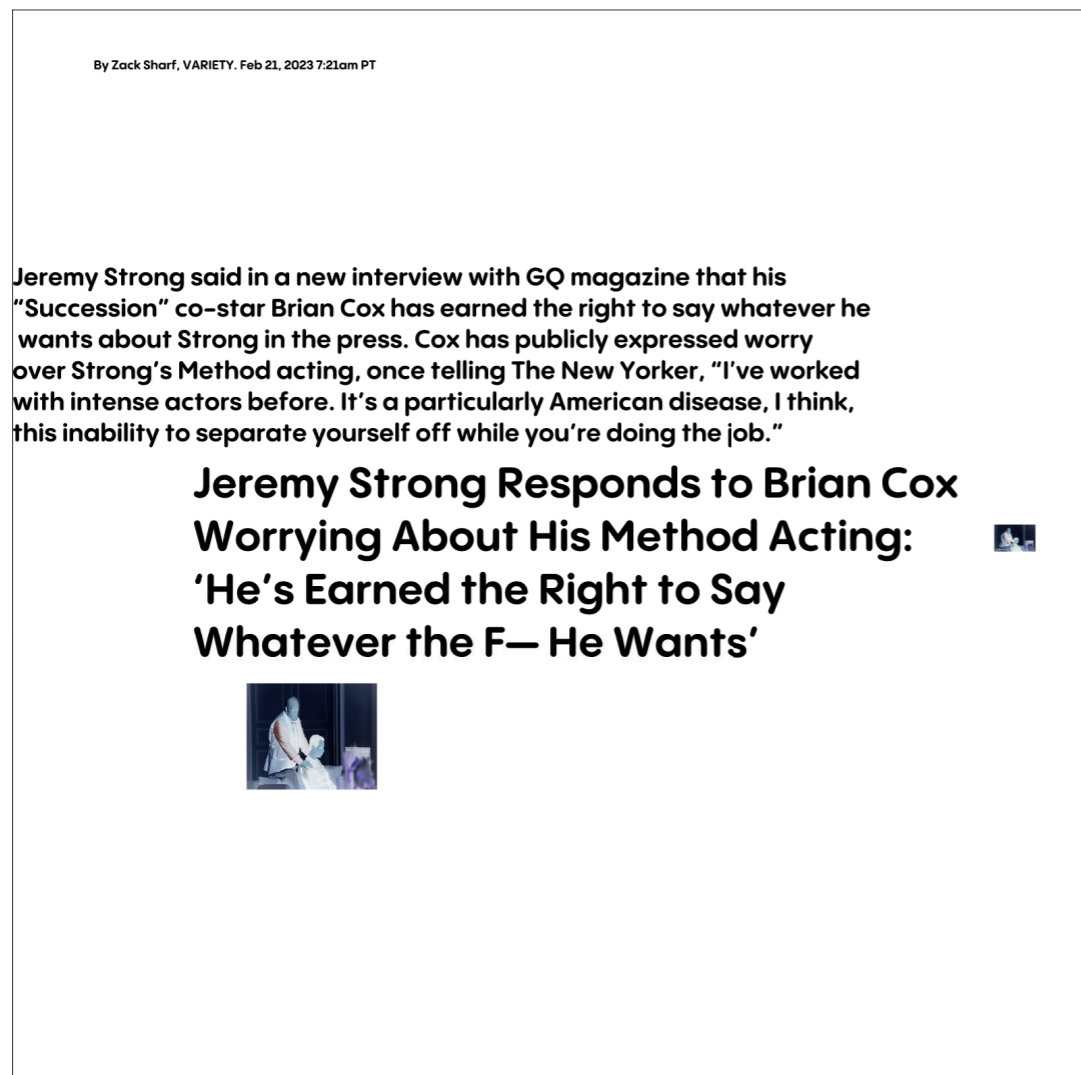


Fig 85 - Resultado (800x800) por Ana Moreira



Fig 86 - Resultado (800x800) por Ana Moreira



Fig 87 - Resultado (800x800) por Ana Moreira

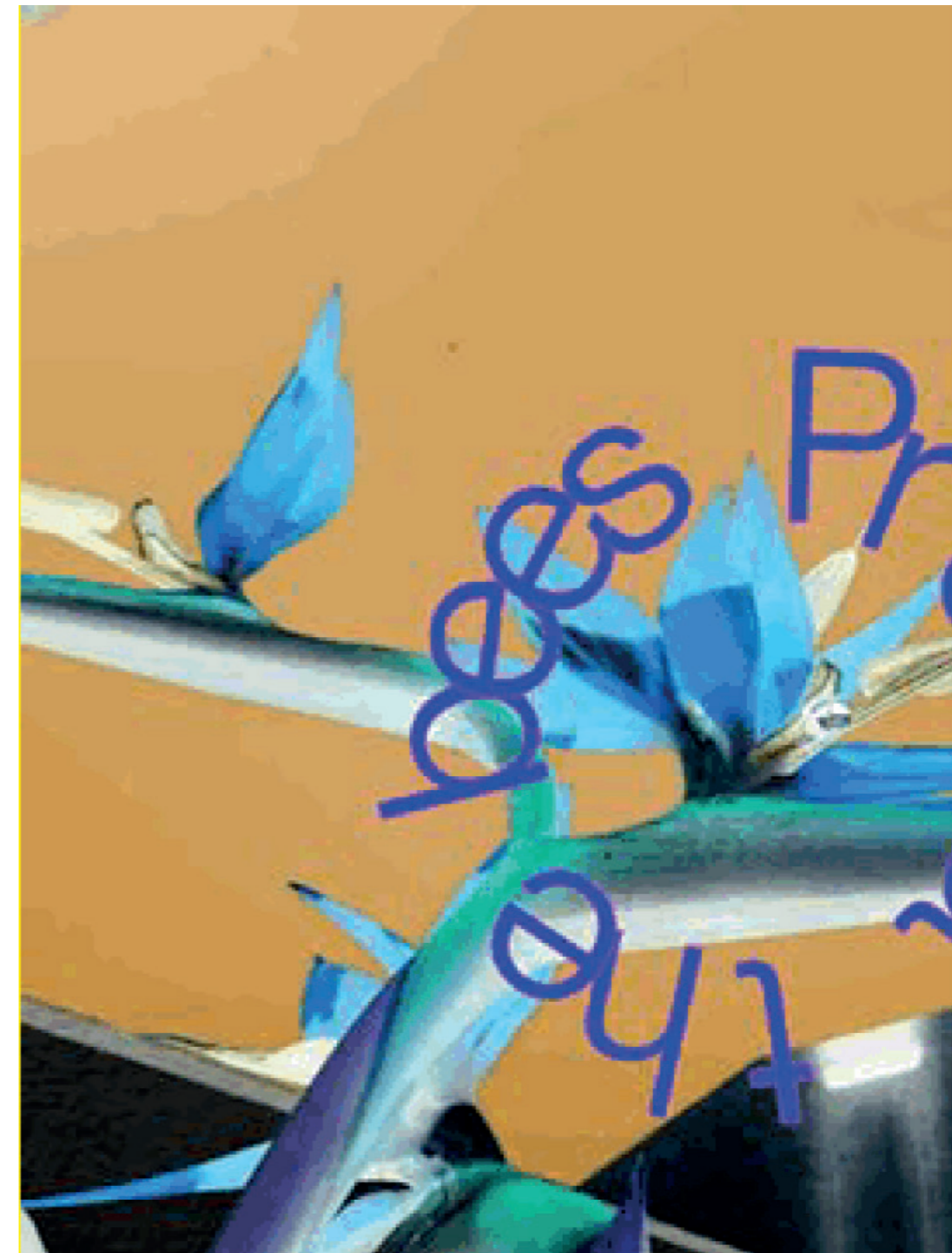


Fig 88 - Resultado (800x600) por Clara Costa



Fig 89 - Resultado (800x800) por Francisco Morais Soares



Fig 90 - Resultado (800x800) por Francisco Morais Soares



Fig 91 - Resultado (800x800) por Francisco Morais Soares

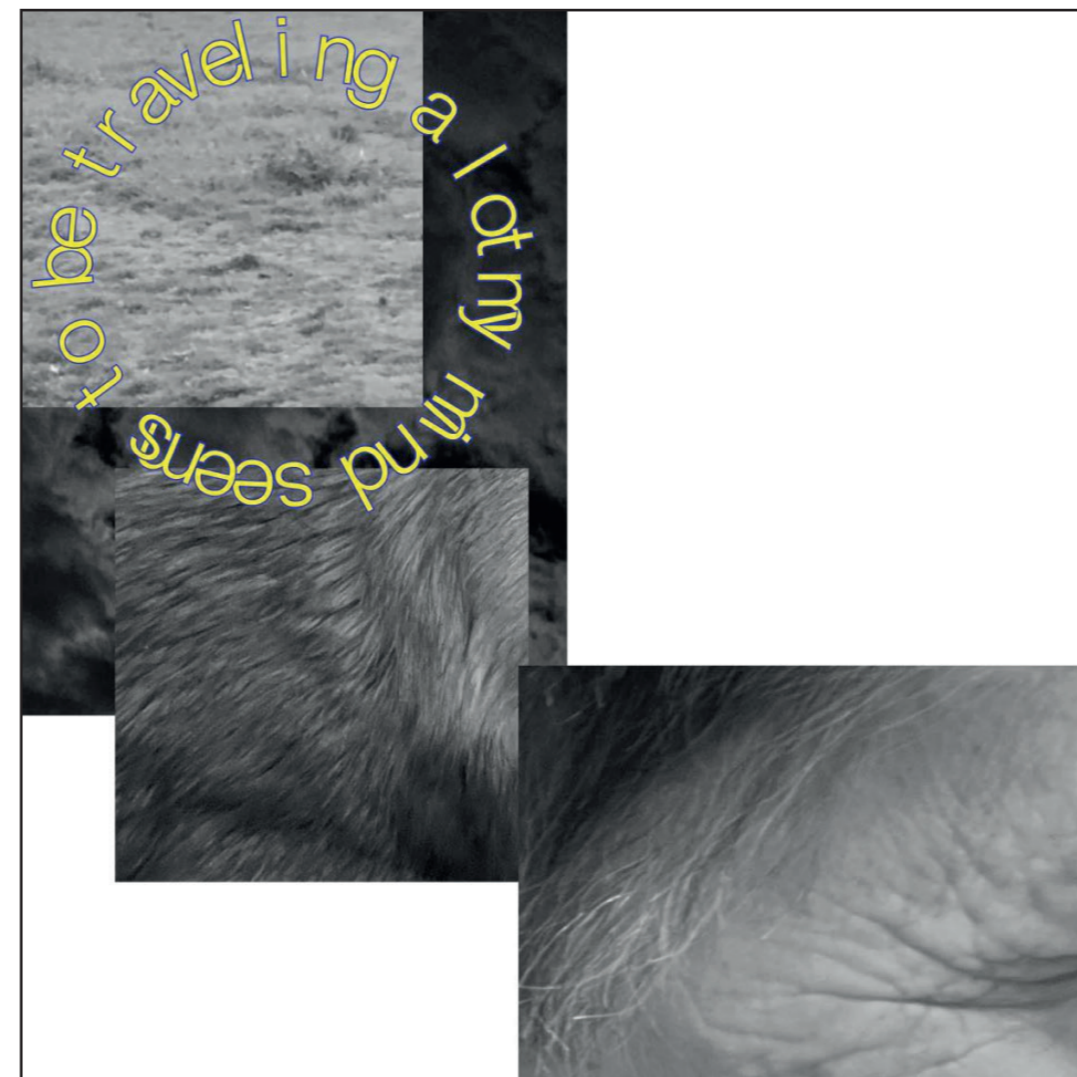


Fig 92 - Resultado (800x800) por Jéssica Teixeira



Fig 93 - Resultado (800x800) por Jéssica Teixeira

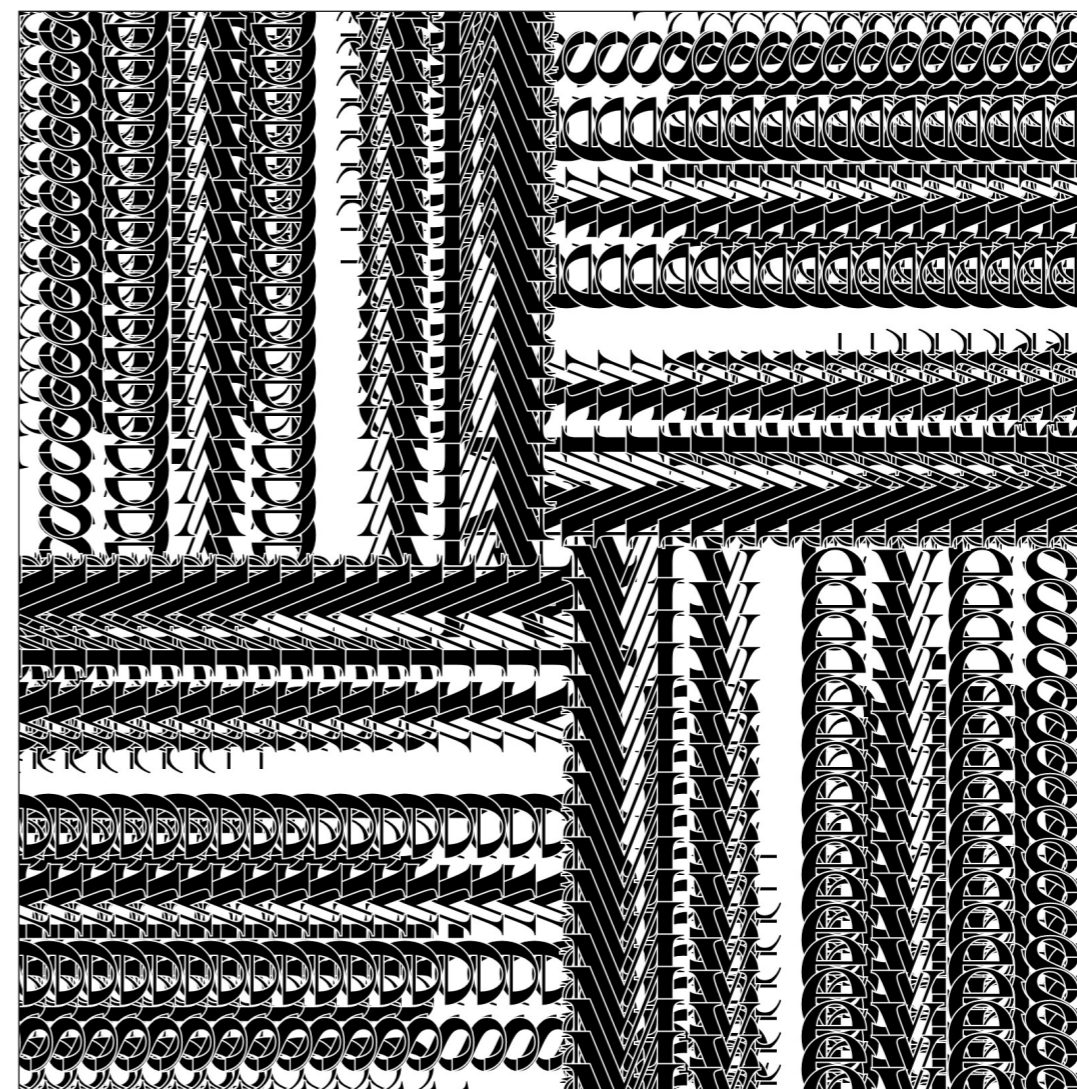


Fig 94 - Resultado (800x800) por Mariana Jerónimo



Fig 95 - Resultado (800x600) por Tim Rodenbröker

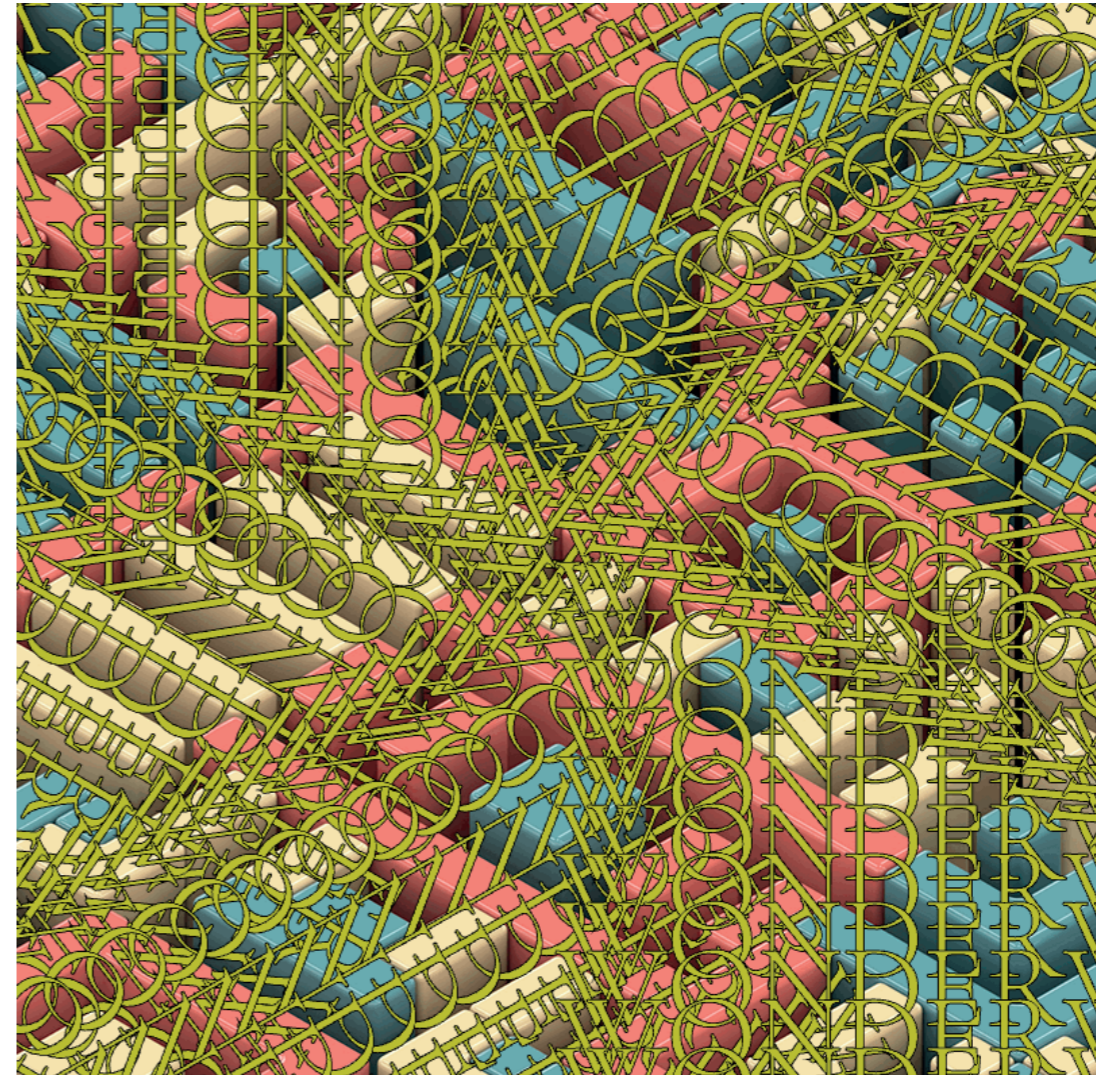


Fig 96 - Resultado (800x800) por Pedro Fonseca

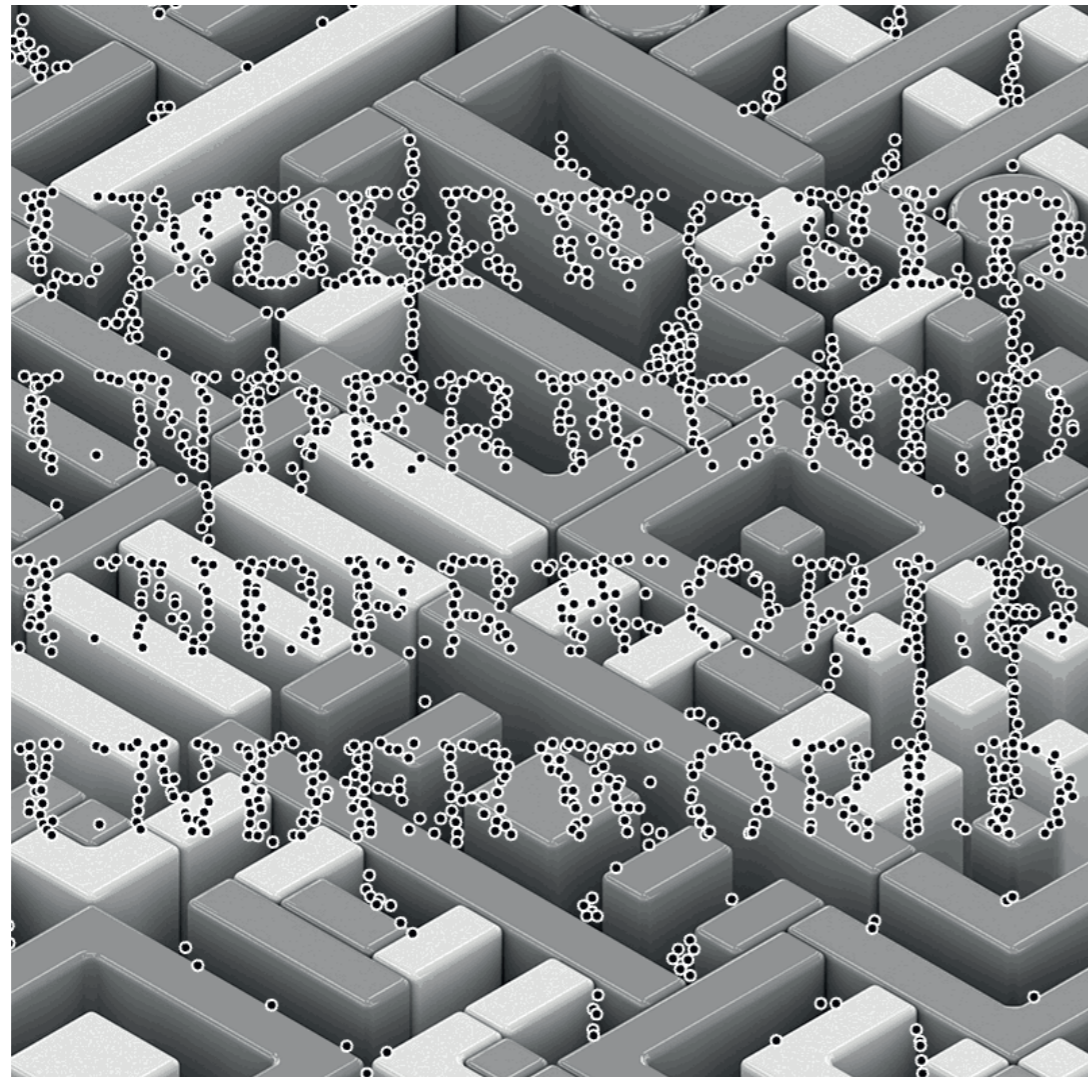


Fig 97 - Resultado (800x800) por Pedro Fonseca



Fig 98 - Resultado (800x600) por Margarida de Matos



Fig 99 - Resultado (800x600) por Margarida de Matos

# (5) CONCLUSÃO

## (5.1) Conclusão

Sendo o creative coding um processo interativo baseado na experimentação, contém um método de design não intencional, que o distingue do design generativo, onde o resultado final é parcialmente pré-definido e pensado, o que faz com que o creative coding utilize um método de design indutivo.

O raciocínio indutivo é o processo inverso do dedutivo, isto é, parte do específico para o geral, e embora não produza novos conhecimentos, procura induzir o conhecimento já existente a uma validação através de uma experimentação.

Está relacionado ao método empírico que significa obter conhecimento através dos cinco sentidos, que é a experimentação e a observação.

O design indutivo e não intencional é amplamente subestimado e o paradigma do design, enquanto ferramenta para resolver problemas existentes, é um enorme obstáculo para jovens designers que desejam aprender creative coding.

