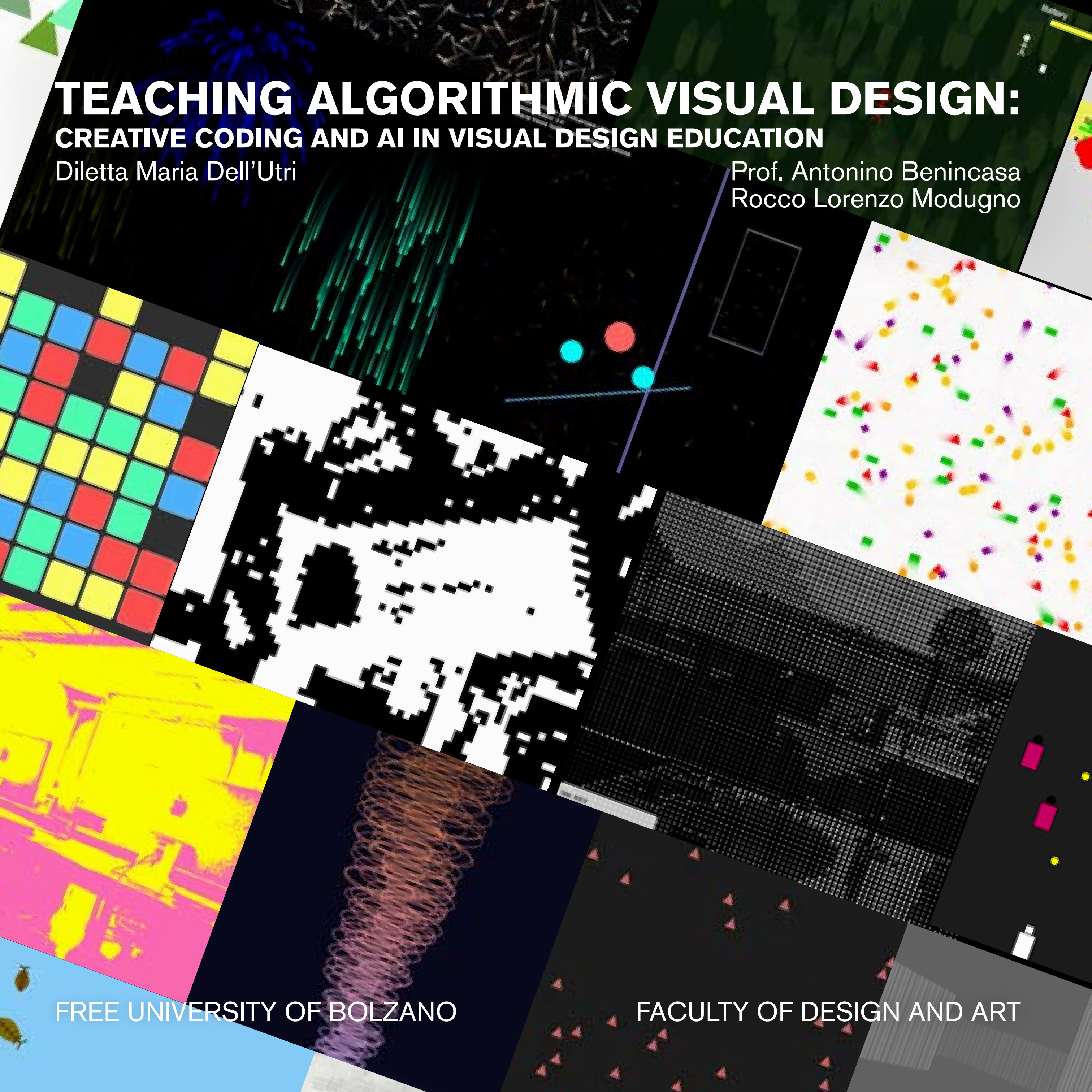


TEACHING ALGORITHMIC VISUAL DESIGN:

CREATIVE CODING AND AI IN VISUAL DESIGN EDUCATION

Diletta Maria Dell'Utri

Prof. Antonino Benincasa
Rocco Lorenzo Modugno

FREE UNIVERSITY OF BOLZANO

FACULTY OF DESIGN AND ART

ABSTRACT

Creative coding, long practiced within a niche of the design community, has become increasingly accessible, through high-level languages, libraries and now artificial intelligence. Through the use of libraries for creative coding and of Large Language Models, designers can emancipate themselves from their traditional software suites and focus on tools customized for their needs.
However, the use of these tools is still practiced only by a niche of the design world. Therefore, this thesis investigates didactic approaches that integrate classic computational graphics methods with AI-driven generative approaches, aiming to expand the creative workflow of design students, while also reinforcing their technical competencies and their critical engagement with AI generated content. In particular, one central concern is the risk of over-reliance on LLM outputs without sufficient technical understanding.
To explore the findings in practice, a workshop was conducted with 113 third-year students from the “Graphic and Communication” curriculum at IIS Andrea Palladio in Treviso over the course of five days. During eight hours, the students were introduced to basic algorithm concepts and machine learning foundation knowledge, as well as AI-assisted coding with p5.js, and prompt engineering with LLMs.
More than 200 works were analyzed, revealing not only creative potentials of these tools, but also crucial gaps regarding digital literacy and English proficiency, trends that align with broader national data, and that prompt to reevaluate the needs of modern education.
This research concludes that while AI can greatly support creative workflows, its responsible and successful use requires above all digital literacy, English proficiency and active engagement, and only then algorithmic thinking and basic understanding of the technical concepts behind them.

CONTENTS

D.P. Henry, Image produced with Drawing Machine 1. Ink on paper, 1962

1. INTRODUCTION

The intersection of design and technology has long shaped visual culture and creative practice. Today, the increasing accessibility of programming tools and the rapid evolution of generative artificial intelligence (AI) are transforming how designers work. This thesis situates itself within this transformation, focusing on the pedagogical integration of coding and generative AI into design education.

While computational design has been present within a niche of the design community for decades, the diffusion of high-level programming languages such as Python and visual ones such as Grasshopper 3D has significantly broadened access to computational tools. These practices allow designers to move beyond the constraints of traditional software, engaging in more experimental and autonomous creative processes. This change reflects a broader trend, as technology has democratized over the years, allowing an increasing amount of people to freely experiment with these tools.

In response to these evolutions, the aim of this research is to explore the teaching methodologies through which integrate coding and generative AI in design education, while also applying them in practice to evaluate their effectiveness. This way, this work seeks to develop algorithmic thinking within designers, giving them the cognitive tools to explore the challenges posed by modern technology.

Technological advancements have always impacted on the design field and workflow, often revolutionizing it and pairing with previously present techniques. The rise of publicly available generative AI tools has made it possible for anyone to create a wide range of content, even without understanding the medium or how the tools work.However, these possibilities are not without their dangers, as many ethical and legal questions emerge over its uses. For instance, as machine learning models require enormous sets of data to generate convincing material, there's an ongoing legal debate over whether training machine learning algorithms with copyrighted material constitutes as an infringement. In particular, the European AI Data Act (European Parliament and Council, 2024) does not directly address this topic, but it requires providers of general-purpose AI models to publish detailed reports of the used content, so that the detainers of copyrighted data can enforce their rights and request the removal. Furthermore, there has been especially in the art world a fierce debate over supposed "art theft" committed by the tools providers, as well as an alleged collusion of its users with the aforementioned theft.
In light of these debates concerning artificial intelligence and its uses, the course realized in the context of this thesis also aims to teach the fundamental theoretical concepts of machine learning, encouraging informed and critical analysis of the topic.

European Union, European Parliament and Council. 2024. *Artificial Intelligence Act, Regulation 2024/1689.*

Furthermore, s graphic design software becomes increasingly available and sophisticated, there is a growing need for designers to emancipate themselves from these tools. John Frazer warned of this risk as early as 1995 in *An Evolutionary Architecture*, where he cautioned against "an atmosphere where any utterance from the computer is regarded as having divine significance", a risk that this project directly addresses.

Frazer John 1995. *An Evolutionary Architecture.* London

This research opens the possibility for further investigation within the integration of creative coding and AI in design education. As technology continues to evolve rapidly, particularly in visual fields, addressing this gap becomes increasingly important. Future studies could build on this work by developing more extensive programs, such as semester-long courses, to deepen understanding and application.

Ultimately, this research encourages a more nuanced approach to computational design teaching, as it requires deeper research and application.

The thesis is therefore structured as follows: firstly, an overview of computational graphics and generative artificial intelligence history and techniques is presented, then of teaching methodologies that were applied. Finally, the findings were tested with an 8-hour workshop at the IIS Palladio High School in Treviso with 113 third year students of the "Graphics and Communication" curriculum, while a course was also written in the form of a manual and published on the "Design Ex Machina" website (currently hosted on *https://dilidede.github.io/DesignExMachina/*).

On the top: Anonymous IBM technician, *Girley1*, Vector Outlines, 1956-58
On the bottom: G. Petty, Esquire's Pin-Up Calendar, Acrylic and Watercolor, 1956.

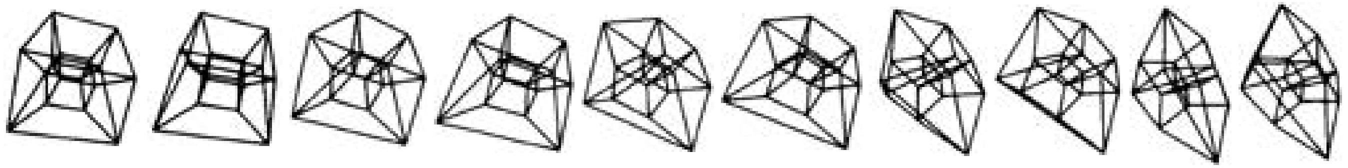
2. THEORETICAL FRAMEWORK

Although modern high-level programming languages, and AI in more recent time frames, have allowed a niche of the visual design community to work with computational technologies, it is not a new practice. As this thesis examines pedagogical strategies for designers unaware of such tools, it is pivotal to discover the historical evolution and core concepts of two related domains within this practice: Generative Artificial Intelligence (GAI) and Computational Design (CD).
Therefore, this section will examine the technical aspects and historical backgrounds of the topic of the pedagogical strategy provided in this thesis.

2.1. HISTORICAL BACKGROUND

While in the current time GAI and CD are for the most part two different practices, they do share intertwined origins in the mid 20th century computing researches, although they then diverged significantly in the late 1990s.

In particular, at the end of the 1950s and through the 1960s, the majority of the artistic and design computational works were born from the interest of engineers. For instance, as computers were initially a military tool, the earliest known computer artwork was realized between 1956 and 1958 by an anonymous IBM technician, who created "Girley1" using punch cards to trace a pin-up figure, presumably



Lee N. 2014. “From a Pin-up Girl to Star Trek’s Holodeck: Artificial Intelligence and Cyborgs” Springer.

taking inspiration from the woman on the December 1956 calendar page (Lee, 2014).

In 1961 Desmond Paul Henry, a philosophy lecturer at the Manchester University, created a “Drawing Machine”, which was able to produce algorithmic artworks, which were also exhibited in galleries (O’Hanrahan, 2018). In the summer of 1962 Micheal Noll, one of the pioneers of the field, began to use the “IBM 7090 and the Stromberg Carlson plotter to create computer art deliberately” (Noll, 1999) during his traineeship at the Bells Lab, a research facility where free exploration of modern computers was incentivized.

O’Hanrahan E. 2018. “The Contribution of Desmond Paul Henry (1921–2004) to Twentieth-Century Computer Art.”

