Marilia Lyra Bergamo, André Luiz Silva, Victor Hugo Soares Valentim*

Design Gráfico com Codificação Criativa: Utilizando a Arte Abstrata como Lógica Bidimensional no Desenvolvimento de Projetos

* **Marilia Lyra Bergamo** Dra Marília Lyra Bergamo é pesquisadora e artista computacional. Ela lecionou design gráfico e de interação, artes digitais e game art nos últimos 17 anos. Atualmente é professora de arte de jogos/design interativo na Escola de Humanidades, Ciências Criativas e Sociais da Universidade de Newcastle, Austrália. Ela se formou em Design e Ciência da Computação e trabalhou com design de interfaces no início da carreira. Seu trabalho como educadora também começou em Design de Interface e gráfica impressa tradicional. Como artista, buscou produzir e pesquisar arte relacionada à interação, às imagens digitais e ao conceito de evolução e interação com sistemas digitais.
marilia.lyrabergamo@newcastle.edu.au
ORCID 0000-0001-9105-7886

André Luiz Silva MA André Silva é um designer multidisciplinar que trabalha na intersecção entre arte, arquitetura e design digital. Tem interesse em metodologias de projetos, Interação Humano-Computador, Técnicas Tradicionais e Tecnologia de Computação para desenvolver Arquitetura, Produtos e Arte. Atualmente é doutorando na linha de pesquisa Poéticas Tecnológicas do Programa de Pós-Graduação em Artes da Escola de Belas Artes da UFMG. Ele tem experiência no ensino de disciplinas de design gráfico, de interiores e de moda, com foco em codificação criativa e abordagens de design generativo. Ele ensina arquitetura, com foco em história da arte, representação e design. Sua experiência de ensino e pesquisa concentra-se em metodologias de design digital, criatividade e fluxos de trabalho influenciados pela inteligência artificial.
andrearqurb23@gmail.com
ORCID 0000-0002-4984-4912

Resumo Este artigo descreve uma abordagem para ensinar codificação criativa para estudantes de design. Pode responder a uma situação em que a "apreensão da substituição da automação" da inteligência artificial e da aprendizagem automática no que diz respeito à produção de design gráfico pode afetar os alunos. Como tal, este artigo também pretende evocar a importância da diversidade na resolução de problemas e nos métodos. Este texto começou com uma revisão da literatura sobre estratégias para codificação criativa no ensino superior em design. É apresentada uma metodologia detalhada de uso da arte abstrata como princípio para aplicar a lógica bidimensional na criação de codificação. O método foi aplicado durante dois semestres em um curso de design de 120 horas composto por aulas práticas de laboratório e desenvolvimento de projetos. Por fim, dois projetos significativos onde o aprendizado tutorial de lógica e codificação resultaram em produtos de design gráfico foram selecionados para discutir o método. Este trabalho contribui para o tema, fornecendo dados adicionais que apoiam métodos de codificação criativa na educação em design visual, ao mesmo tempo que demonstra o potencial gráfico de tal abordagem.

Palavras Chave Design Gráfico, Lógica, Arte Abstrata, Codificação Criativa, Método de Design

Graphic Design with Creative Coding: Using Abstract Art as Bidimensional Logic in Project Development

Victor Hugo Soares Valentim Professor Adjunto de música, arte e tecnologia no CECULT - Centro de Cultura, Linguagens e Tecnologias Aplicadas da Universidade Federal do Recôncavo da Bahia, Músico, Produtor Musical, Artista Multimídia, Pesquisador de música e novas tecnologias, graduado em composição musical e mestre em arte e tecnologia pela Universidade de Brasília e doutorando em poéticas tecnológicas pelo PPG-Artes. Atua no cenário da música e da arte tecnologia, com experiência em pesquisa e criação artística das relações entre o som e a imagem, com trabalhos apresentados no Brasil e diversos países. Desenvolve pesquisas de criação de obras para fulldome (cinema para planetários), projeção mapeada, projetos em música, arte e tecnologias emergentes.

zivitoufmg@gmail.com

ORCID 0000-0002-0282-7947

Abstract *This paper describes an approach to teaching creative coding to design students. It may respond to a situation where 'apprehension of automation replacement' from artificial intelligence and machine learning regarding graphic design production may affect students. As such, this paper also intends to evoke the importance of diversity in problem-solving and methods. This text started with a literature review regarding strategies for creative coding in design tertiary education. A detailed methodology of using abstract art as a principle to apply bidimensional logic in creating coding is presented. The method was applied during two semesters in a design course of 120 hours consisting of laboratory tutorials and project development. Finally, two significant projects where logic and coding tutorial learning resulted in graphic design products were selected to discuss the method. This work contributes to the topic by providing further data supporting creative coding methods in visual design education while demonstrating the graphic potential of such an approach.*