A dedução está relacionada ao pensamento analítico, também conhecido como pensamento convergente. É o pensamento que procura analisar várias informações para convergir em direção a um único resultado, do geral para o específico. Este raciocínio acredita que podemos prever o futuro com base no passado. Prevê que se conclui algo a partir de informações que já existem, portanto a dedução não produz conhecimentos novos. Já o raciocínio abduutivo atua entre os dois extremos anteriores, procura tanto confiança como validação. O raciocínio abduutivo procura a validação, assim como a indução, e procura a melhor explicação possível, tal como a dedução procura a verdade. O interessante é que o método abduutivo é o único raciocínio que produz criatividade e inovação, por ser a única lógica que introduz uma nova ideia. Com isso, o raciocínio abduutivo não resulta em verdades absolutas que são inquestionáveis, muito pelo contrário, procura novas ideias e conhecimentos que possam validar algo. Este método não prova que algo é de determinada forma, apenas diz que é mais provável que seja e procura a melhor explicação para isso.

Este método também é o único que projeta o futuro sem se prender ao passado, é o único procedimento racional de aquisição de conhecimento, enquanto que o indutivo e o dedutivo servem para verificar ou comprovar a verdade de um conhecimento já adquirido. (Alpa, D., 2022).

Podemos estabelecer uma analogia, entre os conceitos abordados em todo este projecto e os raciocínios aqui presentes, em como o creative coding procura um raciocínio abduutivo.

Cada vez mais, como designers, temos diversas ferramentas ao nosso dispor até pela evolução dos meios digitais. Com este projeto pretendeu-se deixar uma reflexão sobre o papel do designer e o papel da programação no nosso dia-a-dia e como ferramenta na nossa prática. Como Steve Jobs referiu, todos devíamos aprender a programar um computador, a aprender linguagem de programação, pois a mesma ensina-nos também como pensar. Jobs S., The Lost Interview (2012).