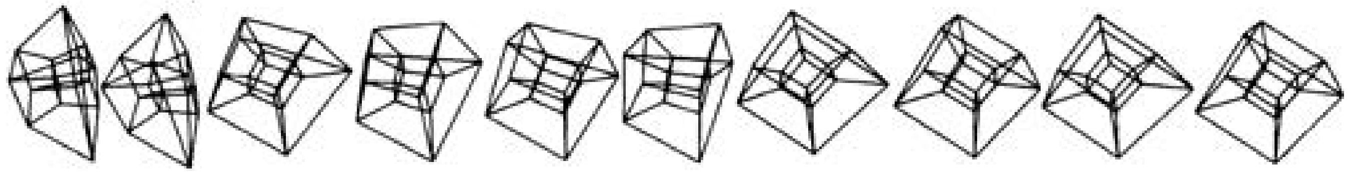
Noll A.M. 1994. “The Beginnings of Computer Art in the United States: A Memoir.”

In 1965, the first major exhibition emerged, in particular in the “Generative Computergrafik” exhibition at the Technische Hochschule of Stuttgart, as well as the Howard Wise Gallery Show “Computer-Generated Pictures”, showcasing seventeen works from Noll and eight from Bela Julesz, a visual neuroscientist (Noll, 2016).

Noll A.M. 2016. “The Howard Wise Gallery Show of Computer-Generated Pictures (1965): A 50th-Anniversary Memoir.”

Around the same time, Ivan Sutherland’s 1963 Sketchpad pioneered interactive graphical interfaces, laying groundwork for modern CAD systems, as his program allowed for precise geometrical drawing, as well as a primitive creation of 3D models (Sutherland, 1963).
Almost a decade later, in 1971, Harold Cohen began producing computer art and exposing it at the Fall Joint Computer Conference

Sutherland I.E. 1963. “Sketchpad. A Man-Machine Graphical Communication System.”



Cohen, P. 2017. “Harold Cohen and AARON.”

(Cohen, 2017). AARON, his computer system, was arguably one of the first and most important examples of computer generated artwork, as it was able to generate abstract line drawings and colored human figures by the 1990s.

At the top: Noll A.M., Projections of a rotating four-dimensional hypercube
At the bottom: Cohen H., Image produced by AARON,1990s.

A. Warhol, *Andy 1*. Digital Illustration. 1985

Malloy, J. 2003. “Algorithmic Art, Scientific Visualization, and Tele-immersion.”

O’Regan G. 2015. “Xerox PARC.”

Mazzone M. 2020. “Andy Warhol: Computational Thinking, Computational Process.”

Another central figure in the history of algorithmic art is Vera Molnár, who co-founded the Centre de Recherche d’Art Visuel in 1968 and used Fortran, one of the first programming languages, to create generative works (Malloy, 2003).

By 1970, the Xerox Alto was introducing Graphic User Interface, which personal computing technology researches were then implemented more successfully by other companies, for instance Apple with its Macintosh 128K in 1984 (O’Regan, 2015). As GUI democratized the use of computers and digital tools, prominent art figures including Andy Warhol began to adopt computers to create art (Mazzone, 2020), for instance the Andy1 or the Reigning Queens, both works from 1984. Around the same time, also Joseph Nechvatal produces large computer-robotic paintings, such as The Informed Man in 1986.

Concurrently, Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs), statistical models that date back to the 1950s, thanks to the advent of deep learning, enabled early sequential data generation, for instance, speech synthesis, foreseeing generative AI ethos.

The 1990s marked a schism between the two algorithmic design fields we analysed, with the popularization of CD tools on one side and the evolution of GAI ones on the other.

These years saw an explosion of computer graphics programs to be released to the public, for instance the first Adobe Inc. softwares, such as Illustrator in 1987 and Photoshop in 1990, respectively for vector and raster graphics editing, as well as PageMaker (1985) by Aldus Inc. for publishing design.

Additionally, high-level programming languages and libraries emerged, which were specifically realized for teaching non-programmers fundamentals of computer programming, for instance Processing in 2001, a Java library with simplifications in the syntax (Fry and Reas, 2007).

J. M. Allen. *Théâtre D'opéra Spatial*. Midjourney. 2022

Later in the decade, David Rutten introduced Grasshopper 3D, a visual programming language and environment in the Rhinoceros 3D CAD application.

Fry B. and Reas C. 2007. *Processing. A Programming Handbook for Visual Designers and Artists.*

In the mid 2010s, Ian Goodfellow's Generative Adversarial Networks (GANs) and Max Welling's Variational Autoencoders (VAEs) finally enabled high-fidelity data synthesis. In particular, Goodfellow's GANs made it possible to train models for image and text generation more precisely than ever (see next section). Already in 2015, Google's DeepDream began the process of popularizing generative AI, although still far from the omnipresence we have today. Finally, with the first public release of Dall-E in January 2021 and of Midjourney in July 2022, AI generated increasingly occupied place in the public debate, especially in the art field. For instance, the first place awarded by the Colorado State Fair to an AI-generated artwork by Jason Allen in 2022, was at the center of public debate on the value of these tools and their role in our society (Roose, 2022). Since then, AI presence in society and public discourse exponentially increased, as well as the quality itself of the generative technology. This time has cemented GAI's divergence from CD rule based paradigms, preferring autonomous content generation.

Roose K. 2022. "An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy." New York Times.

2.2. TEACHING METHODOLOGIES

Mishra P. and M.J. Koehler. 2006. "Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge."

As the objective of the course was finally defined, it was finally possible to outline the teaching methodologies to use. The pedagogical approach was informed by the TPACK framework (Mishra and Koehler, 2006), a model that recognizes how effective teaching emerges from a dynamic interplay between three fundamental knowledge domains: content (i.e. what we teach), pedagogy (i.e. how we teach) and technology (i.e. the tools we use). Through this lens, a methodology was designed where these domains converge.

First, the content knowledge (CK) is what anchors our work, in our case algorithmic logic and generative AI systems fundamental works. These aren't seen as mere technical skills, but rather a conceptual foundation that can empower designers to think computationally, thus using these skills in different programming languages and also in the prompt engineering. However, teaching these topics to creatives requires a thoughtful and planned pedagogical knowledge (PK). Therefore, we prioritized strategies that could lower intimidation barriers, in order to present the use of these technologies as an opportunity, rather than a challenge. This aim was achieved through offering students an immediate response of the possibilities that creative coding and LLM are able to provide them, for instance through the presentation of selected works during the workshop and animations on the website realized through p5.js, which all offered a direct link to the commented code. This sort of approach acknowledges that visual design students often engage differently with abstract and technical concepts than traditional computer science and STEM learners.

Finally, technological knowledge (TK) bridges theory and practice. The tools chosen for this course, p5.js and LLMs directly democratize access to coding while aligning with designers' immediate and visual workflows. Through this thesis, the intersection proposed by TPACK is the pivotal approach where efficient learning happens.

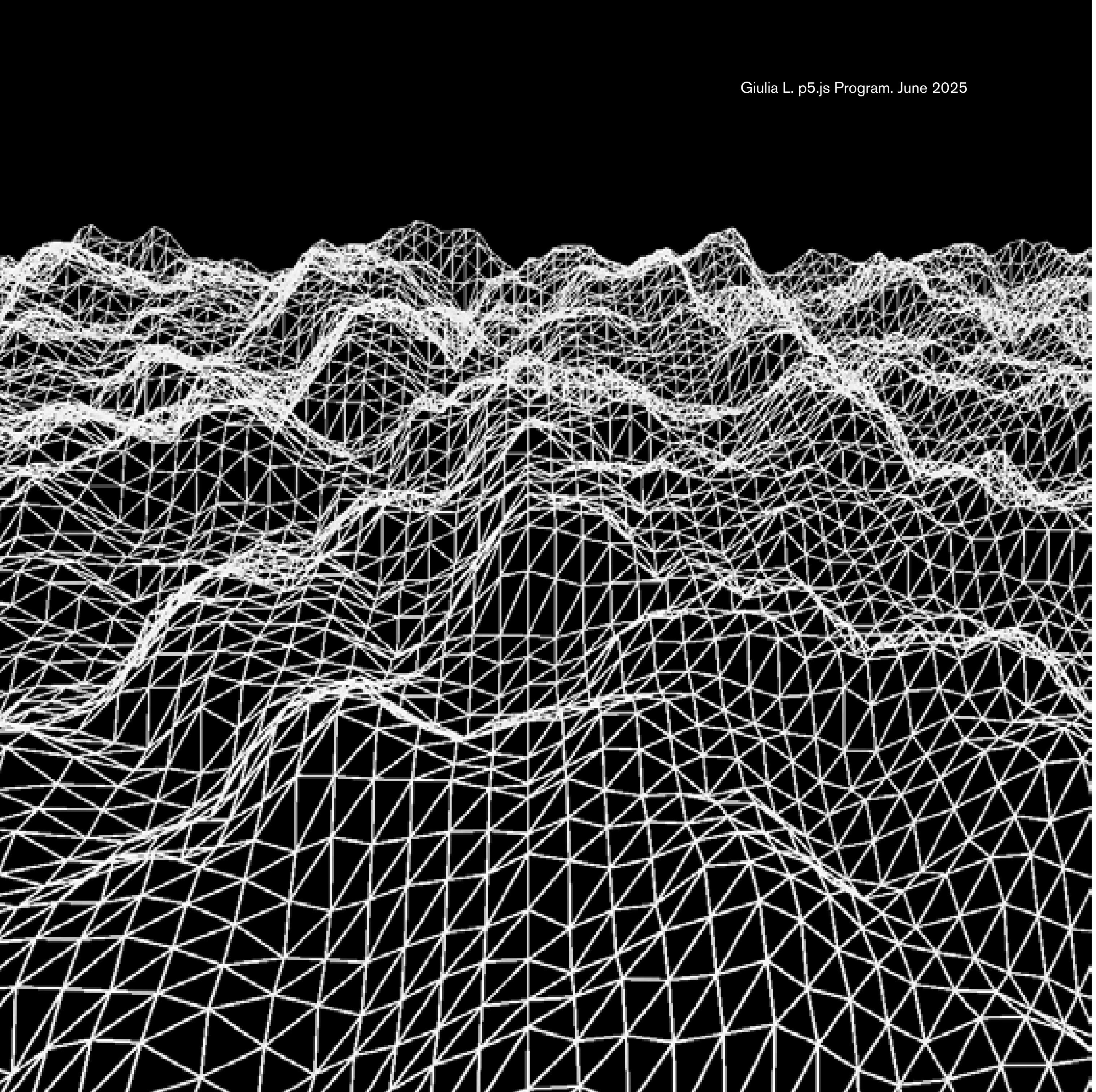
One of the most important resources for this process was *Four Approaches to Teaching Programming* (Selby, 2011), as it presented four current methodologies that could also be used in our course. Selby identified four strategies: code analysis, simple units, building blocks and full systems, which were throughly analyzed, in order to find the better fitting one for the project.

"Code Analysis" focuses on reading and understanding code before writing, which means that students begin by reading pseudocode or structured examples, emphasizing logic over syntax. This way, students are able to develop algorithmic thinking, avoiding to rely on a specific programming language. However, this sort of approach can cause disengagement within students that want to work with code and that might feel as they are not actually learning anything.

"Simple Units" instead divides programming into small, reusable tasks, as creating circles or take the webcam input. By gradually combining these units, students learn how to build more complex programs without the risk of feeling overwhelmed by having to start from scratch.

The third approach described by Selby is "Building Blocks", which introduces language constructs one at a time, often through "fill the blank" exercises. This allows immediate feedbacks in the coding environment, which facilitates learning. However, a common issue for students is the discouragement when correct syntax doesn't correlate to correct logic and vice versa.

Given the creative background of the target students, an important issue to address in our course was the risk of them being intimidated by programming, a field that might be perceived as needing a certain kind of rigor that does not align with the typical designer workflow and approach. Because of this reason, "Code Analysis" was the primary approach adopted in the workshop, also because of the limited time available. while the website course also integrated the "Building Blocks" approach, as the learner has all the time needed to understand the concepts and try them with the embed p5.js Web Editor links.