Keywords *Graphic Design, Logic, Abstract Art, Creative Coding, Design Method*

Diseño gráfico con codificación creativa: Uso del arte abstracto como lógica bidimensional en el desarrollo de proyectos

Resumen *Este artículo describe un enfoque para enseñar codificación creativa a estudiantes de diseño. Puede responder a una situación en la que la "aprehensión de la sustitución de la automatización" por la inteligencia artificial y el aprendizaje automático en la producción de diseño gráfico pueda afectar a los estudiantes. Como tal, este artículo también pretende evocar la importancia de la diversidad en la resolución de problemas y los métodos. Este texto comenzó con una revisión de la literatura sobre estrategias para la codificación creativa en la educación terciaria en diseño. Se presenta una metodología detallada del uso del arte abstracto como principio para aplicar la lógica bidimensional en la creación de codificación. El método se aplicó durante dos semestres en un curso de diseño de 120 horas que consta de tutorías de laboratorio y desarrollo de proyectos. Finalmente, se seleccionaron dos proyectos importantes donde el aprendizaje tutorial de lógica y codificación dio como resultado productos de diseño gráfico para discutir el método. Este trabajo contribuye al tema proporcionando más datos que respaldan los métodos de codificación creativa en la educación en diseño visual y al mismo tiempo demuestra el potencial gráfico de dicho enfoque.*

Palabras clave *Diseño gráfico, lógica, arte abstracto, codificación creativa, método de diseño*

Introduction

This paper presents a detailed description of using abstract art as a base to teach creative coding for design proposes. What is being demonstrated here is the establishment of a specific form of problem-solving in graphic creation, which consists of visualising a logical composition through the abstract art image and transforming this logic in code. Once this skill is acquired, the student is motivated to create logic from a specific graphic design context and apply it to alternative image generation, resulting in the development of visual communication products.

Besides the desire to practice-based teaching, the writing was also motivated by searching for alternatives to the contemporary problem regarding the impact of automation, artificial intelligence and machine learning on visual communication (Matthews, Shannon and Roxburgh, 2023). While the method was previously developed in 2022, the year 2023 can be seen as one where the automation tools have reached a level of complexity never seen before. As such, the authors understood that presenting alternative methods where logical thinking is used in the opposite to prompt description image generation is a necessary skill for graphic design communication students. The argument here rests on creative coding and critical thinking studies where the material for creation is logic education. These ideas have recently been shared by other researchers, such as Thais Xisto (2023) and Maria Costa (2023); the former presented an investigation where creative coding can be seen through the lens of the concept of critical consciousness as the ability to intervene in one's reality to change. The latter investigates creating and exploring visual proposals through programming as an integral and structuring part of the creative process in graphic design.

Another important point is that the argument about using computational thinking and creative code in art education is not a novelty, and a central paper on the field was written in 2015 by Aaron D. Knochel and Ryan M. Patton. The authors expressed in their paper, "Part of our argument for teaching art students computational thinking is to develop students' critical awareness regarding the electronic devices and software they use daily" (Knochel and Patton 2015, 27). Coding has now been extensively accepted as material for art production, even considered a medium by Ernest Edmonds (2020). However, design students may now worry about replacing their skills with automation, as presented by Matthews, Shannon and Roxburgh (2023). As such, the same critical awareness and material appropriation in art education can be applied to visual communication design regarding acknowledging the potential of machine learning and other forms of automation, allowing students to overcome apprehension and empower their creativity. For example, the late publication of the tool COLE (Jia et al. 2023) presents an automation system for graphic design creation as one problem-solving method aided by training different large multimodal models in specific phases within the method. It represents an advance

in the automation process but does not cover the complexities of method development that occur with visual communication design.

As such, thinking logically about images can still be a powerful tool in critical thinking for creative minds to support the diversity of many different forms of problem-solving regarding design.