Este projeto pretendeu também explorar a potencialidade do conceito de design, que aceita a complexidade da tecnologia e da programação para seu proveito, de forma também a repensarmos o nosso papel enquanto profissionais. É importante distanciarmo-nos de aspetos visuais, e encararmos também o papel de criadores e pensadores, já que a programação nos ajuda em sistemas de pensamento.

Sem qualquer objetivo de produto final foi desenvolvida esta ferramenta que limita as escolhas do utilizador, aplicando conceitos de parametrização e randomização, de forma a explorar outro tipo de elementos e composições distintas, ao invés do dia-a-dia onde temos liberdade criativa total. Retirando diversas escolhas ao utilizador, o mesmo pode fazer uma interpretação diferente das opções que tem, produzindo soluções, não expectáveis, nem previsíveis.

O creative coding, ao invés do design gráfico, permite-nos desenvolver algo dinâmico, contrariamente a algo estático. Aqui também temos a oportunidade de desenvolver um projeto colaborativo e participativo. A evolução digital levou a uma cultura participativa, onde as práticas experimentais se baseiam em processos abertos de design e em estratégias baseadas em valor. O design tem na sua essência a procura de resposta dos diversos desafios, através da resolução do problema em si ou da proposta de mudanças inovadoras. A prática do design permite-nos criar melhores produtos, serviços, sistemas e espaços. O design participatório/ participativo segue um processo focado no utilizador bem como em torno da ideia de que todos podemos ser intervenientes no projeto de design, o que pode tornar o processo bastante complexo e demorado. No entanto, os resultados obtidos têm uma durabilidade e uma aceitação maior.

