Inclusive Coding Perspectives: Youth Insights From Trento

1st Francesca Fiore
Department of Computer Science and Engineering)
University of Trento, Italy
francesca.fiore@unitn.it

2nd Alberto Montresor
Department of Computer Science and Engineering)
University of Trento, Italy
alberto.montresor@unitn.it

***Abstract*—This paper presents an exploratory sociological research study focusing on the perceptions and awareness of coding among youth aged 16 to 24 in Trento, Italy, with an emphasis on gender inclusivity. It combines theoretical and practical perspectives to understand the role of *coding* in today's knowledge society, focusing on gender dynamics in coding education. Using qualitative methods such as interviews and focus groups, the study involved a gender-balanced mix of students and educators from various backgrounds. It centers on four themes: coding knowledge among youth, the benefits of coding, the effectiveness of coding education, and the role of gamification in enhancing coding awareness. The findings highlight significant gender differences in interest and participation in coding, stressing the need for inclusive teaching methods like gamification to promote coding literacy across genders. The paper underscores the importance of gender inclusivity in computer science education to enhance future digital competencies and employability.**[1]

***Index Terms*—Student-centered Learning Environments; Women for Leadership in Engineering Equity, Diversity, and Inclusion; Future-oriented and Personalized Educational Concepts.**

I. Introduction

Given the well-recognized importance of coding for newer generations within the evolving framework of the knowledge society, we undertook an exploratory sociological study in 2022 in Trento, Italy. This paper provides an overview of such research study, which engaged students from the Qualitative Research courses in the Sociology Department at the University of Trento, where one of the authors served as a teaching assistant for five years.

Our investigation particularly centered on young people's familiarity with coding and their understanding of its significance for their education–not just in terms of hard skills but also concerning vital transversal competencies. We specifically targeted individuals aged 16 to 24, who were either enrolled in secondary schools or pursuing university courses in Trento. By adopting a diverse approach in terms of educational background and gender, our study included both computer science experts and novices, along with educators and trainers dedicated to fostering coding awareness and skills. This methodology allowed us to capture a more accurate picture of coding perceived value among young people currently on educational paths and to collect insights from professionals about the critical role of computational thinking.

From a structural point of view, the paper is organized into two main sections. Section II details the methodology employed for participant selection, data collection, and analysis, highlighting the criteria for participant inclusion and the challenges faced during data collection. Section III lays out the findings of the research, categorizing them into four main themes: the prevalence of coding knowledge among young people, the benefits of coding education, the limited effectiveness of traditional school teaching in coding, and the role of gamification in promoting coding awareness. Each theme addresses a specific research question, thoroughly analyzing the data derived from interviews and focus groups.

II. Research methodology

A. Methodology and planning

Our decision to explore young people's knowledge of coding and the value they attribute to it was motivated by both an observed phenomenon of interest [1]: the age of knowledge has elevated coding to a highly demanded skill, culminating in its recent integration as a mandatory part of education at all levels [2]. This increasing significance of coding is further underscored by children's literature aimed at introducing computational thinking basics, as well as the surge in associations, workshops, training days, academic articles, and newspaper articles dedicated to coding.

Building on a review of existing literature [1], our research design was shaped by the confirmed significance of coding for young individuals, highlighting both hard skills and transversal competencies that are valuable in the labor market and beyond. Intrigued by the focus on coding education for children and pre-adolescents, we aimed to determine whether students in Trentino are aware of coding and its relevance to their education. Our goal was to either raise awareness about coding if it was previously unfamiliar or to explore effective teaching methodologies if already recognized.

[1]Acknowledgments: We would like to extend our gratitude to the students from the Qualitative Research courses in the Sociology Department at the University of Trento, under the guidance of Prof. Chiara Bassetti and Prof. Vincenzo D'Andrea, for their contribution to this research: Alice Conati, Giulia Fasoli, Irene Negri, Alessio Roat, Nadia Scatola, and Filippo Tomezzoli. Their engagement in conducting the interviews and coding the results was instrumental to the success of this study.

To comprehensively address these research questions, we adopted a qualitative methodology, employing a mix of semi-structured interviews and a focus group.

This approach enabled us to collect data that offered a comprehensive and descriptive insight into the phenomenon under study, moving beyond mere exploratory observations or the constraints of limited statistical models. Through our interviews, we aimed to explore the educational and career choices of individuals, understanding their motivations. Furthermore, we aimed to assess how coding is taught in Trento's secondary schools, evaluating students' interest and proficiency. We engaged with three distinct groups:

- 16 to 24-year-olds with minimal coding knowledge and no related education;
- 16 to 24-year-olds proficient in coding and enrolled in technology or IT programs;
- educators of all ages who are involved in coding instruction in schools, extracurricular settings, or universities.

In addition, we organized a focus group with university students, encompassing both those experienced and those inexperienced in coding, to foster dynamic discussion.

The selection of interviewees and focus group participants was influenced by several criteria. Primarily, we aimed to ensure a diverse mix of genders and a broad age range within the 16 to 24-year bracket to promote inclusivity. Additionally, we targeted individuals connected to Trento, including students residing in the city and educators who view it as a pivotal location for their professional activities.