Strategies for Creative Coding in Tertiary Education

Systematic research in Google Scholar, Project Muse, Research Gate and ACM DL Digital Library with the term 'creative coding' has yet to return many research outputs focused on tertiary education applying the term. However, the term is widely used in children's and high school education approaches. For teaching strategies for tertiary education, this paper discusses a few to evoke the importance of diversity in problem-solving and methods. This systematic research also led to the discovery of a recent book that approaches the teaching of creative coding from different perspectives and could be applied in future course development.

Another research worth highlighting is Teresa Terroso and Mário Pinto's (2022) paper describes a course method to introduce creative coding to master's degree students of uniMAD – ESMAD, Polytechnic of Porto, Portugal. According to the authors, the students have different backgrounds but are interested in graphic creation. Their method starts with introducing computing structures, then drawing with primitives of 2D and 3D coordinate systems, followed by control structures, user interaction, functions, arrays, objects, mathematics, and physics applied to creative coding and examples. This structure resembles the educational flow of Daniel Shiffiman's (2008) book. The authors recognise that the 'contents and topics ordering follows more or less the same as traditional introductory computer programming courses' (Terroso and Pinto, 2022 13:4). However, this paper agrees with the authors that having an image as the inspirational point proves effective in their method as well as this paper's one. The pacing of content delivery is presented in their paper as challenging in creative coding teaching, leading the process to curatorial advice to students' online learning resources, which also happens to us.

Additionally, a paper from Ilias Bergstrom and R. Beau Lotto (2015) discusses a specific method of creative coding used during years of practice. The technique, Code Bending, is based on the ideas of circuit-bending from electronics; the authors affirm that extensive training is not an absolute barrier to entry and facilitates experimentation and, as such, provides a complementary alternative to creatively learning code. Those practices are usually recognised for using functional boxes that can be connected or reconnected and have had their parameters changed to facilitate experimentation. It is used by coding environments such as Max/MSP for sound, OpenFrameworks for video, or Unity in game development. However, many of those environments support more than one form of art expression.

Also it may be important to present a recent study from Arne Duyver, Wouter Groeneveld and Kris Aerts (2023) presents a paper where software engineering and graphic design students were randomly paired to work on an open-ended creative coding assignment. The study aimed to increase creativity in software engineering students by emphasising experimentation and having fun as a goal. Interestingly, the study identifies a possible mindset in graphic design students towards the attitude described as intolerance towards coding knowledge and practices—also difficulties in getting the experimentation output from the heterogeneous groups in opposition to the homogeneous ones.

Finally, concerning books, Form+Code in Design, Art, and Architecture (Reas, Casey. et al. 2010) is an essential source of instruction, as Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction (Shiffiman 2008) and Processing: A Programming Handbook for Visual Designers and Artists (Mosher 2016) and Coding Art: The Four Steps to Creative Programming with the Processing Language (Zhang and Funk 2021). Regarding tools, Processing and P5.Js are used with this method as coding environments, depending on the student's availability to work online. Finally, the book Geometric Patterns with Creative Coding for the Arts (Artut 2023) may be a compelling source to experiment with in future course production, as it could be an alternative to visualising a logical composition to abstract art.

Using Abstract Art as a Principle to Apply Bidimensional Logic

Abstract art is a product of modern societies and reflects a worldview free from representation. Not mimetic, it consisted of autonomous graphics. Even colour has independent force within the composition, like points, lines, surfaces, volumes and shades. Cultural symbolism can be found in some abstract artworks, but total detachment from them can also be observed. Graphic design theory and the concept of visual elements are also products of this worldview, and the associations between both are easily identified in constructivism, suprematism, De Stijl, and other cross-productions between modern art and graphic design.

This historic crossing in Western culture favoured our students. Also, the city where the course was developed is one of many in Brazil where modern art and architecture have found their way into the day-to-day life of many students within and around the campus. Students could access internationally recognised Brazilian modernists near the campus, such as Oscar Neymar's monuments, Candido Portinari's paintings and illustrations, and some of Athos Bulcão's tiles and compositions. Having cultural and local connections with the images helps create a sense of ease when transforming them into descriptive coding.

The tutorial lessons were distributed as follows:

- **Module 01 - Learn the Cartesian space**
  Place circles at specific points in Cartesian space.
- **Module 02: Learning 2D primitives**
  Writing code using artwork.
- **Module 03 - Repetition**
  Learning repetitive distribution.
- **Module 04 - Condition**
  Learning change in conditions of repetition.
- **Module 05 - Exploring Color Effects**
  Dep understanding of colour space.
- **Module 06 - Angles and Waves**
  Learning angles to reposition images by rotation and translations.
- **Module 07 - Circles, Arcs and Spirals**
  Learning to create compositions with arcs.
- **Module 08 - Random**
  Learning the concept of aleatory.
- **Module 09 - Noise**
  Learning soft colour transitions.