Os participantes/utilizadores são as pessoas indicadas para experimentar o produto/serviço pois são eles que na finalidade o vão utilizar e que sabem as suas próprias necessidades/frustrações. Apesar de ser um processo mais lento, é bastante proveitoso, pois é graças à interação dos participantes que é possível implementar novas funcionalidades, aumentar o interesse, aumentar a confiança para com os mesmos e desenvolver novas perspetivas. A contribuição de diferentes pontos de vista e experiências permite que os resultados do processo sejam flexíveis e acessíveis a mais indivíduos. Ao integrar diferentes tipos de pessoas, estamos também a evitar más interpretações entre o utilizador e o designer, atribuindo a este método uma responsabilidade ética.

O creative coding assume, portanto, esta direção participativa, quer na sua forma humana com pessoas de diferentes áreas, quer nas ferramentas que podemos explorar: código, som, imagem, câmara, etc, sendo que os inputs e output são infinitos. É sem dúvida esta vertente que distingue a ferramenta de outras tantas já existentes e utilizadas como Adobe XD, Adobe Animate, Figma, Adobe Illustrator, etc.

Na sua essência resultou numa exploração de componentes ainda não desenvolvidas, como foi o caso da ligação com a câmara, o som e o telemóvel (giroscópio), no pressuposto de se tornar visionário no desenvolvimento futuro e porventura exaustivo da capacidade destes mesmos componentes.

O facto de permitir ser utilizada em qualquer fase de desenvolvimento de um projecto, seja numa fase inicial para despontar ideias, seja numa fase intermédia de desenho, seja no final como ferramenta e suporte de visualização é também o que distingue este conceito de outros tantas. E também o facto de permitir ser uma ferramenta e um suporte final com animações em tempo real e ligação com elementos como a câmara, o som e reconhecimento facial que outros softwares não nos permitem. A vertente de mobile também foi algo a ser explorado dada a pertinência e uso constante deste dispositivo no nosso dia-a-dia e a facilidade que nos propõe de acesso a qualquer tipo de website, e também por ser algo que ainda não tinha sido muito explorado tendo em conta as referências encontradas.

*Design participatório* - Design participatório/participativo, surge da revolução do design centrado no ser humano, que exigiu uma redefinição e reformulação do processo de design. A crescente vontade de aproximar os utilizadores e restantes intervenientes ao longo do processo de design levou à expansão de novas ferramentas e técnicas participativas. Embora esta metodologia participativa aparente ser recente, e de facto ganhou popularidade como uma forma de conectar as pessoas, foi um termo que surgiu na Escandinávia por volta dos anos 70.

No desenvolvimento do projeto existiu a ambição de implementar uma secção de objectos, onde o utilizador poderia colocar formas 2D pré-estabelecidas como retângulos, triângulos e circunferências, bem como as respetivas formas 3D cubos e esferas, mas também o desenho “à mão livre”, semelhante a uma pen tool como nos é familiar no Adobe Illustrator. No entanto, ainda que tenha sido tentada, a curva de aprendizagem e o tempo despendido com os objetivos a atingir impossibilitaram essa realização, colocando-a para segundo plano, dando prioridade às componentes que mais se relacionam com o intuito do projeto, que é sem dúvida a experimentação e animação dos componentes com uma vertente de aleatoriedade. Esta possibilidade poderá ser desenvolvida futuramente, a partir do projecto. Algumas animações com um maior grau de complexidade foram ambicionadas, no entanto só serão possíveis de desenvolver posteriormente.

Como já referido anteriormente, em vez de produzir algo preciso com idealização do output final, conseguimos produzir resultados imprevisíveis, infinitamente e quantas vezes desejarmos, dada a rapidez de produção e resposta do código/ferramenta.

Este projecto visa questionar de que forma o Creative Coding pode ser parte integrante no campo do design gráfico de forma a proporcionar a criação de diferentes possibilidades visuais com ambição da inegração do código como ferramenta para o design gráfico.

Rather, the computer is a meta-tool: it makes other tools.  
We must learn to create tools ourselves. [...] a new phase of maturation for design's relationship to technology, when the definition of design extends to the creation of new tools that enable and empower others to design. After all, the computer is exactly that: a tool for creating tools.

(Blauvelt, A. 2011)

In the computer, man has created not just an inanimate tool but an intellectual and active creative partner that, when fully exploited, could be used to produce wholly new art forms and possibly new aesthetic experiences.

(Noll, M. 1982)

Conclui-se o projecto com estas duas citações de Blauvelt e Noll que nos dizem que o computador é a meta-ferramenta que o designer utiliza no desenho de estruturas e sistemas desbloqueadores ao exercício do design, é um parceiro do designer que quando bem explorado permite criar novas formas de arte e novas experiências visuais.

O creative coding expande o design gráfico nesta perspetiva de criação de ferramentas, ferramentas essas disponíveis para todos, neste caso na web de forma a não necessitarmos da instalação de software, o que facilita a sua utilização tanto a título institucional como a título individual. A proposta de um desenvolvimento em mobile facilita ainda mais o acesso de forma imediata a esta ferramenta.

O projecto pretendeu refletir no futuro do design que tendencialmente se desenvolve na criação destas ferramentas ao invés do foco exclusivo na criação de soluções.

No entanto, qualquer que seja o nosso envolvimento na criação da ferramenta, está intrínseco o nosso papel enquanto designers e o nosso contributo na mesma, e, por isso, nunca nos distanciamos do design em si.

# (6) BIBLIOGRAFIA E WEBLOGRAFIA

**about | openFrameworks.** (n.d.). Openframeworks.cc. Retirado em Fevereiro 12, 2022 de <https://openframeworks.cc/about/>

**A. Edmonds, M., & A. Edmonds, E.** (2009). WHAT IS GENERATIVE ART? (p. 31). Retirado em Março 21, 2022, de [https://www.researchgate.net/profile/Ernest-Edmonds/publication/233128802\\_What\\_is\\_generative\\_art\\_links/00463521693ec679c8000000/What-is-generative-art.pdf](https://www.researchgate.net/profile/Ernest-Edmonds/publication/233128802_What_is_generative_art_links/00463521693ec679c8000000/What-is-generative-art.pdf)

**Alessandro, L.**(1894). Post-Digital Print: a future scenario. Retirado em Novembro 15, 2021 de [https://s3.us-west-2.amazonaws.com/secure.notion-static.com/c6963eb8-6873-4132-aa59-d57c49f022b5/2ano\\_Ludovico\\_Alessandro\\_-\\_Post-Digital\\_Print\\_The\\_Mutation\\_of\\_Publishing\\_Since\\_1894.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Credential=AKIAT73L2G45E IPT3X45%2F20220323%2Fus-west-2%2Fs3%2Faws4\\_request&X-Amz-Date=20220323T114931Z&X-Amz-Expires=86400&X-Amz-Signature=6b6e7e40befa7161c0391f5d9241451cee31652e9ee6664f7d03108813f26026&X-Amz-SignedHeaders=host&response-content-disposition=filename%3D%2ano\\_Ludovico%252C\\_Alessandro\\_-\\_Post-Digital\\_Print\\_The\\_Mutation\\_of\\_Publishing\\_Since\\_1894.pdf%26x-id=GetObject](https://s3.us-west-2.amazonaws.com/secure.notion-static.com/c6963eb8-6873-4132-aa59-d57c49f022b5/2ano_Ludovico_Alessandro_-_Post-Digital_Print_The_Mutation_of_Publishing_Since_1894.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Credential=AKIAT73L2G45E IPT3X45%2F20220323%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20220323T114931Z&X-Amz-Expires=86400&X-Amz-Signature=6b6e7e40befa7161c0391f5d9241451cee31652e9ee6664f7d03108813f26026&X-Amz-SignedHeaders=host&response-content-disposition=filename%3D%2ano_Ludovico%252C_Alessandro_-_Post-Digital_Print_The_Mutation_of_Publishing_Since_1894.pdf%26x-id=GetObject)

**Algorithmic Art: Beyond the Artist – USC Viterbi School of Engineering.** (n.d.). Retirado em Fevereiro 02, 2022 de <https://illum.usc.edu/algorithmic-art-beyond-the-artist/>

**Algorists** (n.d.). Retirado em Fevereiro 12, 2022 de <http://dada.compart-bremen.de/item/collective/1>.

**Almeida, A.** (2020). Development of a platform for generation of icon-based graphic codes. Faculdade de Ciências e Tecnologia, Coimbra.

**A. Michael Noll: A Computer Generated Ballet – ACM SIGGRAPH ART SHOW ARCHIVES.** (n.d.). Digitalartarchive.siggraph.org. Retirado em Fevereiro 8, 2022, de <https://digitalartarchive.siggraph.org/artwork/a-michael-noll-a-computer-generated-ballet/>

**Basu, R.** (2019). Algorithms Are a Designer’s New BFF—Here’s Proof. Retirado em Novembro 3, 2021 de <https://eyeondesign.aiga.org/algorithms-are-a-designers-new-bff-heres-proof/>.