3. THE PROPOSED METHODOLOGY

This section outlines the framework for integrating algorithmic design and AI into visual design education, detailing the choice of tools, the structure of the workshop, and the complementary role of the website as a manual and documentation work.

3.1. THE TOOLS SELECTION

In both of its versions, the course used p5.js as its programming tool to be taught to the students. Since from the beginning of the research, it was clear that the preferred tool would likely be chosen between the ones of the Processing Foundation. This preference primarily comes from how Processing focuses on allowing creators to learn how to realize creative work through coding, especially to those that do not have any prior background in the field. Furthermore, all the tools provided by the Foundation are open-source and some of the most popular within the design community, both aspects that grant an extensive amount of educational content online that can help the students to expand their skills even after the completion of the course.

Finally, p5.js was identified as the best tool to use for the project, which is a JavaScript library that also offers a simplified syntax that is less intimidating for laypeople. Unlike its counterparts for other languages, such as Python or Java, it operates directly in the browser, allowing instant experimentation and use. This immediacy is crucial

On the opposite page: Stella G. p5.js Program. June 2025

in our context, with a high chance students might disengage when facing technical issues or extensive waiting times for downloads and installation. Moreover, p5.js is based on JavaScript, a language that might also be useful for designers as it is used in the Adobe ecosystem through the JSX syntax extention for the automation of iterative processes, a hurdle that designers are probably familiar with.

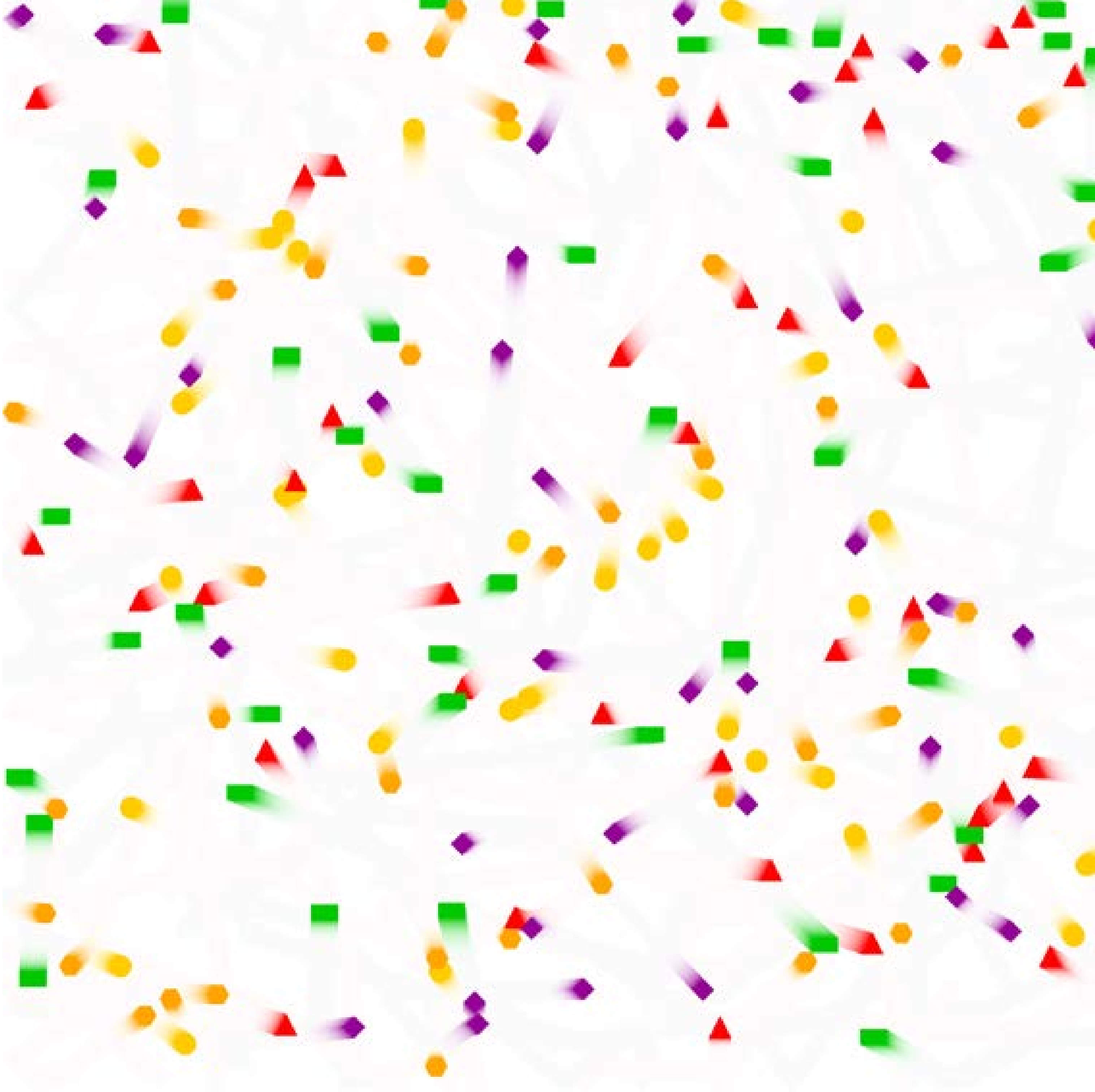
Equally important as the selection of the language, it was the integration of AI-driven code generation, in order to reach results that would otherwise be excessive of the designer newly learned skills. This integration was achieved through the use of Large Language Models, such as Open AI's ChatGPT or DeepSeek chatbot. In the workshop held in Treviso, all the students used ChatGPT, mostly because it allows more direct conversation sharing, but also because most of the students would probably be already familiar with it, given its larger popularity (Google Trends, 2025).

Google Trends. "ChatGPT and DeepSeek Search Comparison." Accessed 20 April 2025.

However, the website instead invited the learners to use the DeepSeek chatbot, as it often offers better code writing, while also being an open source software, as well as not being as environmentally detrimental as its OpenAI competitor (Moravec V. et al., 2025).

Moravec V. et al. 2025. "Environmental footprint of GenAI – Changing technological future or planet climate?"

Furthermore, students are invited to avoid passive reliance on AI and instead critically evaluate each generated response in order to use the AI not only as a "code generator", but also as a tool for effective learning. This proposed balance between the AI-aided automation and the comprehension of these technologies and of programming per se is central to this thesis goal of enhancing the designer workflow and understanding of modern technologies.



3.2. THE MANUAL WEBSITE STRUCTURE

The project website Design Ex Machina takes its name from the latin phrase *Deus Ex Machina*, which refers to a plot device in Greek and Roman drama where an unsolvable problem was suddenly resolved by the arrival of a god. Over time, the phrase has evolved to signify any sudden, seemingly miraculous solution to a problem. The metaphor relates deeply with the thesis proposition: programming and AI, tools emerging from the "machina", offer designers not only creative possibilities but also a transformative, almost providential power to overcome technical hurdles.

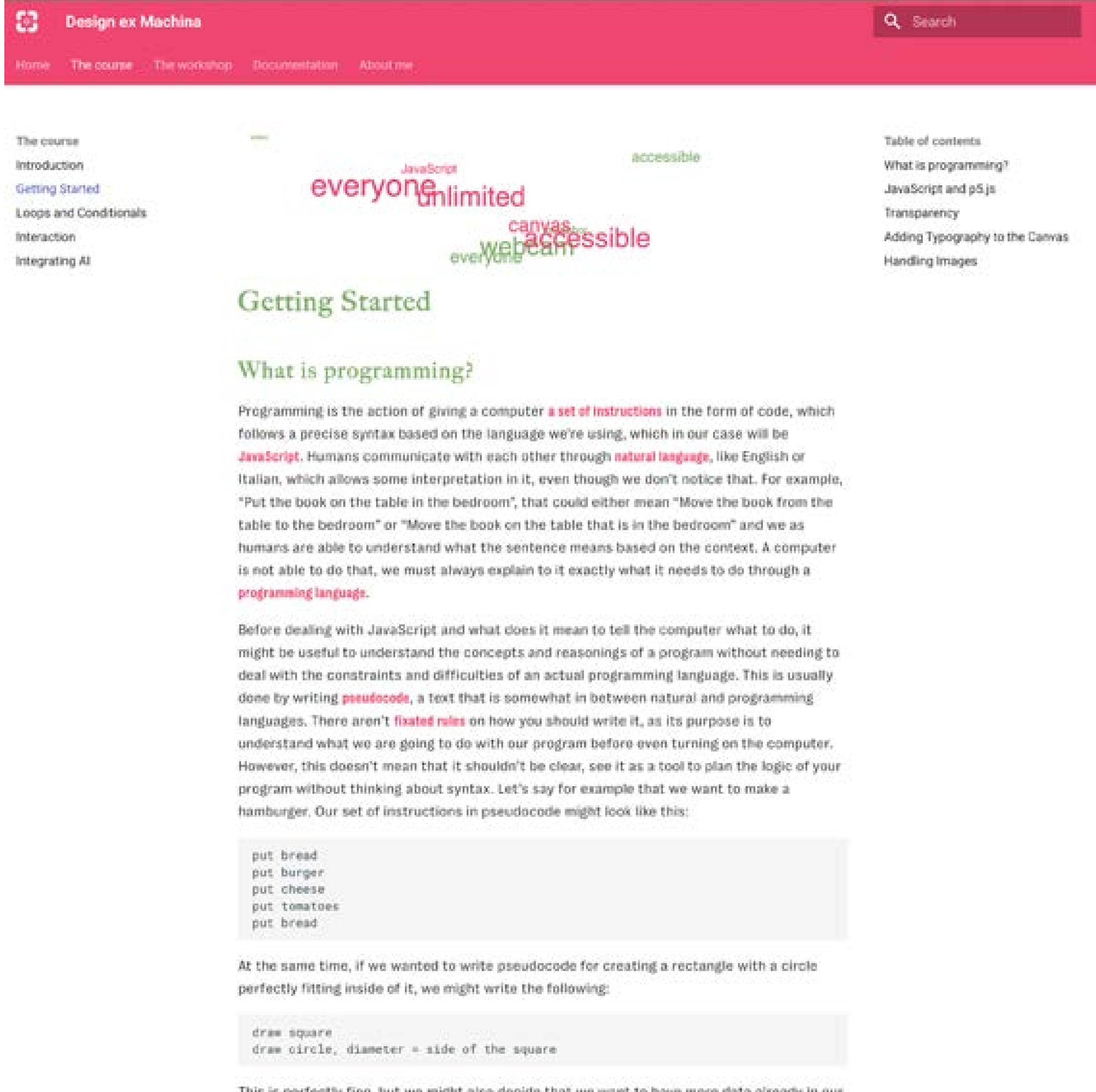
Therefore, Design Ex Machina was conceived as an openly accessible resource for any designer that desires to improve and expand their workflow through the use of programming and artificial intelligence, and it is designed to preserve this research methodology on the longer term, possibly proving useful for the more people as possible. The website was built with Materials for MkDocs, a static site generator optimized for markdown files, which allowed content to be easily organized and rendered as a navigable web interface without relying on any external resources for further customization. In fact, through the Materials extension it was possible to further modify and customize the website through additional CSS and inline HTML, for instance for typography, color schemes and embed p5.js runnable code.