Tutorial classes start with Module 01, 'Learn the Cartesian space, ' where no specific image is introduced, appealing to the student's previous understanding of graphic composing. It is paramount to note that the method uses square paper and watercolour to introduce the concept and continue using it during class before any coding time. Figure 1 represents one of the exercises in the Module 02 tutorial, 'Learning 2D primitives'. The goal of this exercise was to incentivise graphic design students to see images in the form of logical instructions through the practice of code writing.

Images such as Figure 1 help because they can be written only using the function quad(). Quad() is a function that allows students to draw images only by setting the coordinates of each square. This process requires little logical thinking and meticulous observation of the predicted point on the draw to the one that will work in coding. Adjustments usually had to be made, and some students had found it necessary to change the grid scale to 100 x 100. The method is generally applied by asking students to start the process by not including colour and allowing the P5.js or Processing environment to maintain its default configuration, resulting in a line version of the coding with white squares. This act of writing is the moment they realise how the drawing command works: what is written last is the figure on the top. This tutorial introduces (R, G, B) by values of 0-255.

It is important to point up that students learn that in Processing or P5.js, first, they define the characteristics of the shape, such as colour or stroke, and then they place the command to write the form. Understanding the colour principle here is a significant milestone; it helps students

understand that the values (255,0,0) represent pure red, and something like (120,0,0) means a shade red where the value of the light is reduced to accommodate the perception of dark. Following the same principle, a value such as (255, 120, 120) will add brightness to the red. Secondary colours are also introduced by the same principle, using a value such as (255,255,0) to set pure yellow and using the last value, B, only if they want to add shade or brightness.

Figure 1 Reconstruction of Suprematist Composition (with Yellow, Orange and Green Rectangle) (A 7675) 1915-16 by Kasimir Malevich. In the image, students are allowed to make adaptations in size to understand the composition through cartesian positions in a grid of 10x10 measurements
Font Marilia Bergamo Personal Archive, final code can be accessed in <https://editor.p5js.org/marilialb.mb/sketches/43k7RsuZ0>

Another important point on learning is that repetition is the next step in computational thinking. Once acquired, it is a fundamental concept that reduces the writing of the code and creates the jumping between mecha-

nically writing forms on a paper to understanding the logical configuration. The Module 03, 'Repetition' tutorial, can be inspired by Piet Modrian's images, such as Broadway Boogie-Woogie (1942). Introducing repetition is demanding because they need to calculate distance variations, and students are liberated not to replace abstract images precisely as they exist but to create variations of them using coding repetition. Here, primitive shapes like rectangles are added to the process. Those primitives use a different approach, and instead of thinking about each point on the drawing form, students are free to write squares only by their initial position followed by the width and height. It is also the moment where the concept of rectMode() is introduced, which determines if the initial point of the form will be on the corner, in the centre or as a radius. rectMode() is not a straightforward concept to introduce, and repetition is where the necessity to work after-hours with students who need help becomes a reality. It is also essential to emphasise that it is a bifurcation moment in the tutorial because the students who easily grasp the concept will quickly develop autonomy from this point forward. Simultaneously, the moment to curate online content to fill students' time becomes necessary, or they get bored in laboratory classes.

As a education processes, the tutorial introduces conditional thinking added to the repetition process in Module 04, and Frank Stella's artwork was once particularly efficient in promoting this understanding. Conditional teaching is more accessible than repetition. It is impossible to affirm that it results from intensive work with students after-hours, the access to online mixed instructions tutorials that create a better ground for the content delivered, or if the content is a more concrete concept. Further research is necessary to affirm any of the above possibilities. If() and else() for composition work with little rejection from the students in class. This moment is also when it becomes possible to introduce the concept of flexible canvas and allow students to work with variables to create and change conditions on the drawing.