**Benney, M., & Kistler, P.** (n.d.). Robbie Barrat – Artist Profile (Photos, Videos, Exhibitions). AIArtists.org. Retirado em Novembro 19, 2022, de <https://aiartists.org/robbie-barrat>

**Boden, M. A., & Edmonds, E. A.** (2010). What is generative art? University of Sussex, University of Technology, Sydney, Centre for Cognitive Science, Creativity & Cognition Studios.

**Bohnacker H., Gross B, Laub J., Lazzeroni C.,Lazzeroni, Frohling, M.** (2018). Generative design : visualize, program, and create with JavaScript in p5.js. Princeton Architectural Press.

**Carvalhais, M.** “Miguel Carvalhais.” Www.carvalhais.org, www.carvalhais.org/. Accessed 8 July 2023.

**Carvalhais, M., & Cardoso, P.** (2018). Empathy in the ergodic experience of computational aesthetics. Editorial Coordinators: Rufus Adebayo, Ismail Farouk, Steve Jones, Maleshoane Rapeane-Mathonsi, 220

**Colomina, B., & Wigley, M.** (2016). Are we human? : notes on an archaeology of design. Lars Müller.

**Copeland, M.**(2016). A Diferença Entre Inteligência Artificial, Machine Learning e Deep Learning. Retirado em Outubro 25, 2021 de <https://medium.com/data-science-brigade/a-diferença-entre-inteligência-artificial-machine-learning-e-deep-learning-930b5cc2aa42>

**Dinamo Font Gauntlet.** (n.d.). Dinamo Font Gauntlet. Retirado em Setembro 20, 2022, de <https://fontgauntlet.com/>

**Domenico Quaranta.** (2013). Beyond new media art (p. 293). Link Editions., Retirado em Julho 20, 2022, de [https://www.ocopy.net/wp-content/uploads/2016/04/quaranta-domenico\\_beyond-new-media-art.pdf\\_pag\\_36](https://www.ocopy.net/wp-content/uploads/2016/04/quaranta-domenico_beyond-new-media-art.pdf_pag_36)

**DrawBot Documentation — DrawBot 3.129.** (n.d.). Www.drawbot.com. Retirado em Julho 13, 2022, de <https://www.drawbot.com/>

**Drumm, P.** (2016). How Digital Design Will Shape What it Means to be Human. Retirado em Fevereiro 13, 2022 de <https://eyeondesign.aiga.org/algorithms-are-a-designers-new-bff-heres-proof/>.

**Dunne, A., & Raby, F.** (2013). Speculative Everything : Design, Fiction, and Social Dreaming. Mit Press.

**Eye Magazine | Feature | Reputations: Karsten Schmidt.** (n.d.). Eye Magazine. Retirado em Junho 20, 2022, , from <https://www.eyemagazine.com/feature/article/reputations-karsten-schmidt>

**Ferreira, D.** (2019). Design Editorial Algoritmico. Faculdade de Ciências e Tecnologia, Coimbra.

**Frieder Nake - 3 Artworks, Bio & Shows on Artsy.** (n.d.). Retirado em Julho 10, 2022, de [www.artsy.net](http://www.artsy.net). <https://www.artsy.net/artist/frieder-nake>

**Galanter, P.** (n.d.) Computer Art. Retirado em Janeiro 3, 2022 de [https://www.philipgalanter.com/downloads/ga2003\\_paper.pdf](https://www.philipgalanter.com/downloads/ga2003_paper.pdf)

**Galanter, P.** (2003). What is Generative Art? Complexity Theory as a Context for Art Theory. 21., Retirado em Maio 8, 2022, [https://philipgalanter.com/downloads/ga2003\\_what\\_is\\_genart.pdf](https://philipgalanter.com/downloads/ga2003_what_is_genart.pdf)

**Galanter, P.** (2004). ‘Generative art is as old as art’. An interview with Philip Galanter.

**Gaspar, R.** (2020). Future Forms of Nature: Sistema Generativo para o Festival Forte. Faculdade de Ciências e Tecnologia, Coimbra.

**Generative art** (n.d.),Retirado em Outubro 15, 2022, de Scribd: <https://pt.scribd.com/document/121065935/Generative-Art>

**Generative Art: Best Examples, Tools & Artists** (2020 GUIDE).

(n.d.). AIArtists.org., Retirado em Março 11, 2022, <https://aiartists.org/generative-art-design>

**Gosling, E.**(2019). No. 122: Psychedelic animation, design by artificial intelligence + more. Retirado em Novembro 3, 2021 de <https://eyeondesign.aiga.org/design-diary-no-122/>

**Hansen, M. S.** (2019). public class Graphic\_Design implements Code { // Yes, but how? }: An investigation towards bespoke Creative Coding programming courses in graphic design education

**Hansmeyer, M.** (n.d.). Michael Hansmeyer - Digital Grotesque I. Michael Hansmeyer - Computational Architecture. Retirado em Agosto 20, 2022, de <https://www.michael-hansmeyer.com/digital-grotesque-I>

**Heiskanen, K.** (2018). Parametric Modelling in Website Design: A Solution for Effi- cient Requirements Specification and Fast Development. Aalto University - Master’s Thesis

**Hiesinger, K. B., Michelle Millar Fisher, Emmet Byrne, Maite Borjabad López-Pastor, RyanZ., Blauvelt, A., Juliana Rowen Barton, Emma Yann Zhang, Srećko Horvat, Cogdell, C., LebrónM., Gorbis, M., Syms, M., Latour, B., Wood, D., Jackson, N., Bove, M., Orkan Telhan, Linyee Yuan, & Rapley, C.** (2019). Designs for different futures. Philadelphia Museum Of Art ; Minneapolis.

**Hisour** (n.d.). Retirado em Dezembro 11, 2021 de <https://www.hisour.com/pt/generative-art-21197/>.

**Howarth, D.** (2017, Fevereiro). Generative design software will give designers “superpowers”. Dezeen. Retirado em Fevereiro, 15 de 2022 de <https://www.dezeen.com/2017/02/06/generative-design-software-will-give-designers-superpowers-autodesk-university/>

**IBM.** (n.d.). What are Neural Networks? | IBM. Www.ibm.com. Retirado em Agosto 11, 2022, de <https://www.ibm.com/topics/neural-networks>

**Information Aesthetics: An heroic experiment - Scientific Figure on ResearchGate.** Retirado em Fevereiro 17, 2022, de [https://www.researchgate.net/figure/One-of-the-drawings-on-display-at-Georg-Nees-show-Generative-Computer-Grafik-February\\_fig3\\_241715762](https://www.researchgate.net/figure/One-of-the-drawings-on-display-at-Georg-Nees-show-Generative-Computer-Grafik-February_fig3_241715762)

**Kistler, P., & Benney, M.** (n.d.). Mario Klingemann – Artist Profile (Photos, Videos, Exhibitions). AIArtists.org. Retirado em Maio 5, 2022, de <https://aiartists.org/mario-klingemann>

**Klein, A., Kries, M., & Christoph Thun-Hohenstein.** (2022). Hello, Robot. Vitra Design Museum.

**Lehni, J.** (2011). Typeface As Programme. Retirado em Novembro 15, 2021 de [https://www.typotheque.com/articles/typeface\\_as\\_programme](https://www.typotheque.com/articles/typeface_as_programme)

**Leon A. Gatys,Alexander S.Ecker, Matthias Bethge.** (2015). A Neural Algorithm of Artistic Style. Retirado em Janeiro 3, 2022 de <http://www.csse.monash.edu/~jonmc/research/Pa-pers/genDesignFG04.pdf>

**Lima, J.** (2015). Contribuição da identidade dinâmica generativa para a criação

de uma identidade corporativa. ESAD - Escola Superior de Artes e Design, Porto.

**Maeda, J.** (2019). How to speak machine : computational thinking for the rest of us. Portfolio//Penguin.

**Magalhães, M.** (2020). Design generativo: experiência sinestésica na música eletrónica. ESAD - Escola Superior de Artes e Design, Porto.

**Mark M., Oliver C. Bown:** Towards a Creativity Support Tool in Processing. Understanding the Needs of Creative Coders. ACM Press 2013, Page 143-146, Cited According to: Stig Moller Hansen: Public Class Graphic\_Design Implements Code { //yes, but how?}: An Investigation Towards Bespoke Creative Coding Programming Courses in Graphic Design Education, Aarhus 2019, Pág 13 de <https://ebooks.au.dk/aul/catalog/book/340>

**McCormack, J., Dorin, A., & Innocent, T.** (2004). Generative design: a paradigm for design research.

**Melo, R.** “FBAUP - on Serendipity in the Digital Medium: Towards a Framework for Valuable Unpredictability in Interaction Design.” Sigarra. up.pt, 18 June 2018, sigarra.up.pt/fbaup/pt/pub\_geral.pub\_view?pi\_pub\_base\_id=274022. Accessed 8 July 2023.

**Mendes, S.** (2020). Post-Print: graphic design and augmented reality. ESAD - Escola Superior de Artes e Design, Porto.