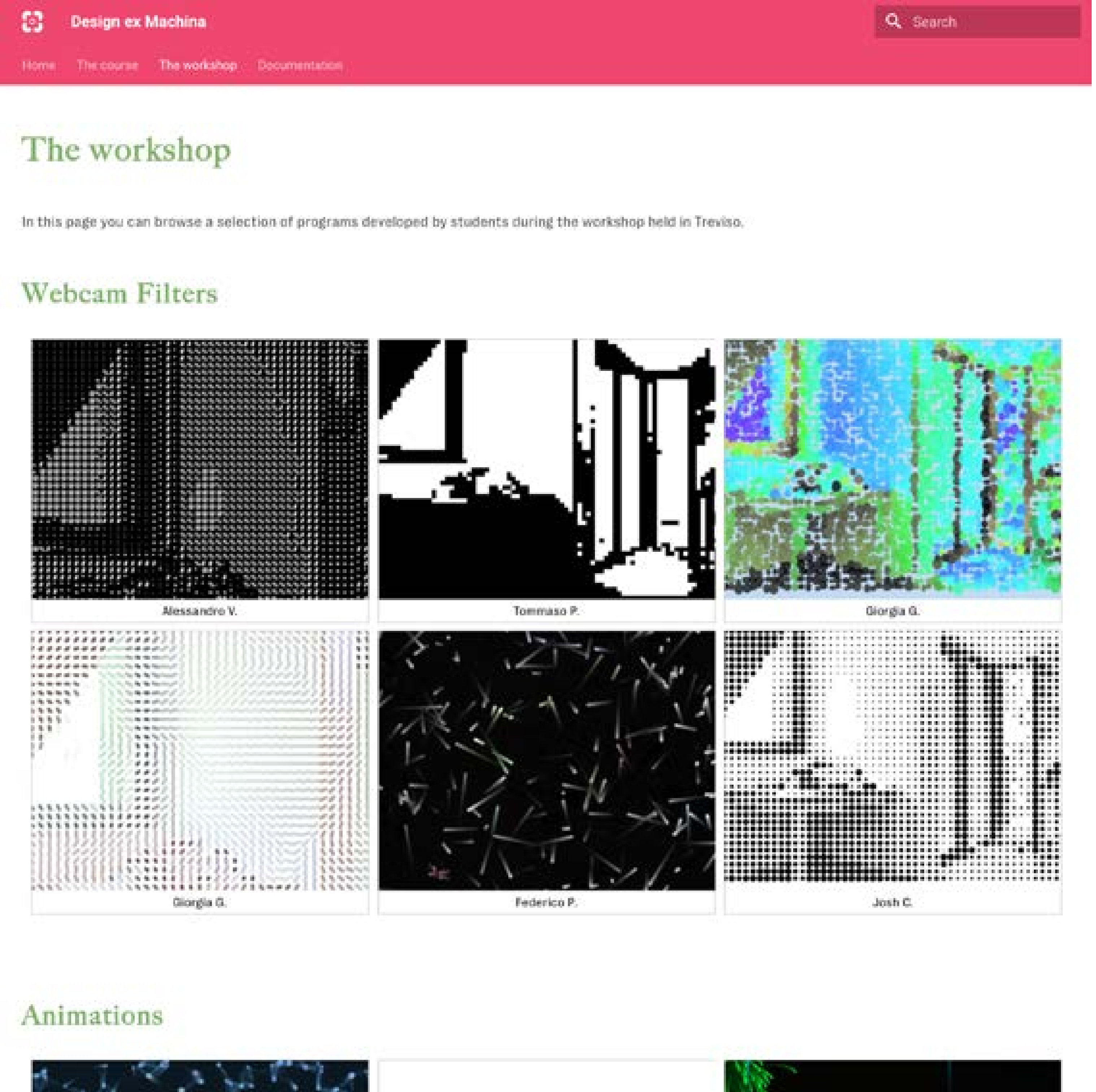
The manual's core is structured around progressive skill acquisition and firsthand experience with the tools, beginning with simple pseudocode explanations to introduce algorithmic logic, even before introducing the p5.js web editor interface.

The first sections focus on fundamental functions, such as `setup()` and `draw()`, as well as calls for shapes, text and color manipulation, providing functional code to be run at any time on the DEM website itself, but also summaries, exercises and animated explanations.

After this introduction, a section is dedicated to loops and conditional, a fundamental concept in programming, which serves to understand

On the left: Design Ex Machina Web Course, *https://dilidede.github.io/DesignExMachina/course/01gettingStarted/*





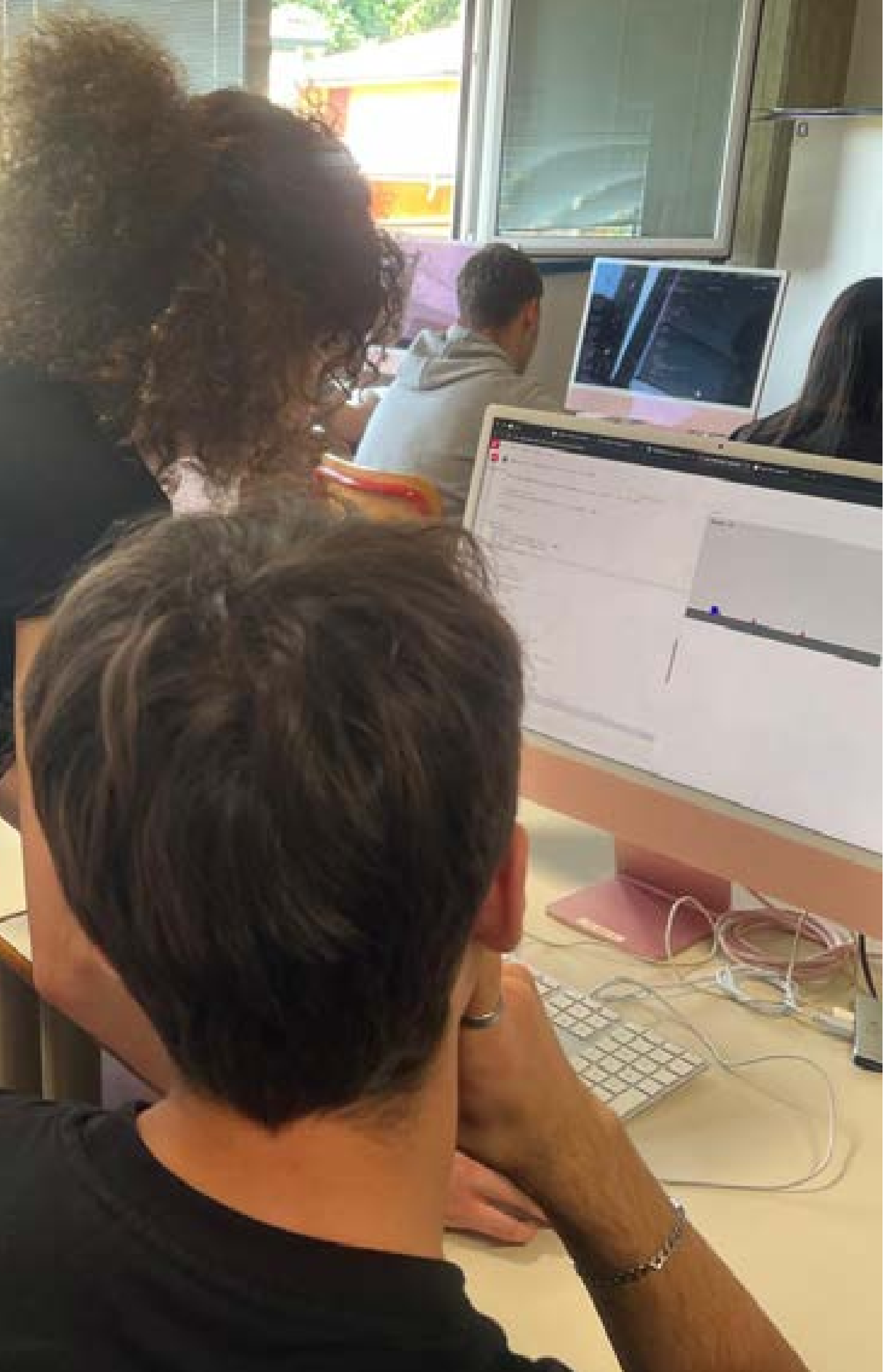
On the left: Design Ex Machina Web Course, *https://dilidede.github.io/DesignExMachina/course/01gettingStarted/*

the following section, which is dedicated to the interactivity of p5.js programs, for instance with the keyboard and the mouse, but also through the webcam, allowing students to realize projects and exercise that would not be feasible with a typical designer toolkit. This is the last section of the programming part of the manual, as the website does not aim to give an extensive knowledge on Processing, JavaScript or programming in general, but rather a base that allows for a critical and effective use of AI generated code.

The final module addresses AI integration, guiding students on the employment of LLM not only for code generation employed in our projects, but also as an additional learning tool.

Beyond the course, the website hosts two additional sections to complement the project ideals. The first one archives some of the works realized by the students of the workshop, cured thematically to engage the user to follow the course. For instance, a grid showcases variations of webcam filters, another one shows various animations kinds and so on.

The second section allows to freely read the present documentation through an online viewer, contextualizing the project within the broader thesis research.



First day of the workshop at IIS Palladio,
29 May 2025. Ph. L. Pavan

4. THE WORKSHOP EVALUATION

This section contextualize and evaluates the practical workshop conducted in the course of a week across May and June 2025 at the IIIS Andrea Palladio in Treviso, Italy. It was conducted as part of the investigation for the research to look at the practical application of our proposed methodology and to evaluate its efficacy, particularly on third-year students of the “Graphics and Communication” curriculum. Specifically, after the specific context exposition, the lecture structure is throughly presented, followed by a data evaluation and further observations upon the findings.

4.1. CONTEXT

At the beginning of May 2025, the *Istituto di Istruzione Superiore* (Secondary School Institute) Andrea Palladio of Treviso (Veneto, Italy) published a call for applications for an external expert and an external tutor to deliver eight-hour lessons to the third-year classes on the “Graphics and Communication” curriculum, focusing on the employment of artificial intelligence in visual and graphic design. As the opportunity suited the thesis research, a joint application with the thesis co-supervisor Rocco Lorenzo Modugno was submitted, which was successfully awarded.

For the preparation of the lesson it was crucial to consider its context, in particular the students educational background and their age group.

Moreover, another aspect to be taken in consideration was the language of the lesson, which was naturally held in Italian, as that is the language both of the students and the school.
Additionally, the Italian "Graphics and Communication" secondary school curriculum characterizes itself, among other aspects, for an extensive study on graphic design and visual communication and with the apprenticeship of graphic and design tools, often becoming accustomed to the Adobe Creative Cloud (Ministero dell'Istruzione e del Merito, n.d.). The workshop was held over the course of five days with an average of 22 students per day.

Ministero dell'Istruzione e del Merito. n.d. "Istituti Tecnici, Grafica e Comunicazione."

The second attribute to take in consideration was the age of the students, which averaged at 16 years. As statistical reports indicate that secondary school students exhibit diminished attention span due to the prolonged use of smartphones (Simsek, 2023), particular emphasis was placed on mantaining the students attention. Furthermore, as the institute had an interest in artificial intelligence image generation, the lesson also granted space to this aspect of AI's role in the visual design field. However, as the thesis focuses itself on the role of AI-aided coding, the data gathered and analyzed in this instance only concerns to this subject.

Simsek H. 2023. "The Effect of Smartphones Addiction on Attention Level in High School Students."