In the Module 05 tutorial 'Exploring Color Effects', students can explore colour and experiment with their learning, giving them time to create compositions voluntarily. However, they are asked to explore colour deeply by using restricted arrangements of colour values and using them with conditions and repetition. The 'pause' to experimentation is crucial to check students' pacing, reinforce lost knowledge during the tutorials and acquire some autonomy. After this time, trigonometry is introduced in Module 06, 'Angles and Waves', but from a visual perspective. The first tutorial does not use abstract art images but only explains the rotation and translations as valuable tools to organise the content. It is an exciting class where transparent paper is used over the grid to exemplify these abstract concepts in creative coding. The translucent paper helps understand the concrete separation of drawing and cartesian multiple spaces used in coding.

Figure 2 represents one of the exercises of the Module 07 tutorial 'Circles, Arc and Spirals' and introduces the concept of arcs. The function arc() works with students like the function quad() did at the beginning of tutorial lessons. Using arcs to recreate Signal, a 1970 artwork by Sonia

Delaunay, helps because students have to forget about repetition or conditions and start the same mechanical process of understanding the logic of positioning arcs in the initial value again, thinking about their size and proportional reduction, and most importantly their initial and final degree openness. Writing each arc works as a memorisation activity, and using degrees instead of PI notation works better for the student's cultural background in Brazil. Repetition and conditions used to create circles and spirals are presented when students have acquired this sense of using degrees;

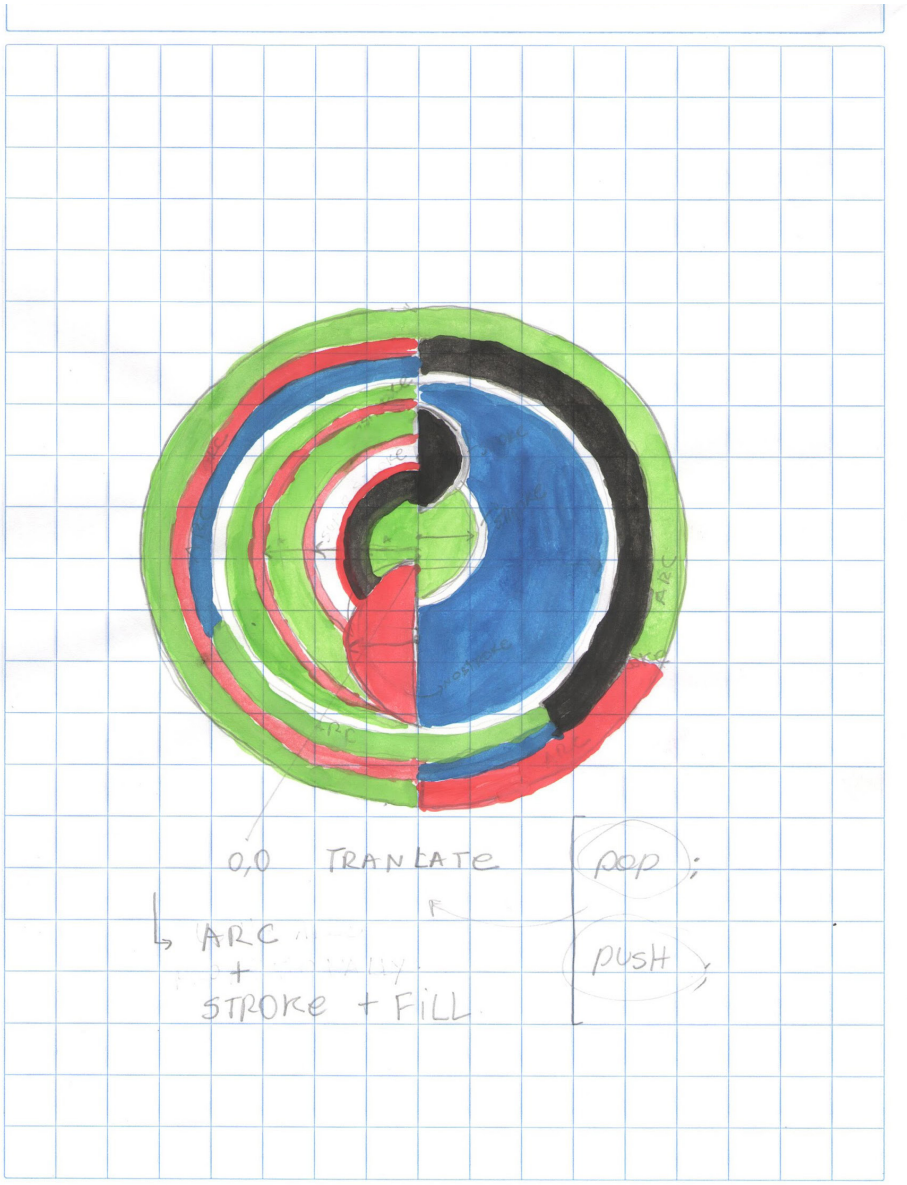

Figure 2 The recreation of Signal, a 1970 artwork by Sonia Delaunay. Here, the students are asked to think about the arcs from the point of 0,0 to be translated into the center of the composition by code.
Font Marilia Bergamo Personal Archive, final code can be accessed in <https://editor.p5js.org/marilialb.mb/sketches/zpQyvzZzL>