To find non-expert students, we used personal contacts from students involved in our research. For teachers and trainers, we partnered with Glow Association and contacted students by email. To recruit skilled coding students, we used the "Spotted UniTrento " Instagram page, a popular platform for students to share advice and information. We posted an invitation there for tech-savvy and computer science students to participate in our interviews.

In this research, we engaged a total of 21 individuals (Table I). Among the 15 interviewees, non-expert coding students comprised 2 males and 4 females (totaling 6), expert coding students were represented by 3 males and 1 female (totaling 4), and the teacher category included 3 male trainers and 2 female trainers (totaling 5), leading to an overall participation of 8 males and 7 females.

In the focus group, which included 6 participants, both the expert and non-expert student subgroups were composed of 2 males and 1 female, resulting in 4 males and 2 females in total. Consequently, the study featured the perspectives of 12 male and 9 female participants, nearly achieving gender parity but ultimately falling short, despite our targeted efforts. This discrepancy, particularly in identifying female computer science students, was anticipated given their relative scarcity in the field, as highlighted by previous research [3].

In the structured interviews, which targeted different participant groups, we initiated each session with preliminary questions about their backgrounds. These questions served as an introduction and were followed by a more focused inquiry into their knowledge, study habits, and application of coding in daily life. For each category of interviewee, we devised a set of 10 to 20 questions. These inquiries ranged from specific questions such as "What did you study in high school?" to more expansive discussions like "What do you think about the inclusion of coding in all educational paths?" Although the questions were arranged in a logical sequence, we maintained the flexibility to adjust to the natural progression of the conversation.

TABLE I
SUMMARY OF RESEARCH PARTICIPANTS

Participant Category	Male	Female
Non-expert Coding Students	2	4
Expert Coding Students	3	1
Teachers/Trainers	3	2
Total (Interviews)	**8**	**7**
Non-expert Coding Students	2	1
Expert Coding Students	2	1
Total (Focus Group)	**4**	**2**
Overall Total	**12**	**9**

The interviews were structured to last anywhere from 20 to 60 minutes. This timeframe was chosen to ensure a thorough exploration of the topics of interest, while also providing the interviewees with the opportunity to introduce additional subjects they deemed important.

The preliminary questions aimed to facilitate a smooth transition into the discussion, setting the stage for the subsequent responses. The second set of questions, customized for each participant group, aimed to uncover the reasons behind the expert and non-expert students' decisions to either pursue or not pursue coding, along with their perceptions of coding. This portion of the interview also intended to extract insights from teachers and trainers about the application of coding in various settings, both professional and otherwise.

The focus group was carefully organized to consist of six participants who were not previously acquainted with each other. We strived for a balanced composition of expert and non-expert students to create an environment where all participants felt encouraged to express their views. The goal was also to facilitate the moderator's ability to steer the discussion efficiently, minimizing potential confusion and thoroughly exploring the most compelling subjects, thereby fostering a dynamic and engaging dialogue [4].

The focus group was structured into five phases, designed to span 90 to 120 minutes, incorporating individual reflection, paired discussions, and plenary sessions to facilitate diverse interactions among participants. Our goal was for each participant to find comfort in at least one communication mode.

- **Phase 1:** We initiated with a brainstorming session in mixed expert-non-expert pairs, aiming to collaboratively define coding. This shared understanding would then be presented and discussed with the entire group.
- **Phase 2:** Following the brainstorming, the focus was on sharing and debating the definitions of coding developed during Phase 1 with the whole group.

- **Phase 3:** We introduced an experiential component, inviting participants to individually solve one or more puzzles from a selection of seven. These puzzles, mainly wooden objects representing logic problems, were intended to simulate the playful aspect of learning coding.
- **Phase 4:** A plenary reflection on the activities experienced in Phase 3 was planned to encourage collective contemplation and insights.
- **Phase 5:** The final phase involved a group discussion on strategies to introduce young people to coding, focusing on the reactions of non-expert students to the ideas proposed by their expert counterparts.

B. Activities and problems

After finalizing the methodological design, we arranged meetings with the interviewees. These interviews took place in a variety of settings, including department study rooms, the participants' homes, or online, with each session lasting an average of approximately 33.8 minutes. The focus group session was held in a department study room, facilitated by a moderator with the support of two assistants. The timing of the session was collectively agreed upon by all six participants via a WhatsApp group. Both the interviews and the focus group were recorded using mobile devices and audio recorders, with participants' consent.

Upon completing the data collection, we began the verbatim transcription of the recorded material and the coding of the emergent content using the qualitative research software `Atlas.ti`, to highlight agreements and disagreements of thoughts and lived experiences among the various participants. To reduce the risk of linguistic or semantic misinterpretations, each research team member was tasked with transcribing the recordings they were responsible for.