On the bottom: Directions and road signs as an example of algorithms during the lecture

Mappa con indicazioni per arrivare da Bolzano a Treviso in automobile

Bolzano – 39100

Prendi Via Roma/Romstraße fino a Via Innsbruck/Innsbrucker Str./ SS12
3 min (950 m)

Prendere A22/E45, SS 47 della Valsugana e SPV fino a Via Feltrina Sud/SR348 a Volpago del Montello. Uscire da SPV
2 hr 6 min (172 km)

Seguire la SR348 fino a Viale Monte Grappa a Treviso
19 min (15.4 km)

Continuare su Viale Monte Grappa fino a Porta Calvi
2 min (1.0 km)

Prendere Viale Cesare Battisti fino a P.za del Duomo
2 min (650 m)

Treviso 31100

Istruzioni per arrivare da Bolzano a Treviso in automobile

4.2. ORGANIZATION AND STRUCTURE

Following an analysis of the aforementioned context and factors, the course of action that was deemed as the most appropriate was to alternate regular theoretical explanations with practical exploration and distensive moments, in order to maximize the attention the students could give during the theory explanations.

As the students were more likely to be paying attention at the beginning of the workshop day, the first theory section was the most content-intensive. It began with the introduction of the concept of algorithms with familiar and simple examples, such as dance steps in a dance, instructions in a route and recipes. Then the difference between conventional algorithms and machine learning algorithms was introduced, accompanying with simple and clear examples. In order to gain a solid grasp of a variety of models explained, engaging examples were provided, such as by employing the example of how two different machine learning approaches, symbolic (or Good Old Fashioned AI) and connectionist artificial intelligence (or Neural Network), would classify an amousing photo of a dog dressed in the costume of an imaginary character. To emphasize that neither approach is intrinsically better than the other and to demonstrate how these aren't necessary novel technologies, outstanding examples of results using both symbolic and connectionist AI were presented, namely the MYCIN system (Buchanan and Freigenbaum, 1979) and Harold Cohen's AARON drawing machine (Mihály, 2021). Then the students were introduced to the concept of discriminatory models through the example of the MNIST Database and the difference between uni and multimodal generative models with applicable examples. At this point, the students were familiarized with the concept of generative adversarial networks (GANs), maintaining simplicity in the explanations. Here, the discriminator model was portrayed as a "bouncer" that could only allow cats to enter, and the generator model was portrayed as someone who attempted to disguise himself as a cat in order to fool the "bouncer".

Buchanan, B. G. and Freigenbaum, E. A. 1979. *Proposal to the Advanced Research Projects Agency for the Continuation of The Heuristic Programming Project*

Mihály H. 2021. "AARON.

Finally, a simplified flowchart of the Stable Diffusion models' image generation was provided, placing emphasis on the concept of the transition from noise to the reconstruction of a non-existent, therefore generated, high-quality image.
At this stage, the students were invited to begin utilizing Midjourney to observe the manner in which, given the similarity of the overall process, they could also notice how the images transitioned from blurred, low quality images to defined and precise pictures. As aforementioned this time was given to the students in order to revive their attention for the next theory section, as well as to satisfy their desire of beginning to use the tools promptly.

Following the acquisition of a certain degree of familiarity with a text-to-image model, the students were introduced to Memo Atken's *Learning to See* series. These artworks effectively illustrate not only the contrast between different human beings perception of the world, but also the way AI models can only "understand" what they previously encountered in their training data. For instance, the video shown to the students had a towel and some electronic cables interpreted by different models as sea, fire or flowers.
Following this brief digression, the lecture succeeds to explain how large language models work, with particular emphasis on their fundamentals as statistical models and therefore on the significance and risks that their subsequent hallucinations phenomenon can cause.

On the opposite page: Image from one of the pieces of Memo Atken's *Learning To See* series. AI-generated picture (left) based on the real video footage (right).

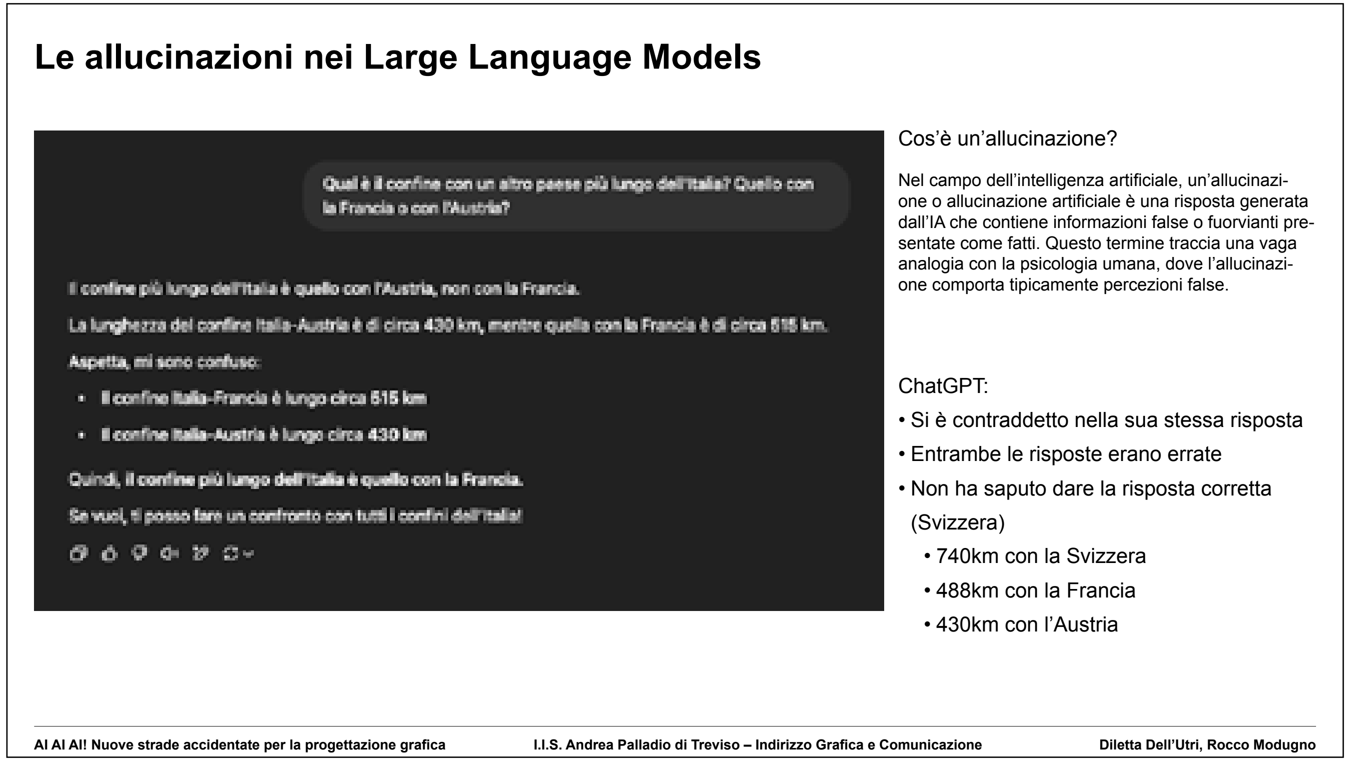
The lecture proceeds to mention different related topics related to the lesson, however due to the limited time of the workshop, no time was reserved for more substantial reflections. The first of these mentions is the question on the importance of knowing at least the basics of the topics presented, as this knowledge can lead to be more conscious parts of the political and social discussion. At the same time, students are invited to critically examine the extent to which an increasing number of companies have applied the "Artificial Intelligence" label to their products, often driven by market trends and branding strategies rather than by a real necessity or advantage derived from these technologies. Finally, this excursion concludes with various examples of generative art, for instance the Italian Arte Programmata art exhibition and the *Tentativo di formare dei quadrati invece che dei cerchi attorno a un sasso che cade nell'acqua* ("Attempt to form squares instead of circles around a stone falling into water") from Gino de Dominicis in 1968.

In the following section programming with p5.js was finally introduced through its interface and fundamental functions(`setup()` and `draw()`, following a brief theoretical overview of the difference between natural and programming languages. To stimulate students creativity and curiosity, a selection of example programs was presented. These included demonstrations of image manipulation and interaction design, and a

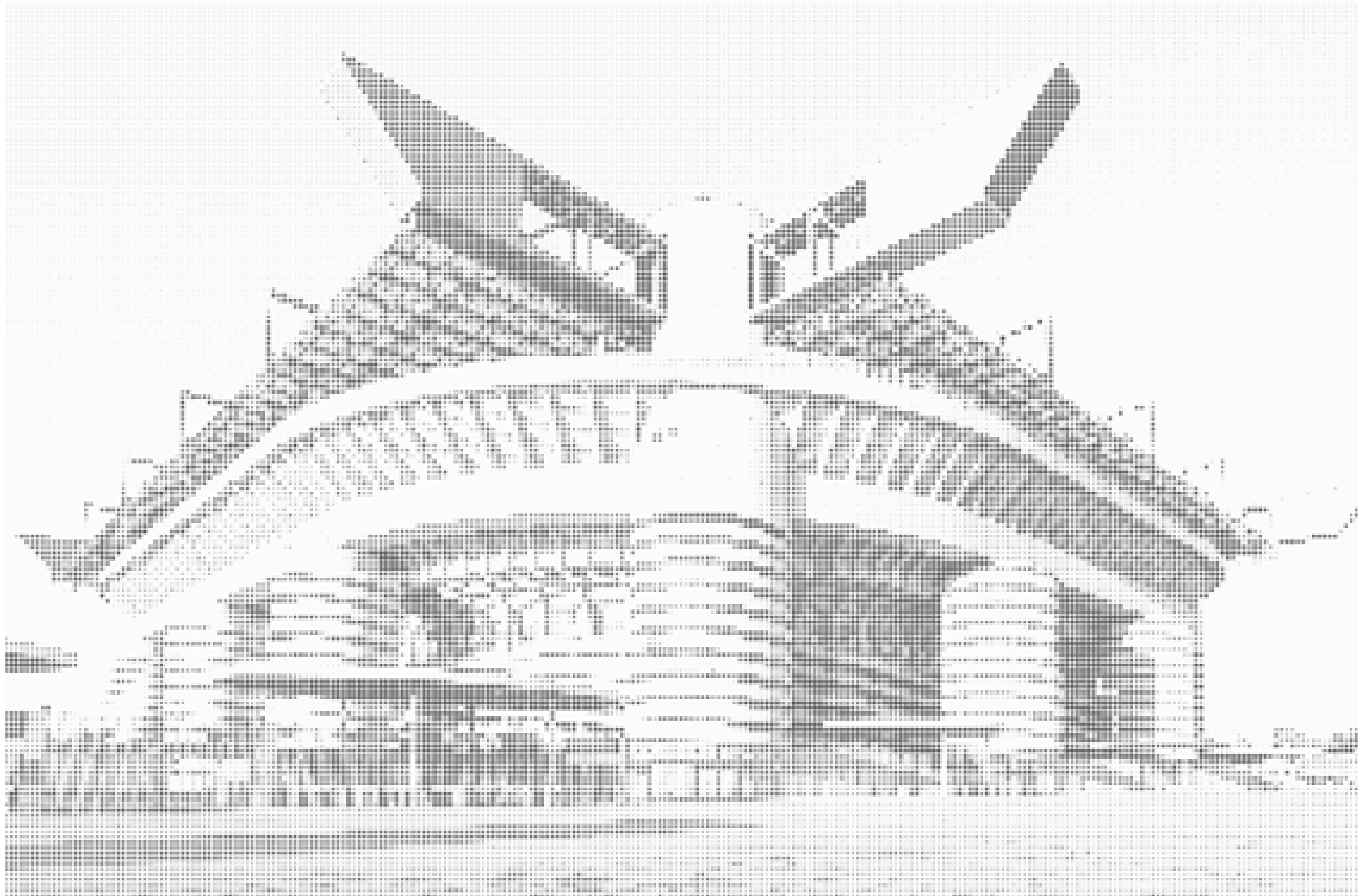
On this page: Slide with the explanation of hallucinations presented to the students.