It is only In the Module 08 of tutorials called 'Random', that Rectangles Arranged According to the Laws of Chance, a 1917 artwork by Jeans Hans Arp, can be used to introduce the concept of random values. Students are asked to play with the square positions and random colours within a

strict pallet. Finally, in Module 09, the artwork Fire Evening, 1929 by Paul Klee can be introduced as an exercise to teach noise with colours. It is essential to draw attention to the examples of art that may deviate from pure abstract art once chance is not a modern concept but a more contemporary one. Also, Klee is not an artist of modernity but from cosmic and intuition against mechanical modernity. It is hard to present those philosophical distinctions within the method. However, creative coding requires a broader understanding of computational thinking. And that is why the method tries to function with a broader sense of abstraction to lead to future contemporary and generative reflection.

Finally, this method understands generative codging as a specific art that goes beyond what is presented in this paper, and as such, it should be a stage further. However, it is essential to highlight that some students demonstrated interest in moving to generative art.

Course Description and Student's Project

The method was applied within a discipline of over 120 hours of graphic design courses in Design in Federal University of Minas Gerais during two different semesters in 2022. The course was distributed with 75 hours of laboratory tutoring and 45 hours of project development. The laboratory hours were dedicated to transforming abstract images from specific artists of modernism into code, the method proposed and previously described in this paper. Students also use laboratory time to experiment with implementing their project logic in coding and transform the results into tangible visual images. The project development hours were exclusive to group problem-solving thinking, where the students proposed a graphic design problem and generated an abstract logic as a design alternative. The book Form+Code in Design, Art, and Architecture (Reas, Casey. et al. 2010) has been selected as an introduction to the theme for project development lecture hours. However, due to the nature of our method, classes rapidly change to modern art history and abstract art to create a bridge between coding and conceptual information. Also, it is crucial to emphasise that during project hours, students followed design thinking methods such as brainstorming, mind mapping, visual research, and mood boards before defining a design 'logic'. The logic was extensively used as a base to generate alternatives, and both logic and options were discussed between co-design-focused group sessions with peers before the development of final proposals.

To exemplify the process, this paper presents two students' works to demonstrate the course results. The students' names are anonymised to preserve their privacy. Figure 3 represents the first code generation of a group of students (Group 1); their project entitled 'New Views, Old Cities' can be described as 'Through a graphic synthesis, the old buildings and spaces urban areas are seen from new perspectives, transcreating their structure and its social relations in an abstract and synesthetic way'. The

goal was the production of posters to promote the visitation of specific architectural sites. They looked for modern architecture buildings in five different cities around the country and created a graphic logic representing each one. Figure 4 demonstrates the final poster resulting from the coding experimentation of Group 1. The second group (Group 2) had chosen to work with typography to explore specific worlds of different country areas. Entitled 'Brazilian Typography on Poster', the project uses words selected for their cultural significance from each region and strong accent sonority. They also described their work as 'Using typographical elements of fonts inspired by various types of vernacular letters from Brazil, present on posters, banners, signs, walls, etc, exploring their diversity and visual potential'. Figure 5 shows the coding experimentation; a set of rules followed each experiment to define the graphic result. Figure 6 shows the final posters developed using five different typographies for each word that had become the font name.