**MuirMcNeill, H.** (2017). MuirMcNeil generates 8,000 unique covers for Eye magazine. Retirado em Novembro 10, 2021 de <https://www.dezeen.com/2017/09/11/muirmcneil-8000-unique-covers-eye-magazine-design-graphics/>

**Noll, A. M.** (2016). The Howard Wise Gallery Show Computer-Generated Pictures (1965) A 50th-Anniversary Memoir. Leonardo, 49(3), 232–239. Retirado em Julho 20, 2022, de <http://www.jstor.org/stable/43834354>

**O que é o processamento de linguagem natural?** (n.d.). Google Cloud. Retirado em Abril 15, 2022, de <https://cloud.google.com/learn/what-is-natural-language-processing?hl=pt-br>

**Oliveira, F.** (2017). Tipografia experimental em sistemas generativos. Tese de Mestrado. IADE-U – Instituto de Arte, Design e Empresa - Universitário, Lisboa.

**Oliveira, V.** “P5js – MDGPE.” Master in Graphic Design and Editorial Projects, 30 Mar. 2023, mdgpe.fba.up.pt/?tag=p5js. Accessed 8 July 2023.

**Ögüt, A.**(2019). From Self-Design to Algorithmic-Design. Retirado em Outubro 25, 2021 de <https://eyeondesign.aiga.org/processing-the-software-that-shaped-creative-coding/>

**Parente J., Martins T., Bicker J., Hardman P., and Machado P.,** “Working with type: approaches on generative and evolutionary typographic creation,” in Joint Proceedings of the ICCO 2020 Workshops (WS 2020), 2020.

**Parente J., Martins T., Bicker J.**, “Which type is your type?,” in Proceedings of the Eleventh International Conference on Computational Creativity, 2020.

**Parente J., Martins T., Bicker J., and Machado P.**, “Designing Dynamic Logotypes to Represent Data,” International Journal of Art, Culture and Design Technologies (IJACDT), vol. 8, iss. 1, 2019.

**Parente, T. Martins, and J. Bicker**, “Data-driven Logotype Design,” in 22 International Conference Information Visualisation (IV2018), 2018.

**Parente J., Martins T., and Bicker J.**, “Generative Type Design,” in Proceedings of the Ninth Typography Meeting (9ET), 2018..

**Pickard, N.** (2022). Parametric Design: What is It & How Is It Shaping The Industry. Transcend. Retirado em Dezembro 9, 2022, de <https://transcendinfra.com/parametric-design/>

**Reas, C., & Fry, B.** (2014). Processing, second edition. MIT Press.  
Rebelo, J. (2020). Desenho algorítmico de websites. Faculdade de Ciências e Tecnologia, Coimbra.

**Rehn, K., & Caluag, S.** (2012), Retirado em Janeiro 18, 2022, de <https://zzz.softdetours.com/animation/1961--Catalog--John-Whitney--Circles-B/>

**Riechers, A.**(2018). What’s the Difference Between Variable and Parametric Fonts?. Retirado em Outubro 25, 2021 de <https://eyeondesign.aiga.org/parametric-and-variable-typeface-systems-shape-shifters-for-letterforms/>

**Rodenbröker, T.** (2022). Tim Rodenbröker – Creative Coding als Schule des Denkens. Master.timrodenbroeker.de. Retirado em Setembro 14, 2022, de <https://master.timrodenbroeker.de/fazit.php>

**Rodenbröker, T.** (2018, November 23). Programming Posters • tim rodenbröker creative coding. Tim Rodenbröker Creative Coding. Retirado em Setembro 14, 2022, de <https://timrodenbroeker.de/programming-posters/>

**Rodríguez-Alegria E.**, (2008), “Narratives of Conquest, Colonialism, and Cutting-Edge Technology,” American Anthropologist 110, no. 1

**Rosa, M.** (2020). Uma exploração criativa da generatividade : do conceito à prática. Faculdade de Ciências e Tecnologia, Coimbra.

**Sadowski, J.**(2015). From Mega-Machines to Mega-Algorithms. Retirado em Outubro 17, 2021 de <https://eyeondesign.aiga.org/processing-the-software-that-shaped-creative-coding/>

**Sarcar, M. M. M., Mallikarjuna Rao, K., & Lalit Narayan, K.** (2008). Computer Aided Design and Manufacturing. Retirado em Setembro 14, 2022, de [https://books.google.pt/books?hl=pt-PT&lr=&id=zXdivq93WIUC&oi=fnd&pg=PR23&dq=Computer-aided+design&ots=sj9UHgjff&sig=414gzedWYb-9ItCgxrJ6L8H4bh8&redir\\_esc=y#v=onepage&q&f=false](https://books.google.pt/books?hl=pt-PT&lr=&id=zXdivq93WIUC&oi=fnd&pg=PR23&dq=Computer-aided+design&ots=sj9UHgjff&sig=414gzedWYb-9ItCgxrJ6L8H4bh8&redir_esc=y#v=onepage&q&f=false)

**Sernadela, J.** (2020). Abordagens de design generativo no contexto de identidade visual. Faculdade de Ciências e Tecnologia, Coimbra.

**Shea, K., Aish, R., & Gourtovaia, M.** (2005). Towards integrated performance-driven generative design tools. Automation in Construction.

**Shiffman, D., Fry, S., & Marsh, Z.** (2012). The nature of code. D. Shiffman.

**Shiffman, D.** (2015). Learning processing : a beginner’s guide to programming images, animation, and interaction. Elsevier/Morgan Kaufmann.

**Silva, A.** (2018). Literatura e Cibernética: para uma poética dos processos generativos automáticos. Faculdade de Letras, Coimbra.

**Soddu, C.** (2005). Generative Art: Visionary Variations. Hong Kong: Visual Art Center.

**Stinson, L.**(2019). An AI-powered Typeface that Looks Like Computer Handwriting. Retirado em Outubro 25, 2021 de <https://eyeondesign.aiga.org/an-ai-powered-typeface-that-looks-like-computer-handwriting/>

**Stinson, L.**(2021). Processing: the Software that Shaped Creative Coding, Part I. Retirado em Novembro 3, 2021 de <https://eyeondesign.aiga.org/processing-the-software-that-shaped-creative-coding/>

**Stinson, L.**(2021). Processing: the Software that Shaped Creative Coding, Part II. Retirado em Novembro 3, 2021 de <https://eyeondesign.aiga.org/an-oral-history-of-processing-part-two/>

**Stinson, L.**(2019). When Artificial Intelligence Becomes an Artform. Retirado em Novembro 3, 2021 de <https://eyeondesign.aiga.org/processing-the-software-that-shaped-creative-coding/>

**Tassinari, V., & Staszowski, E.** (2020). Designing in Dark Times. Bloomsbury Publishing.

**Tjark Ihmels, J. R.** (2007, Fevereiro 15). Die Methodik der generativen Kunst. Retirado em Dezembro 14, 2021, de [http://www.medienkunstnetz.de/themen/generative\\_tools/generative\\_art/1/links/00463521693ec679c8000000/What-is-generative-art.pdf](http://www.medienkunstnetz.de/themen/generative_tools/generative_art/1/links/00463521693ec679c8000000/What-is-generative-art.pdf)

**Van de Seyp, V.** “Vera van de Seyp.” Vera van de Seyp, veravandesey.com/about/. Accessed 8 July 2023.

**What is Computer Graphics? | Program of Computer Graphics.** (n.d.). Retirado em Maio 4, 2022, de [www.graphics.cornell.edu](http://www.graphics.cornell.edu). <https://www.graphics.cornell.edu/about/what-computer-graphics>

**What is Max? | Cycling ’74.** (2020). Cycling74.com. Retirado em Março 7, 2022, de <https://cycling74.com/products/max>

**Wu, T.** (2019). Into my Eyes. Retirado em Fevereiro 13, 2022 de <https://tongwumedia.com/blog/into-my-eyes>

# (7) ANEXOS

## GLOSSÁRIO

Algorithms É um termo usado para artistas digitais que criam arte algorítmica. Os co-fundadores foram Jean-Pierre Hébert e Roman Verostko. Hébert é creditado por cunhar o termo e a sua definição que está na forma do seu próprio algoritmo:

```
if (creation && object of art && algorithm && one's own algorithm) {
  return * an algorist * }
else {
  return * not analgorist *
}
```

Algoritmo Um algoritmo é um conjunto de instruções ou regras matemáticas que ajudam a calcular uma resposta para um determinado problema.

Arte Computacional A arte computacional é qualquer arte na qual os computadores desempenham um papel na produção ou exibição da obra.

Arte Algorítmica Como arte gerada por um computador, que é um subconjunto da arte generativa, a arte algorítmica é arte gerada por um algoritmo. Iniciou-se nos anos 60, juntamente com o desenvolvimento da tecnologia computacional, onde algumas das primeiras práticas foram realizadas por uma plotter. No entanto, esta arte precede o uso dos computadores. Hoje, a arte algorítmica refere-se a obras de arte determinadas por algoritmos ou geradas por computador, ou por um sistema que executa um conjunto de regras.