On the opposite page: one of the effects achievable through p5.js presented to the students during the lecture.

variety of projects sourced from the online community, such as animations and games. Afterwards, students were encouraged to explore a broad range of other examples that could increase their interest and curiosity for the following part of the day.
Once the students had seen the potentiality of the p5.js library, they were introduced to the concept of AI aided coding as a mean for creating programs and tools that exceed the complexity a beginner could aim for. However, even though the potential of generative artificial intelligence code is enormous also outside the design field, it was pivotal to emphasize how these approaches cannot be used outside amateur projects that do not concern security and long term maintenance, at least not without being accompanied by solid programming knowledge and skill.



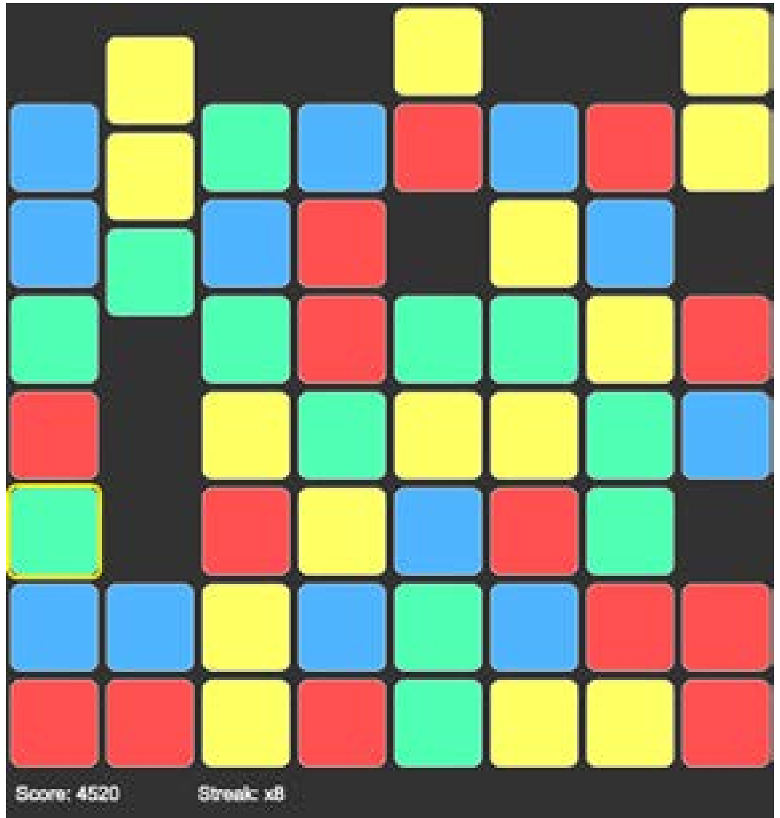
Ultimately, a slide with a clear scheme of simple prompt engineering for an AI chat-bot, in this case ChatGPT from OpenAI, both in Italian and English was shown. They were instructed on how the chatbots usually give better response to prompts written in English, therefore they were also directed to the use of online translators such as DeepL, in order to accommodate the possibility some of them would not feel comfortable in writing in English.
At this point they were left free for around a hour and a half to work with p5.js and LLM prompting.



Subsequently, a more thorough presentation of the Midjourney interface and its array of variables was made, in order for them to finally experiment with the software with greater control for a subsequent hour and a half.

After a pause the students were shown different resources for further exploration of programming tools, such as SonicPI, a programming language for creating music, as well as other Processing libraries and website courses. Finally, they were allowed to explore whichever of the tools seen they were more interested in, which could be from the workshop or from the final assets showcase.

At the end of the day, the students were invited to participate in an anonymous survey to evaluate the workshop itself, their own skills, and also to give feedback on their overall satisfaction.



On the opposite page: Conversation snippet and running program of a tile-matching game by Helena Z.

4.3. DATA GATHERING AND EVALUATION

During the workshop the students were asked to share a document file with links to their works on the p5.js Web Editor and to also attach the links to conversations had with the OpenAI chatbot. This way, the conversations could be reviewed and analyzed later to assess the workshop effectiveness, both in how they prompted the AI and in the quality of programs they created with its help. All the links shared by the students were then organized in a spreadsheet, dividing them by day and student, and by associating each conversation with a specific p5.js sketch when that was possible.

Each project was evaluated not by the code, but rather by the clarity of the prompts, whether the students read and understood the responses of the LLM, and also whether they demonstrated an active and reflective engagement with the model, as opposed to limiting the interaction to a single query and an uncritical application of the output.

However, the analysis of the material shared by the students evidently showed that there was a lack of foundations that often negatively affected not only the results, but also the sharing process itself.

While these evaluations can provide valuable insights on the teaching methodology proposed, its time limitations must be taken in account, as eight hours limits a deep technical immersion, potentially distorting outcomes toward simpler projects and more superficial interaction with the models. Future possible iterations of similar projects would definitely benefit from longer formats, which could potentially range from a week to an entire semester, to allow for deeper exploration, increased complexity, and more evident skill development. A similar observation can also be made on the class sizes, as an average of 22 students per day cannot guarantee a complete understanding of the topics by all the students.

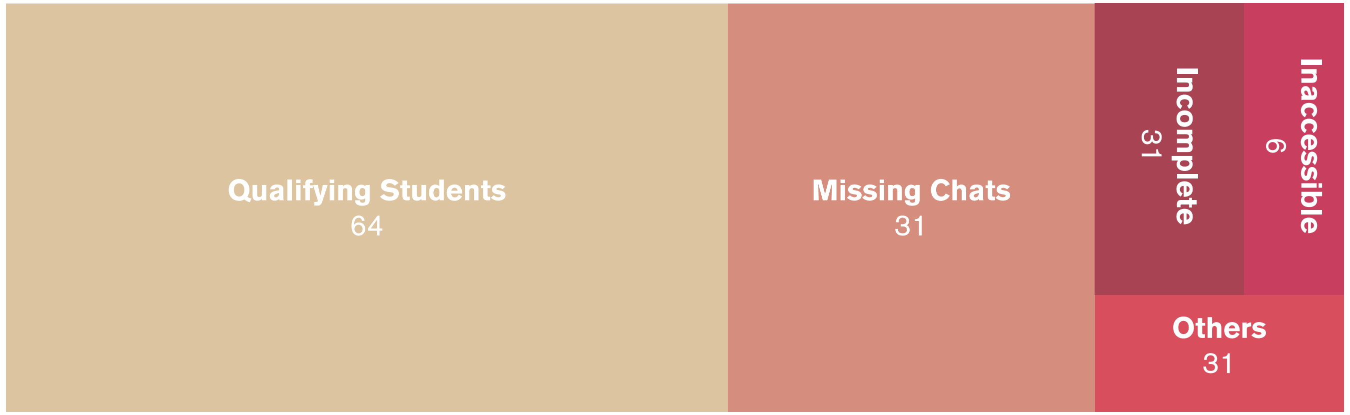
4.4. EVALUATION OF OUTCOMES

4.4.1. DATA POLISH AND DIGITAL LITERACY OBSERVATIONS

It became immediately apparent upon generating the spreadsheet from the students files, that not all data was complete or suitable for the evaluation process. For instance, an astounding 42.8% of the students did not successfully share all the links needed for a complete evaluation. In particular, issues were found with the conversations had with the chatbot related to the p5.js projects: 27.7% of the students did not manage to deliver any functioning ChatGPT link or a p5.js Web Editor project that related to the conversation had, as 26,6% did not have any conversation link on their document, 16,6% of the students shared only part of the conversations had in order to realize their projects, while around 3.7% and 0.9% respectively shared non functional links and only the “Chat-GPT canvas”, a section of the OpenAI chatbot that allows to analyze code more comfortably.

Particular attention should be given to the complete absence of links found in many students documents could be related to a variety of reasons, for instance it might come from minimal engagement on the class and consequently a perceived irrelevance of sharing the conversations. Another reason might be a focus from the students on the JavaScript outputs rather than the process itself, considering the ChatGPT process as secondary or optional, while in the scope of our research it was arguably more important than the program itself. Because of these reasons the incomplete data, such as unmatched Processing projects and LLM conversations, has been discarded, resulting in 57% of the total students that provided valid results for the evaluation.

Moreover, 16% of the students in the first two days shared the same p5.js web editor link more than once, often labeling it in their document each time as a completely different project. Already during the lectures it was clear that this could be an issue, as it was noticed that some students were saving a project and substituting the code with new one, presumably convinced that this way they would be stored



Above: Tree map graph of student submissions categorized by their compliance with the qualifying criteria for evaluation.

as different “Sketches” in the p5.js Web Editor built-in drive system, since the students often acted astonished when the error was shown to them. Because of this reason in the following days the students were invited to share the code directly on the documents. During the lecture an observation was also made on how the students often shown little familiarity with computer basic concepts, such as the management of files and navigation of multiple browser tabs.

These points strongly indicate that many students may lack fundamental digital and computer skills, as they’ve demonstrated struggles with understanding basic digital concepts. While it is also suitable to assume that other factors may have influenced these outcomes, it is also true that the patterns emerged align themselves with a broader problem of digital illiteracy among students. These observations align themselves with the 2023 Survey of Adult Skills from the Organization for Economic Cooperation and Development (OECD, 2023), a research conducted in its members states to analyze different abilities of the population between 16 and 65 years old. The study reported that in Italy young adults aged 16 to 24 scored on average below the OECD average in digital literacy. Furthermore, 34.7% of the Italian population across all ages performed proficiency at Level 1 or below on the five-level scale.

OECD. 2023. OECD Skills Outlook 2023: Skills for a Resilient Green and Digital Transition.

Caponera E. and Di Chiacchio C. 2023. *Indagine sulle competenze digitali. Rapporto Nazionale.* INVALSI

Another data in this regard is the ICILS international study to which Italy took part in 2023 for the second time with 152 schools in the whole country, in order to measure the computer and informational literacy, evaluated in points that correspond to 4 levels. The Italian CIL average points is 491, just two points below the minimum requirement for Level 2, therefore qualifying itself in the lower part of the evaluation grid, yet still higher than the international average result (Caponera and Chiacchio, 2023).

On the opposite page: Program with a set of effects on webcam input by Alessandro V.

The information collected during our workshop, when viewed in the larger context of nationwide low digital literacy among young people, prompts important questions about the root problems in the Italian education system, in particular how computers are often seen more as “dangers” to students attention and how the attempts to make the school more digitally aware limit themselves to the implementation of interactive whiteboards in classrooms, rather than to address the digital literacy issues identified by the relevations. Additionally, new tools and technologies can be truly understood and used with knowledgeable, critical judgment. Otherwise, this lack of digital literacy might lead to an inability to use them effectively or, arguably more dangerously, to uninformed decisions and vulnerability to manipulation.

4.4.2. BARRIERS TO EFFECTIVE USE OF THE LLM

At a first analysis of the conversation links shared by the students, but also during the workshop itself, observations on the prompts effectiveness were made. Although we did not make mandatory the use of English for the LLMs, we strongly recommended it as we explained to the students that since LLMs are mostly trained on English text, they are proved to perform worse when interacted in languages less present in the training data (Thelmann al., 2023). As we anticipated some of the students might feel insecure with their linguistic skills, we provided them a link to DeepL, an effective AI based language translation tool. Once the conversations were read, it was found that 48,4% of the students wrote to ChatGPT in English, a data that aligns itself with the revelations made on the final form, as 37,6% of the students was confident with their English skills, ranking it 4 or 5 on a scale from 1 to 5. However, a review of the English conversations shown that 28% of the total students showed notable grammatical or spelling errors.

Thellmann, K. et al. 2023. Towards Multilingual LLM Evaluation for European Languages.