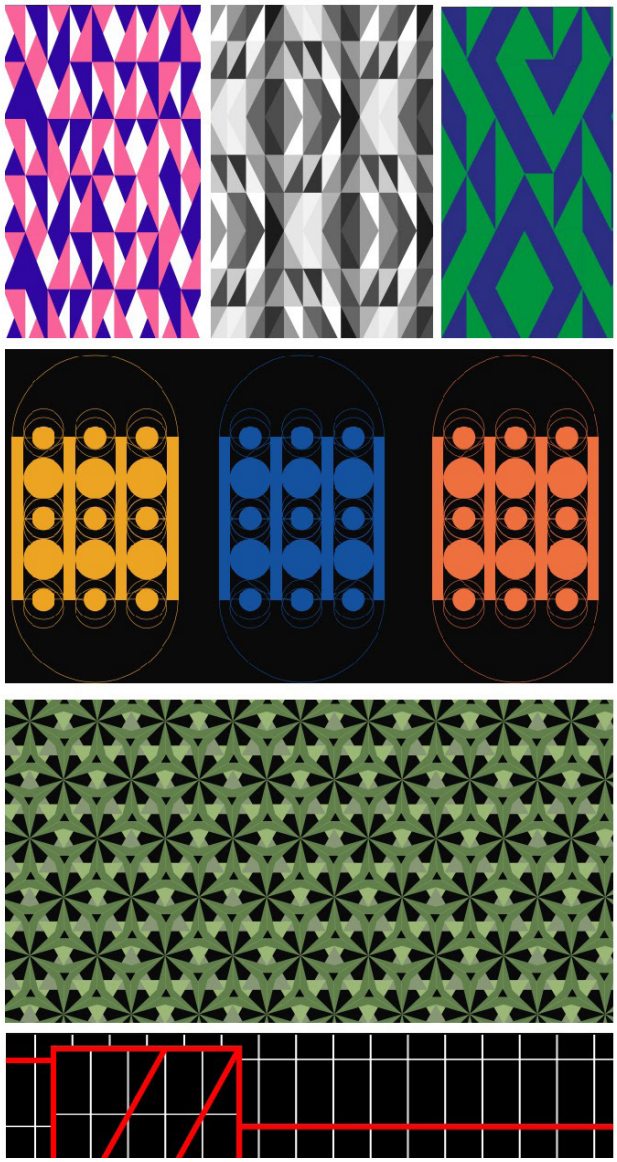

Figure 3 The results of the logic experimentation of Group 1 and can be understood as the generation of ideas in design thinking. .
Font Marilia Bergamo Personal Archive

Figure 4 The posters from Group 1, considered the final product of graphic design resulting from the method applied.
Font Marilia Bergamo
Personal Archive

Figure 5 The results of the logic experimentation of Group 2 and can be **understood as the generation of ideas in design thinking.**
Font Marilia Bergamo
Personal Archive

Figure 6 The posters from Group 2, considered the final product of graphic design resulting from the method applied.
Font Marilia Bergamo Personal Archive

Discussion

Working with graphic design students using this method was rewarding; however, at the beginning of the course, it was evident that if the contents were optional, they would not voluntarily choose to work their laboratory hours with coding writing. The predisposition to work with software for graphic composing is strong, and writing their image took some time to develop. Because tutorial hours required individual work, tutors faced the problems of finding some students who, from a few tutorials, had developed the solid ability to learn logic and start being creative with it and some who required long hours and direct support to develop basic skills in creative coding. A direct consequence is that those for whom creative coding was easily acquired as a graphic skill faced boredom during some tutorial hours, and tutors had to work with curatorial content, indicating third-party online tutorials and alternative resources to fill their time during tutorial classes. In contrast, for the students for whom creative coding was complex to transform in ability, tutors had to reinforce after-hours consulting time in the laboratory to guarantee a flux in tutorial lessons.

As such, the students exposed to this method were not first-year students and had been previously exposed to design thinking, image development, colour theory, and typography studies. As such, they were familiar with project dedication hours. The challenge on project hours was transforming ideas into coding logic while generating drawings and explaining their logic to other students in co-design-focused groups. The 'transformation of ideas into coding logic' required laboratory hours and, in some cases, consulting hours with the tutors. The graphic ideas were transformed into design products using visual and digital tools like bitmap, vector image editors, and editorial layout software.

It's important to point out that the technology and approach described in this article are different from the way of interacting with chatGPT or other Generative Images platforms. The main reason for this distinction is that in this approach, there is an emphasis on teaching and understanding the logic that creates the graphics, whereas interaction with Generative AI is achieved through a prompt that conceals the mathematical transformation model of the logic generation. Generative AI operates as systems that produce hyper realistic collages by assigning weights to words during the training process of the networks that constitute the Generative AI systems. In this article, we focus on how the student can independently discover logics that are under their control and do not represent black boxes for generating graphic images.

To sum up, all students quickly accepted the abstract art images as inspiration to write code. Abstract art and design have very concrete intertwinings in their historical development, and this proximity facilitated the process. Further research is yet necessary.