Auto-Design O auto-design foi criado por contraculturas, comunidades LGBTQI+ e movimentos sociais para a formação de identidade, expressão política e sobrevivência. Mas o auto-design também é usado por conservadores e extremistas de direita. Tudo isto acontece agrupado no design algorítmico, comandado por empresas de tecnologia e governos, caracterizados pela recolha cada vez mais invasiva dos dados dos utilizadores. (Ögüt, A., 2019)

Computação Gráfica É um campo entre a ciência da computação e o design, que é responsável por gerar imagens, na forma de pixels, desenhadas nos computadores. Dentro do processo de criação de imagens é responsável também pelo processo de texturas, renderização e iluminação desses modelos, até à exibição digital final. Com o desenvolvimentos dos softwares e hardwares, a precisão, velocidade e qualidade dos renders criados tem sofrido um aumento exponencial.

Canvas Canvas significa uma tela, um quadro no qual descrevemos e dispomos diferentes elementos.

Class e Id A diferença entre classe e ID em HTML é que os IDs são únicos, as classes não. Na prática, isso significa que vários elementos numa página podem ter uma classe, enquanto que só podem ter um ID e cada página pode ter apenas um elemento com esse ID. O atributo id é usado para apontar para uma declaração de estilo específica numa folha de estilo. Também é usado pelo JavaScript para manipular o elemento com o id específico e adicionar algum tipo de ação. A sintaxe para id é escrita com "#", seguido de um nome de id..

Desenho Assistido por Computador Consiste no uso de computadores para auxiliar na criação, modificação, análise ou otimização de um projeto. Este software é bastante utilizado no setor da engenharia, na indústria automóvel, na construção naval, aeroespacial, e no design arquitetónico e industrial, devido ao seu rigor na criação de objectos/ próteses 3D.

Design Algorítmico O Design Algorítmico é um processo de design baseado em algoritmos. Existe uma correlação entre o algoritmo e o resultado, permitindo com que o utilizador identifique as partes do algoritmo que geraram alguma parte em específico do produto final. O Design Algorítmico é um sub-conjunto do Design Generativo, que é criado com a interseção do Design Paramétrico.

Design Participatório Design participatório/participativo, surge da revolução do design centrado no ser humano, que exigiu uma redefinição e reformulação do processo de design. A crescente vontade de aproximar os utilizadores e restantes intervenientes ao longo do processo de design levou à expansão de novas ferramentas e técnicas participativas. Embora esta metodologia participativa aparente ser recente, e de facto ganhou popularidade como uma forma de conectar as pessoas, foi um termo que surgiu na Escandinávia por volta dos anos 70.

Design Sistémico O Design Sistémico baseia-se em duas disciplinas de estudo - Systems Thinking e Human-Centered Design, com objetivo de resolver desafios de design complexos, originados pela globalização, migração e sustentabilidade.

Figma O Figma é uma plataforma colaborativa com standards de indústria focada no Desenho e prototipagem de Interfaces, que permite simular pequenas animações, e interações. Foi lançada em 2016 por Dylan Field e Evan Wallace com o objectivo de criar uma ferramenta gratuita que permitisse a colaboração entre pessoas e equipas que criasse um produto para diversas plataformas e formatos. Em Setembro de 2022 foi comprada pela Adobe. Até ao momento e dada a brevidade do sucedido, mantém-se uma plataforma gratuita e sem quaisquer alterações.

Generativo Que gera ou que tem a capacidade de gerar, dando origem a algo, produzindo ou reproduzindo algo.

Input O que é inserido, recebido ou operado por qualquer processo ou sistema. Um lugar onde, ou um dispositivo através do qual, energia ou informação entra num determinado sistema.

|                                    |   |
|------------------------------------|---|
| <u>MVP</u>                         | Minimal Viable Product, ou produto viável mínimo é a versão mais simples de um produto que pode ser criada e lançada com uma quantidade mínima de esforço e desenvolvimento. Esta criação mais rápida do produto permite receber feedback dos utilizadores de forma rápida, de quando é posto em prática em testes de usabilidade.  |
| <u>Output</u>                      | A quantidade de algo produzido por uma pessoa, máquina, ou indústria. Um lugar onde a informação sai de um sistema.   |
| <u>Redes Neurais</u>               | As redes neurais, também conhecidas como redes neurais artificiais ou redes neurais simuladas, são inspirados no cérebro humano, imitando a maneira como os neurónios biológicos enviam sinais uns para os outros. As redes neurais artificiais são compostas por camadas de nós, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada nó, ou neurónio artificial, conecta-se a outro e tem um peso e limite associados. Se a saída de qualquer nó individual estiver acima do valor limite especificado, esse nó é ativado, enviando dados para a próxima camada da rede. Caso contrário, nenhum dado é transmitido para a próxima camada da rede. As redes neurais dependem de dados que são inseridos e “aprendidos”, progredindo ao longo do tempo. Tarefas de reconhecimento de fala ou reconhecimento de imagem podem levar minutos ou horas quando comparadas à identificação manual por especialistas humanos. Uma das redes neurais mais conhecidas é o algoritmo de busca do Google. |
| <u>Sistema</u>                     | Um sistema é definido como um conjunto de elementos conectados, associados ou interdependentes, de modo a formar um todo complexo.  |
| <u>Sistema Adaptativo Complexo</u> | Quando falamos de sistemas complexos não nos referimos a um sistema complicado, mas sim a um sistema que é composto por várias partes que entre si criam interações e ligações criando assim um sistema que é autónomo e que se organiza a si próprio dando origem a um sistema dinâmico que se adapta e está em constante mudança o que lhe confere a capacidade de não acabar.  |
| <u>Sistema Caótico</u>             | Ao contrário dos sistemas aleatórios, os sistemas caóticos ainda que difíceis de prever têm uma estrutura, exibem uma espécie de história. Neste tipo de sistemas é bastante complicado antecipar os resultados, isto deve-se à forma de actuar destes sistemas, ainda que o sistema tenha relações entre causa e efeito, o facto de serem sistemas não lineares faz com que pequenas alterações sejam criadas tornando-os cada vez mais complicados de prever ao longo do tempo.   |
| <u>Sistema Complexo</u>            | Dentro dos sistemas podemos subdividi-los em sistemas simples e sistemas complexos, sistemas altamente ordenados, altamente desordenados e complexos. Segundo Philip Galanter, durante o século XIX os sistemas altamente ordenados eram a escolha principal dos artistas, no século XX deu-se a ascensão dos sistemas altamente desordenados. Estes sistemas complexos situam-se entre os sistemas altamente ordenados e altamente desordenados, sendo que o resultado reside na complexidade. A complexidade e as teorias da complexidade são conceitos que nos últimos anos foram adoptados para o estudo de sistemas. Mas a complexidade não é uma ciência exacta, e por isso torna difícil definir qualitativamente os sistemas e se estes têm um alto grau de complexidade ou não. Nem todos os sistemas complexos partilham os mesmos níveis de complexidade.  |

|                                    |  |
|------------------------------------|--|
| <u>Sistema Orientado por Dados</u> | É um tipo de sistema que tem mais autonomia, apoiado nas decisões do artista/designer, os dados.   |
| <u>Sistema Random</u>              | Random, ou aleatório, representa algo que é incerto ou ao acaso. Os sistemas aleatórios são responsáveis pela introdução de sistemas altamente desordenados na arte generativa. A aleatoriedade pode ser utilizada para gerar variantes no processo o que torna o resultado inesperado e novo. Hoje em dia com o computador, o recurso a geradores de números pseudo-aleatórios tornou-se talvez o método mais utilizados nas técnicas generativas digitais. |
| <u>Software</u>                    | Conjunto de programas de computador, procedimentos e documentos associados no que diz respeito à operação de um sistema de processamento de dados.   |
| <u>System Thinking</u>             | Abordagem que contraria a análise tradicional de olhar para um sistema e reparti-lo. System Thinking defende uma visão integral de como as partes constituintes de um sistema se relacionam entre si e como os sistemas funcionam ao longo do tempo e dentro do contexto global (Lutkevich, 2020).   |
| <u>Tab</u>                         | Uma tab é um elemento gráfico que tem como função fazer uma separação de componentes, normalmente uma separação de alguma opção de menu.   |

Como já referimos anteriormente, Phillip Galanter foi um pioneiro no Design Generativo. Sempre que falamos de Design Generativo, a definição que o mesmo deu, em 2003, destaca-se. Por isso Thomas Peterson e Kristine Ploug realizaram a entrevista que se segue, em 2004, de forma a clarificar algumas questões que o mesmo aborda no seu documento *What is Generative Art? Complexity Theory as a Context for Art Theory (2003)*

## Entrevista Philip Galanter por Thomas Peterson e Kristine Ploug (2004)

We'd like to start by asking about your work with generative art in general. Tell our readers a bit about your activities.

First I'd like to thank you and your net publication not only for your interest in my work, but also for your coverage of the field of generative art as well. I've been working in various forms of generative art since entering college in 1971. Of course back then I didn't think of it as generative art. The term hadn't been invented yet. But I was very excited by the then relatively new realm of voltage controlled music synthesizers...the Moog, Arp, and so on. Most synthesizers were played with a piano-like keyboard. Sometimes they were used with a sequencer so playing musical phrases could be automated ... not unlike a player piano. But properly configured such a synthesizer could, in effect, also generate surprise by using noise sources, sample and hold modules, and so on. I found the notion of an instrument that was constantly creative to be very exciting indeed. Randomization seems to be a phase that all generative artists have to go through!