Moreover, this data aligns with the EF English Proficiency Index Report, according to which young adults in Italy score an average of 500 points out of 700, which is just over the low proficiency band (EF, 2024). In fact, as the lectures took place with third year students of a technical institute, they are supposed to reach a B2 level of English at the end of their high school education, however it was reported by the Italian *National Institute for the Evaluation of the Education and Training System* (INVALSI) that only 50% of them appearently do (INVALSI, 2024), which raises questions and doubts not only regarding the possibility of Italian young adults to be active members of a broader globalized society, but also their ability to interact with these artificial intelligence tools, as well as with the wider field of informatics, as they often require a higher English level to be efficiently approached.

Education First. 2024. EF English Proficiency Index. A Ranking of 116 Countries and Regions by English Skills.

Istituto Nazionale per la Valutazione del Sistema Educativo di Istruzione e Formazione. 2024. Rapporto prove INVALSI 2024.

With the analysis of the ChatGPT prompts, an interesting yet arguably preoccupying pattern emerged: many students, after they received an answer from the OpenAI conversation bot, limited themselves to copy and paste the code on the p5.js Web Editor, apparently without reading the considerations and notes generated by the LLM. For instance, many times the code presented placeholder names for files that the student were supposed to upload, and eventually change either the file name in the code or the file name itself, but an alarming 23% of the students did not do that. This is deducible by the fact that many replies to this programs with placeholder names lamented a supposed non functionality of the code, even though when tested with the AI proposed edits, they did not show particular weaknesses. This pattern seems to suggest a superficial engagement with the LLM guidance, prioritizing the code extraction over its comprehension and engaged implementation.

Another interesting pattern is that students achieved better results, even when working on simpler projects, by gradually building them through iterative prompts rather than attempting to achieve complete solutions in single requests.

Below: First day of the workshop at IIS Palladio, 29 May 2025. Ph. L. Pavan

4.5. APPLICATIONS DEVELOPED BY THE STUDENTS

Regarding the programs themselves, the analysis of the projects revealed interesting patterns in the programs chosen to develop. They predominantly fell into two categories: videogame clones (55%) and interactive visual experiments (40%), with utilities consisting of the remaining 5%.

Within the videogame programs, they mostly included arcade classics, for instance variants of Tetris, Pac-Man and Space Invaders; but also video game versions of Tic-Tac-Toe. This reliance on familiar structures suggests a curiosity of the students to see the potentiality of the technique without focusing on developing new ideas. At the same time, as these games have been widely well-documented and recreated through the years, it simplified the prompt engineering required to obtain functioning results, and therefore also indirectly mitigated the aforementioned language barriers. For instance, a prompt such as “Create a Snake game” would be more easy to write, compared to a more thorough description of original mechanics. However, it should not be excluded that this choice could come from a limited ambition in developing new mechanics.

During the later days the students were encouraged to experiment with interactive visual projects, successfully obtaining a higher degree of experimentation. These projects mainly consisted of webcam filters, random walkers and flock mechanics, including implementations with ASCII visualization and perlin noise.

However, regardless of the type of program developed, the success of the programs strongly correlated to the prompt precision. Conversations that featured vague or unclear prompts from the students were frequently malfunctioning or not following the students intentions, such as a Snake game without a grid or horizontal side scroll platformers with collision errors.

On the opposite page: Random Walk based on live webcam input by Federico P.

4.6. SUMMARY AND REFLECTIONS ON WORKSHOP OUTCOMES

Overall, two different approaches in the use of the LLM emerged: on one hand, we observed a part of students who approached the language model as a collaborative tool, and thus achieving more complex and functional outcomes (e.g. a *Brawl Stars* clone refined through 16 separate prompts). On the other hand a part of them approached the model as an infallible source, expecting to immediately achieve a perfectly functional program and copying outputs without scrutiny, thus often coming across technical or logical errors. The prevalence of the second approach, which also generally led to less original outcomes, arguably suggests that without an explicit invite for reiteration and analysis, the students adopted a mechanical application of the AI-generated code. This reflection aligns with the original concerns on the disengagement that students might have when presented with technical difficulties, rather than attempting to debug the code with the AI assistance.

These patterns intersected critically with the two barriers indentified earlier. First, the digital literacy gaps manifested starkly, also during the workshop itself when some students struggled to realize that substituting the code of a p5.js web editor file, then saving, was not equal to creating a new file, which arguably reflects a limited understanding of basic computer principles, despite their regular use of them as graphic designers. As this literacy gaps are unfortunately a national trend, the causes should not be identified with specific limitations of single schools or school types, but rather in the way the Italian scholastic system regards computer literacy (Carbone and Calvi, 2024).

Additionally, the students projects demonstrate that while students were able to use LLMs to produce functioning code, their ability to use the model to produce original ideas through effective prompting should be better addressed in similar projects. The difficulties with the English language or its absence did not preclude success in the projects where more precise descriptions were submitted as a prompt. Thus, although English queries consistently outperformed Italian ones in code accuracy, the core challenge in similar educational settings

On the opposite page: Black and white pixelated webcam effect by Tommaso P.

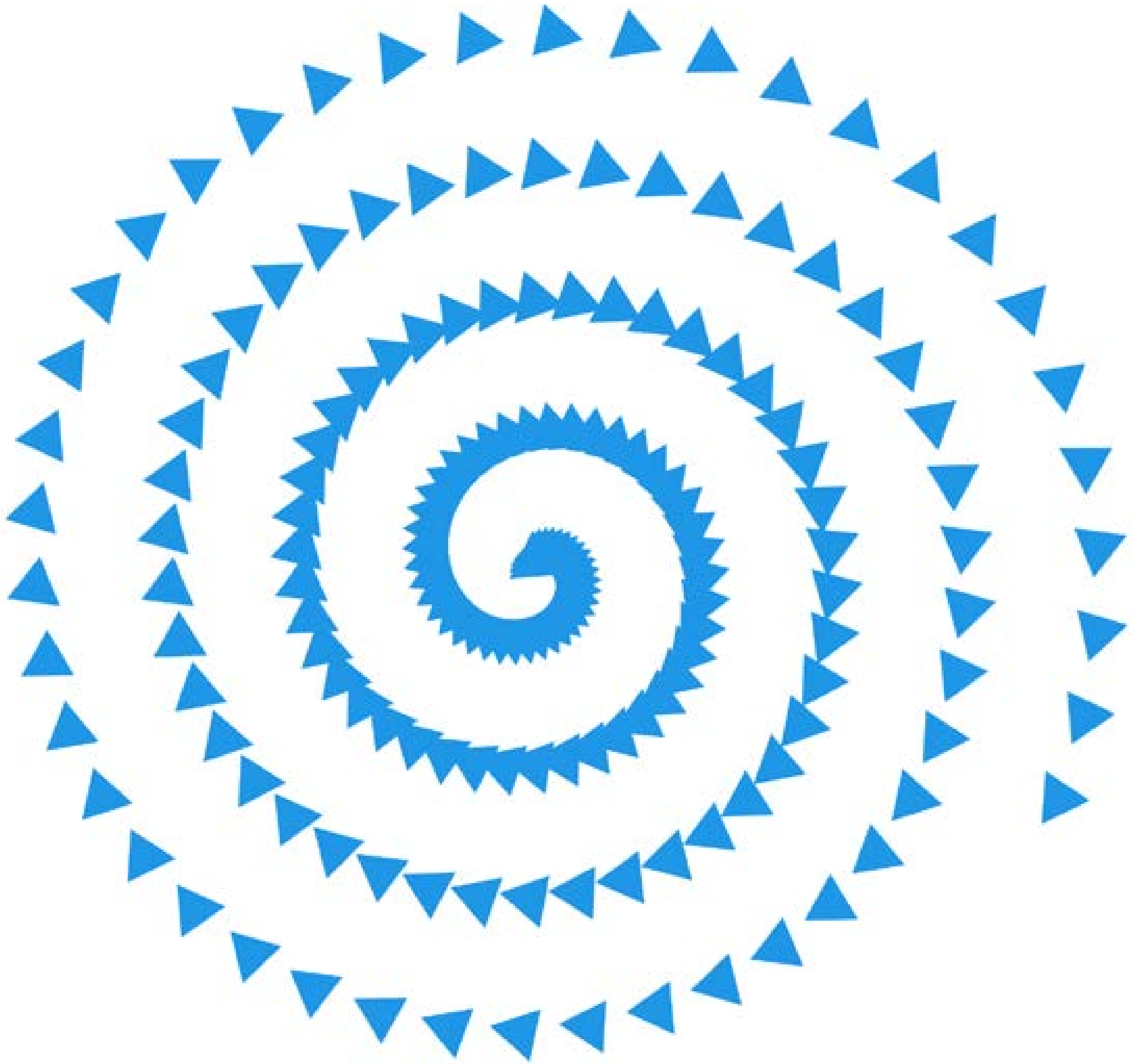
On the this page: ChatGPT Conversation for a spiral made of triangles that can separate and reorder through keyboard interaction
On the opposite page: The running program by Giorgia G.

should extend to foster an algorithmic approach in the use of the AI and in a deeper understanding of its functioning, in order to treat it as a tool that works as well as the prompt clarity is, and not as an oracle.

Finally, the data aggregated from the post-workshop survey reveals that the perception and appreciation of the workshop was generally positive. While only a quarter of the students considered their knowledge sufficient to follow the workshop, only 13% of the students found the machine learning and algorithmic topic difficult (i.e. gave a 4 or 5 to the question "Did you find the topic difficult?"). Additionally, 96,4% of the students rated their understanding of "how to use best these technologies" between 3 and 5 in a scale from 1 to Finally, as seen by the great popularity that the OpenAI Chatbot has today (Google Trends, 2025) it does not come as expected that 98,1% of the students is planning to continue using LLMs, it is a pleasant surprise that the vast majority of the students (79,4%) also intends to continue working with p5.js.

Google Trends. "ChatGPT and DeepSeek Search Compariso1n." Accessed 20 April 2025.

Within the span of eight hours, the workshop managed to not only familiarize students with LLMs and creative coding using p5.js, but also in achieving actual interest and participation. Despite varying levels of digital literacy and mixed initial interest, the fact that nearly all of the participants plan on continuing to explore AI tools, and a vast majority expressed intention to continue using p5.js, is arguably a testament to the effectiveness of the format. These results suggests that even brief yet well crafted exposure can spark interest and push students to explore new technologies. This outcome is particularly encouraging for future educational projects aiming to make emerging technologies more approachable and applicable.