Conclusion

When graphic design is one of the skills to be transformed into automation, this paper argues that adding logic to the design process empowers students to face this challenge. Logic has worked as a written exercise to externalise their graphic thinking process and transform it into a concrete code capable of running and generating an image. As such, the image result is not an alien created by an external brain but a product of their particular intelligence. This method also leaves an open question about how introducing artificial intelligence to code generation is particularly appealing. However, allowing students to externalise their logic is a diverse process, from prompting-generated images to enriching the creative potential of design students. In conclusion, adding logical thinking is a method that contributes to the support of creative coding in tertiary graphic design education, and the results presented in the students' work also demonstrate the extensive visuality potential of this approach.

Future research

In 2023, one of the authors underwent separation from different institutions. Still, the authors of this paper have managed to maintain

distance collaboration in generative art workshops in our old institution and intend to present it in a different paper. The additional situation of different cultures may add exciting variables to the method. Understanding how other forms of images can be transformed into logic is essential. Moreover, how will using artificial intelligence in coding generation affect the empowerment of students facing automation? There is an expansive investigation in this field, and the authors hope this paper can inspire other educators to include logic in design education.

Referências

Artut, S. **Geometric Patterns with Creative Coding Coding for the Arts.** 1st ed. 2023. [Online]. Berkeley, CA: Apress, 2023.

Bergstrom, I. & Lotto, R. B. **Code Bending: A New Creative Coding Practice.** Leonardo (Oxford), Vol. 48, No. 1, pp. 25-13., 2015.

Costa, M . **Creative coding expanding visual possibilities in graphic design.** PhD diss. (ESAD - Escola Superior de Artes e Design). In 2023, (https://comum.rcaap.pt/handle/10400.26/45675) .

Duyver, A., Groeneveld, W., & Aerts, K. **Does Interdisciplinary Creative Coding Boost Creativity? A Mixed Methods Approach.** Preprint, submitted on 03 Jul 2023. (https://arxiv.org/abs/2307.00800)

Edmonds, E. **Art and Code: Programming as a Medium. In: Brooks, A., Brooks, E. (eds) Interactivity, Game Creation, Design, Learning, and Innovation.** ArtsIT DLI 2019 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 328. Springer, Cham, 2020.

Jia, P. & Li, C. & Liu, Z & Shen, Y. & Chen, X. & Yuan, Y. & Zheng, Y. & Chen, D. & Li, J. & Xie, X. & Zhang, S. & Guo, B. COLE: **A Hierarchical Generation Framework for Graphic Design.** Preprint, submitted on 28 Nov 2023. (https://arxiv.org/abs/2311.16974)

Knochel, A. D. & Patton, R. M. **If Art Education Then Critical Digital Making: Computational Thinking and Creative Code**, National Art Education Association, Vol. 57, No. 1, pp. 21-38, 2015.

Matthews, B. & Shannon, B. & Roxburgh, M. **Destroy All Humans: The Dematerialisation of the Designer in an Age of Automation and its Impact on Graphic Design**—A Literature Review, International Journal of Art & Design Education, Vol. 42, No. 3, pp. 367-383, 2023.

Mosher, M. **Processing: A Programming Handbook for Visual Designers and Artists**, Second Edition by Casey Reas, Ben Fry (review). Leonardo (Oxford). 49 (5), 476–477, 2016.

Reas, Casey. et al. **Form+code in design, art, and architecture.** New York: Princeton Architectural Press, 2010.

Shiffiman, D. **Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction.** Burlington, MA: Morgan Kaufmann Publishers, 2008.

Terroso, T. & Pinto, **M. Programming for Non-Programmers: An Approach Using Creative Coding in Higher Education**, International Computer Programming Education Conference (ICPEC). Open Access Series in Informatics (OASIcs), Vol.102, pp. 13:1-13:8, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

Xisto, T, **Playful Occupations: Mobile Creative Coding for Critical Consciousness.** PhD diss., (Program in Media Arts and Sciences (Massachusetts Institute of Technology). In 2023 (https://dspace.mit.edu/handle/1721.1/153095)

Zhang, Y. & Funk, M. **Coding Art: The Four Steps to Creative Programming with the Processing Language.** 1st Edition. [Online]. Berkeley, CA: Apress L. P, 2021.

Recebido: 25 de junho de 2024
Aprovado: 28 de setembro de 2024