At the same time I was teaching myself computer programming and did some experiments with the very basic graphics technologies available then, again trying to create systems that would self-compose and surprise even me, the nominal creator. In terms of my formal studies, however, I started out in Physics but ended up getting my B.A. in Philosophy. After college I did a lot of work in non-generative music ... punk bands, industrial and experimental projects, electronic music and software creation, and very odd performance events. It was a simultaneously torturous and fulfilling time.

But for about the last 15 years I've been working in generative forms again. Even though for various reasons I wanted to work alone rather than collaboratively, I missed the surprise factor of working with others. Working in generative forms restores the surprise factor to some extent. But more and more I've gone beyond that as a motivation, and increasingly working with generative systems is more like experimental philosophy. Philosophy via experimental means.

The generative systems we create as artists are far simpler than the systems we encounter in everyday life. But by creating these comparatively minimal systems we can discover truths about the world that might otherwise be masked by the relative chaos of the day to day. In my own generative work I like to say that I am moving from nouns to verbs. My earlier work was about using systems to create art objects. At first abstract videos using analog feedback. Later digital prints and digital video installations using genetic algorithms, Markov-like processes, L-Systems, reaction-diffusion systems and so on.

From the point of view of the audience, however, the generative process was over by the time the work was presented. It was as if I was showing animal skeletons or dinosaur bones and asking the audience to imagine what it was like when these creatures were walking the earth. Now I am trying to create systems that reveal themselves as processes before the audience. Metaphorically, I want to show living breathing animals and not just bones. The artwork should be the generative process not the trace the generative process leaves behind. Verbs not nouns.

The idea of putting the art making process in the place of a pre-generated artwork is of course a key element in much generative art. One might make the assumption that generative art has grown out of a process-oriented aesthetic based on a twentieth century avant-garde practice. Would 'the art making process as the art work' not have been impossible without these art historical conditions?

In a very real sense art never goes away. It is a process of accretion, a layering of new ideas upon old ideas, creating an ever evolving mass. The old art is always with us even if it is hidden deep inside the newest art.

There is a very specific historical movement called 'process art' which involved the use of flexible or chemical materials, sometimes poured plastics or fiberglass. The artist would indeed give up some control and accept what the material would dynamically reveal itself to be. Generative art has a fuzzy border, and process art is near that border, some on one side, some on the other. Someday soon I hope there will be a second wave of process art. Imagine having buckets full of nano-machines. Splash them on a wall and they slowly etch patterns creating a mural. Or pour them on the floor, or perhaps add sand particles, and they self-organize creating sculptures. And perhaps these murals and sculptures 'stay alive' and are constantly changing. Generative art didn't start with computers, and I don't think it will end there either.

Tell us about a generative work, which has had special meaning for you, and which points out an interesting direction for generative art?

It's hard to pick out just one. I know on your site you have a picture of Hans Haacke's piece 'Condensation Cube'. It's a work I also included in a show I curated with Ellen K Levy called 'Complexity'. It is simple and elegant. A plexiglass cube is sealed with just a bit of water in the bottom. Tiny variations in temperature and surrounding air currents result in a constant cycle of evaporation and condensation patterns on the walls of the cube. The cube is, in effect, a weather system in a box. This work entirely anticipates everything we now know about complex systems and chaos. In fact at about the same time Haacke did this piece Ralph Lorenz, a meteorologist, was discovering the strange attractor and the intrinsic unpredictability of weather systems. And of course I always have to give a tip of the hat to John Cage. His use of the I Ching and other randomization methods to write music has inspired several generations of composers. The Zen inspired impulse to hear the value of sounds as they are rather

than as we wish them to be is a very important lesson both about art and life in general.

But the most exciting generative work I've encountered in the past few years is by an artist we will never know. Most folks know about the prehistoric cave paintings found in France and elsewhere depicting animals and so on. The oldest of those is about 35,000 years old. But in 1999 anthropologists discovered the oldest known artwork, and this work is more than 70,000 years old. It consists of triangular tiles inscribed in hand sized pieces of red ochre. It is an exploration of pattern and aesthetic form that would be clearly recognized as such by someone like M. C. Escher or generations of Islamic artists.

And it is generative in that an abstract autonomous system creates the form rather than the moment to moment intuition of the artist. Tiling systems are algorithms that existed long before there were computers. As I like to say, generative art is as old as art.

It seems that you understand generative art as all art based on systems, digital or non-digital, current or ancient. Does this not have the effect of encompassing too much, especially when you deal with ancient art forms? Isn't it important to remember autonomy as an equally important defining feature in generative art, and would this feature not exclude many historic (and quite static) art forms?

I know some artists emphasize the autonomous aspect of generative systems. The conceptual problem I have with that is that it's too easy to slip into comfortable postmodern/post-structuralist dogma about the death of the author and so on. For me the combination of complexity theory and generative art practice is, in fact, a corrective and partial antidote to the excesses of postmodernism.

But yes, of course, by definition generative art involves the artist releasing control to an external system. In my view, however, this is actually yet another reason to consider various ancient art forms as generative. Keep in mind, first of all, that the notion of the individual heroic expressive artist is a relatively recent and western idea. The vast majority of art worldwide is not about individual expression, but serves more as a sort of cultural glue and shared memory system. Art is one of the ways that cultures maintain and propagate themselves into the future. From this point of view the autonomy of systems for tiling, weaving, decoration, border patterns and so on is a strong survival adaptation. The culture survives even as individual artists come and go.

In addition, before the industrial age people had to manually execute the work we now trust to machines. Generative methods, such as tiling patterns, allowed a designer to utilize less expressive artisans to get the job done. It is exactly the self-contained autonomy of tiling patterns that allows the head architect or fabric designer to not worry about handing off the work. In pre-industrial times this was a great practical advantage offered by generative methods.

Your definition of generative art in the article 'What is

Generative Art, Complexity Theory as a Context for Art Theory' does not link generative art with any particular technology. Generative art works may therefore be non-digital, organic and so forth. Do you think this inclusion of non-digital works into the category of 'generative art' has happened in retrospect after generativity has manifested itself as a specific trend within digital art?

Well, just as a matter of historical sequence it is certainly true that I personally started thinking about generative art in the broadest sense only after first pursuing specific generative methods. But even in my case digital art didn't come first.

Analog systems such as music synthesizers and video feedback did. But there is no doubt that digital technology gives generative artists unprecedented leverage in creating and exploring systems. The tradeoff is that you end up creating these systems in the tidy virtual world of the computer and not the compelling physical visceral world where we live our messy emotional or spiritual lives.

Ultimately though what differentiates generative art from other kinds of art is the use of systems. And this is where new notions from complexity science can give us some critical insight. Some systems are highly ordered, like the tiling systems first explored by our prehistoric artist some 70,000 years ago. Other systems are highly disordered, like the randomized systems of John Cage. But both are, in their own way, quite simple. For generative artists today the real action is the complex in-between realm of partial order and partial disorder. That is where, for example, life itself exists. Haacke's 'Condensation Cube' hints at this, as does the interest of some artists in A-Life. But there is much much more work to be done. And thank goodness for that!

When discussing complex or simple systems as a part of the discussion of generative art works, it often seems that complexity is used as a criterion of quality. Don't generative artists also have some work to do in the conceptual department (Hans Haacke's Condensation Cube being the example of a successful piece)?

Oh, I agree with this and might even push the point further. First I'd say that complexity as a topic is no guarantee of quality. At this point in history it is certainly the right question to address. But some explorations are going to be more useful than others. 70,000 years ago our prehistoric friend exploring highly ordered systems via triangular grids was doing state of the art work. When John Cage explored highly disordered systems via randomization his work was absolutely critical to the evolution of sound art. But now tiling systems are fully integrated into most known cultures. And at the risk of upsetting some young electronic musicians out there, I am not so sure the world needs much more random number generator music. It has its place, but I don't think any new conceptual ground is being covered. At least with complexity you know you are entering relatively virgin territory.

But second, I think it's important to note that just because a work is exploring complexity as a topic that doesn't mean the piece itself has to

be complex in the everyday sense of the word. My own bias is that complexity art may well reinvigorate minimalist values. After all minimalism was never about making a minimum of art, it was about making maximal art with minimal means. It's not surprising that it lead to conceptual art because the stripped down nature of minimalism demanded precision, clarity, and focus to succeed. It's tempting to address complexity with complicated technology, but all too often the point is lost in a gaudy fetishistic spectacle. In my work, such as the recent 'Chaotic Conductor' I am using relatively simple technology that allows the audience to have an intuitive, first hand, visceral experience. A maximal concept via minimal means.

ESAD 2023  
Mestrado em Design de Comunicação

**Creative Coding:  
Expanding Visual Possibilities in Graphic Design**

**Maria Machado e Costa**

**Orientador**  
José Bártolo

**Co-Orientador**  
Rafael Gonçalves

**Contribuições**  
Afonso Azevedo  
Ana Moreira  
Constança Costa  
Francisco Soares Morais  
Jéssica Teixeira  
Margarida Matos  
Maria Paiva  
Mariana Jerónimo  
Matilda Oxelfelt

**Papel**  
Munken Print 90g

**Impressão e Acabamentos**  
Qualquerideia–Produção Gráfica