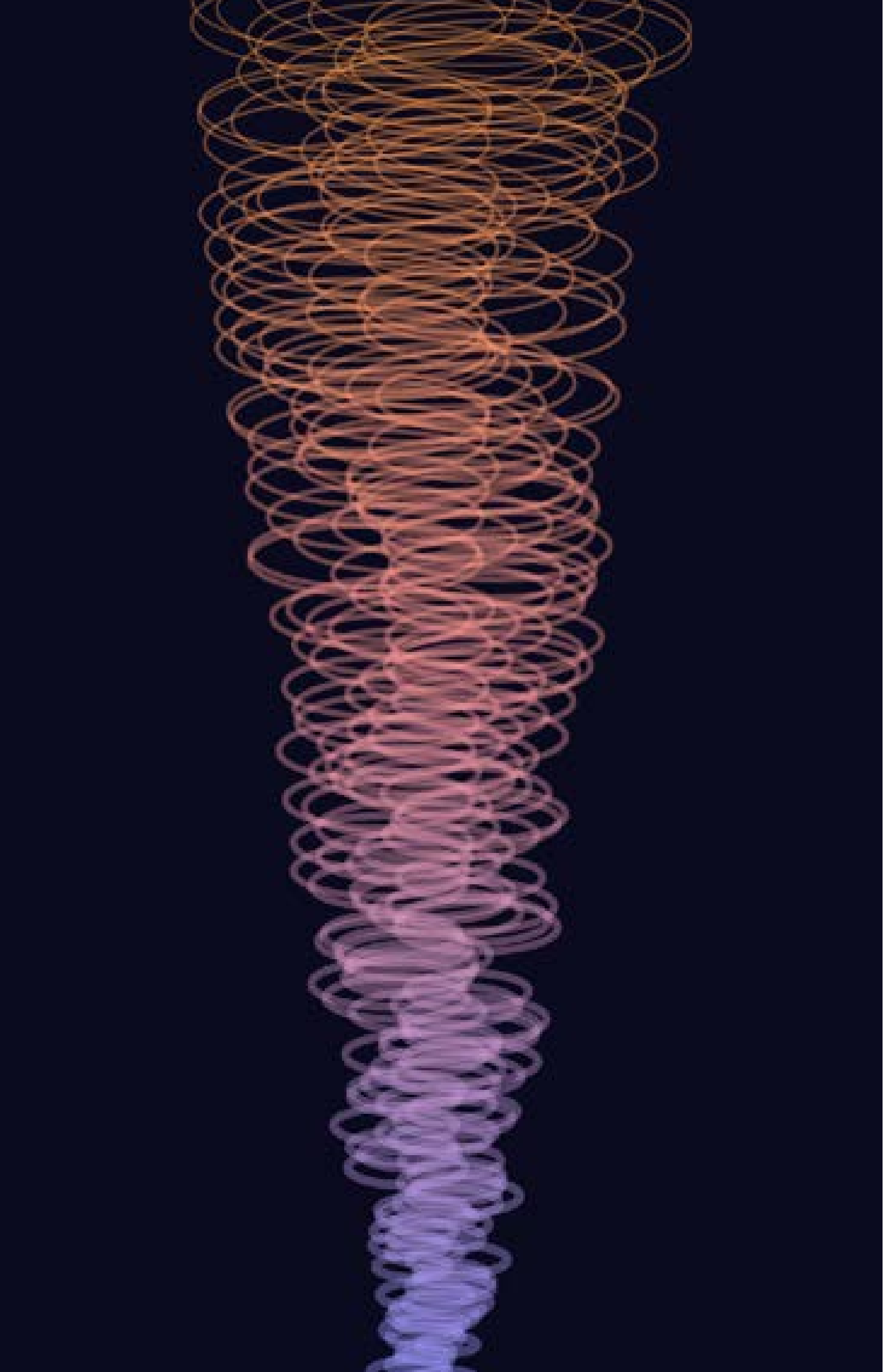


Image from a rings animation by Beatrice M.
June 2025

5. **CONCLUSION**

This research began by posing the question on how can the teaching of algorithmic and AI tools expand designers' creative agency. This exploration of the integration of creative coding and artificial intelligence within design education highlights a central point: technological tools enhance creativity only when approached with critical engagement. Through the workshop held in Treviso it was demonstrated how open source platfroms and models can facilitate algorithmic expression, making it possible for designers to create interactive experiences and unique visual outputs, emancipating themselves from software costraints.

Concurrently, large language models (LLMs) emerged as collaborative agents, expanding the possibilities for novice programmers. However, this potential is accompanied by significant challenges: uncritical replication of AI-generated code, linguistic barriers in prompt formulation, and disparities in digital literacy that transform otherwise straightforward tasks into substantial obstacles.

Our findings suggest that creativity in the digital age flourishes not through passive adoption of computational tools, but through informed and reflective interaction with them. This includes understanding the mechanisms behind generative models, interrogating the ethical implications of training datasets, and cultivating a healthy skepticism towards machine-generated outputs.

John Frazer's cautionary perspective remains profoundly relevant: design should not be dictated by the tools employed.

Looking forward, design pedagogy must evolve into a dialogical practice that integrates code and craft. This could involve semester-long curricula aimed at cultivating algorithmic sensibility, the development of inclusive tools adapted to multilingual contexts, or co-creating frameworks alongside students. However, it should be noted that such reflections do not limit themselves to the design field, but rather could be expanded to other domains, as digital literacy and AI employment have a beneficial potential that exceeds the design niche.

Ultimately, the objective is not merely to incorporate AI into design processes, but to cultivate a generation of designers capable of critically and creatively shaping the technological landscape. The "Design Ex Machina", then, is not about machines delivering transcendent solutions, it is about emancipating designers.

GLOSSARY

Algorithmic Thinking: Breaking down tasks into clear, step-by-step instructions, thinking like a computer.

Computational Design: Design driven by algorithms, code, or data—often used in architecture, art, and product design to explore complex forms or behaviors.

Creative Coding: Using programming as a tool for artistic expression: creating visuals, music, or interactive media through code.

Generative Adversarial Network (GAN) A network with a GAI that generates data and another data tries to detect if the content is generated.

Gaussian Mixture Models (GMM): A way to model and analyze data by assuming it's made up of overlapping clusters with bell-shaped (Gaussian) curves.

Good Old-Fashioned Artificial Intelligence (GOFAI): Traditional AI based on explicit rules and logic rather than learning from data.

Hidden Markov Model (HMM): A method for predicting sequences where the system has hidden (unseen) states—used in speech recognition, music generation, etc.

Large Language Model (LLM): An AI that understands and generates human-like text by learning from massive amounts of written content.

Machine Learning (ML): A way for computers to learn patterns from data and improve at tasks without being explicitly programmed.

Markdown: A simple way to format text using plain characters—used for writing documents, readmes, or blog posts (e.g., **bold** becomes bold).

Neural Network: A computer system inspired by the human brain—made of layers of nodes that learn to recognize patterns in data.

Programming Languages:

High-Level PL: Simpler languages like Python or JavaScript.

Visual PL: Use blocks or graphics instead of typed code (e.g., Scratch, Grasshopper).

Library: Pre-written code to optimize the program writing.

Syntax Extension: Adding new rules or features to a programming language.

HTML: Builds the structure of web pages.

CSS: Styles web pages—controls colors, layout, fonts, etc.

Variational Autoencoders (VAE) An AI model that learns to compress and then recreate data.

BIBILIOGRAPHY

01. European Union, European Parliament and European Council. 2024. *Artificial Intelligence Act, Regulation 2024/1689*. Adopted 13 June 2024.
02. Frazer J. 1995. *An Evolutionary Architecture*. London: Architectural Association Publications.
03. Lee N. 2014. “From a Pin-up Girl to Star Trek’s Holodeck: Artificial Intelligence and Cyborgs.” In *Digital Da Vinci, Computers in the Arts and Sciences*. Springer.
04. O’Hanrahan E. 2018. “The Contribution of Desmond Paul Henry (1921–2004) to Twentieth-Century Computer Art.” *Leonardo* no. 51: 156-162. https://doi.org/10.1162/LEON_a_01326
05. Noll A.M. 2016. “The Howard Wise Gallery Show of Computer-Generated Pictures (1965): A 50th-Anniversary Memoir.” *Leonardo* no. 49: 232-239. https://doi.org/10.1162/LEON_a_01158
06. Noll A.M. 1994. “The Beginnings of Computer Art in the United States: A Memoir.” *Leonardo* no. 27: 39-44.
07. Sutherland I.E. 1963. “Sketchpad. A Man-Machine Graphical Communication System.” In *Proceedings - Spring Joint Computer Conference*, no. 23.
08. Cohen, P. 2017. “Harold Cohen and AARON.” *AI Magazine* 37, no. 4: 63-66. https://doi.org/10.1609/aimag.v37i4.2695
09. Malloy, J. 2003. “Algorithmic Art, Scientific Visualization, and Tele-immersion.” In *Women, Art, and Technology*. MIT Press.
10. O’Regan G. 2015. “Xerox PARC.” In *Pillars of Computing. A Compendium of Select, Pivotal Technology Firms*. Springer. https://doi.org/10.1007/978-3-319-21464-1_35
11. Mazzone M. 2020. “Andy Warhol: Computational Thinking, Computational Process.” *Leonardo* no. 53: 179-182. https://doi.org/10.1162/leon_a_01574
12. Roose K. 2022. “An A.I.-Generated Picture Won an Art Prize. Artists Aren’t Happy.” *New York Times*. https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html
13. Fry B. and Reas C. 2007. *Processing. A Programming Handbook for Visual Designers and Artists*. MIT Press.
14. Mishra P. and M.J. Koehler. 2006. “Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge.”

Teachers College Record V. 108, no. 6: 1017-54.
15. Selby C. 2011. "Four Approaches to Teaching Programming." Paper presented at Learning, Media and Technology: a doctoral research conference.
16. Google Trends. "ChatGPT and DeepSeek Search Comparison." Accessed 20 April 2025. https://trends.google.it/trends/explore?geo=IT&q=chatgpt,deepseek&hl=it
17. Moravec V. et al. 2025. "Environmental footprint of GenAI – Changing technological future or planet climate?" *Journal of Innovation & Knowledge* no. 10. https://doi.org/10.1016/j.jik.2025.100691
18. Ministero dell'Istruzione e del Merito. n.d. "Istituti Tecnici, Grafica e Comunicazione." Accessed June 2, 2025. https://unica.istruzione.gov.it/portale/it/orientamento/guida-alla-scelta/dal-sistema-integrato-0-6-anni-al-secondo-ciclo-di-istruzione/uola-secondaria-di-secondo-grado/tecnici/grafica-e-comunicazione
19. Simsek H. 2023. "The Effect of Smartphones Addiction on Attention Level in High School Students." *Journal of Dependence* 24, no. 2: 113-22. https://doi.org/10.51982/bagimli.1118921
20. Buchanan, B. G. and Freigenbaum, E. A. 1979. *Proposal to the Advanced Research Projects Agency for the Continuation of The Heuristic Programming Project*. Stanford University.
21. Mihály H. 2021. "AARON." In *Encyclopedia of Artificial Intelligence: The Past, Present and Future of AI*, edited by P. L. Frana and M. J. Klein, 1-2. Santa Barbara: Bloomsbury.
22. OECD. 2023. *OECD Skills Outlook 2023: Skills for a Resilient Green and Digital Transition*. Paris: OECD Publishing. https://doi.org/10.1787/27452f29-en
23. Caponera E. and Di Chiacchio C. 2023. *Indagine sulle competenze digitali. Rapporto Nazionale*. Roma: Istituto Nazionale per la Valutazione del Sistema Educativo di Istruzione e Formazione.
24. Thellmann, K. et al. 2023. *Towards Multilingual LLM Evaluation for European Languages*. TU Dresden. https://doi.org/10.48550/arXiv.2410.08928
25. Education First. 2024. *EF English Proficiency Index. A Ranking of 116 Countries and Regions by English Skills*. Zurich.
26. Istituto Nazionale per la Valutazione del Sistema Educativo di Istruzione e Formazione. 2024. *Rapporto Nazionale prove INVALSI 2024*. Roma.
27. Carbone D. and Calvi C. 2024. "The Digitisation of Italian Schools and the Pandemic Trigger: Actors and Policies in an Evolving Organisational Field" In *Societies* 14, no. 6: 94. https://doi.org/10.3390/soc14060094

ACKNOWLEDGEMENTS

I am especially grateful to Prof. Antonino Benincasa, who first introduced me to the integration of artificial intelligence in the design workflow during my first semester at the Free University of Bolzano, and who was open to pursuing a more theoretical approach in this thesis.

Sincere thanks also go to my co-supervisor, Rocco Lorenzo Modugno, who found the opportunity to work at IIS Palladio and encouraged me to take on the rewarding challenge of teaching in a real school environment.

A special aknowledgement goes to Laura Pavan and Marco Trentin, the IIS Andrea Palladio teachers who believed in showing these tools to young designers and who supported us during the days spent in Treviso.

Finally, my deepest thanks go to my parents, whose constant support and sacrifices allowed me to study away from home. Without them, this work would never have come to life.