Following the transcription process, we collaboratively coded one interview from each category of interviewee and the entire focus group. The remaining transcripts were then evenly distributed among the research team members to optimize time management. This approach ensured consistent coding practices, facilitated the formation of coherent code groups, and guaranteed uniformity in the analysis process. As a result, the most frequent and significant codes in the respondents' answers emerged, streamlining the identification of key themes for analysis in relation to the research questions.

During our research, we encountered several challenges. Initially, we intended to conduct six interviews per category of participant to ensure a diverse and balanced sample. However, recruiting a varied cohort across age, gender, and educational backgrounds within a limited timeframe proved difficult. Consequently, we were compelled to reduce the total number of interviews from eighteen to fifteen, which led to the exclusion of two expert students and one teacher from our sample.

There was a notable difference in interview durations among participant categories: approximately 54 minutes for teacher trainers, 15 minutes for non-expert students, and 25 minutes for expert students. While not ideal, this discrepancy was seen as an informative aspect of our data analysis. Logistically, arranging a suitable date, time, and location for the focus group that suited everyone's schedules was challenging. Additionally, during the focus group, it was tough to prevent some participants from dominating the conversation, despite their good intentions.

III. Results analysis

In the analysis of the interviews and the focus group, we identified four main macro-areas, each represented by a Code Group, as identified through the `Atlas.ti` software, addressing specific research questions.

- "Coding among young people:" This theme explores whether students in Trento are aware of what coding involves, aligning with our objective to understand their level of familiarity and perception of coding.
- "Benefits of coding:" This area explores whether young people acknowledge the importance of coding for their education, emphasizing both hard and transversal skills.
- "Ineffectiveness of school teaching:" This theme addresses the challenges and shortcomings of current educational approaches to teaching coding, questioning their effectiveness in engaging students and imparting necessary skills.
- "Raising awareness of coding through playing:" This theme focuses on innovative methods to introduce young people to coding, specifically through playful and interactive means that aim to increase interest and participation in coding activities.

A. Coding among young people

To determine if students in Trentino within our target age range understand coding and their opinions on it, we first inquired about their field of study and reasons for choosing it. We then explored their computer science and coding knowledge and attitudes. This method established a baseline for interpreting our research findings, proving useful against the backdrop of interviewees' consensus on the subject's significance. We also valued the insights from teachers and trainers, whose career choices provided distinct yet insightful perspectives on coding knowledge and opinions.

Among students in high school programs or university courses unrelated to technology, science, and informatics, there is a quickly emerging conflicting picture of the issue. For instance, H.D.M., a student of modern languages, expresses dissatisfaction as her school knowledge of computer science has never found concrete application in the real world: "*I did the ECDL course in the first years of high school* [...] *I had to do it* [...] *but if you ask me, "Has it been useful to you so far?" Well, I don't remember that much anymore, you know* [...] *because I did it in the second year of high school, it's been so many years. So not using it anymore* [...] *I forgot everything.*" Nevertheless, the same student firmly believes that "*knowing the computer nowadays is essential*" and goes on to clarify that she does not object to teaching computer science itself, but rather to the teaching method that pushes students to study computer science "*by heart* [...] *just to pass the exam.*"

R.L., currently engaged in international studies after attending a human sciences high school, echoes similar sentiments. During his high school years, he also approached computer science mainly through the ECDL course. Although he "*always liked the subject,*" he believes it would be appreciated to "*maybe do something a bit more specific*" from earlier grades of education, as in Italy, there is a "*very, very basic* [...] *computer culture*". Similarly, M.Z., a law student after completing a scientific high school, points out that computer science learned during different years of study, always mandatory but superficial, was "*not exciting, in the sense that it was not all that great of an activity*" and it taught him "*very basic things*", useful on a daily basis but quite limited.

Regarding coding, non-expert students appear to be aware of its existence as a branch of computer science, but they describe it with approximate terms and clearly indicate their distance from the field. "*Everything that makes up an application*" (R.L.), "*Video games programming*" (M.Z.), "*entering a code to create a website*" (A.C.), or even "*a page, a project, all nice schematic things*" (S.F.) are some of the definitions given to coding functions, not incorrect but certainly vague and incomplete. Alongside these definitions, there are often comments that reveal the interviewee's insecurity at that moment, ranging from M.D.V.'s "*maybe it's stupid*" to H.D.M.'s "*it's a definition from an ignorant person*" along with various other shades of uncertainty. However, just as with computer science, there is a shared belief that having coding skills is "*very interesting*" (S.F.) and important "*for developing a series of skills, even mental ones,* [...] *in an* [...] *efficient way*" (M.Z.).

However, this attribution of importance does not directly correspond to a perception of coding as something close to them, on the contrary: a vast gap opens up on this juncture, between students who have never actually dealt with coding on one side and vice versa on the other side. Even among the so-called "non-expert students", there are substantial differences based on their familiarity with the actual fields of application of coding that they have had the opportunity to acquire in their educational path. For instance, R.L., despite being a self-taught learner of HTML and expressing interest in coding, feels "*very distant*" from it and doubts whether he has the necessary skills to enter that world. Similarly, A.C. believes that coding is suitable for "*intelligent and good at math*" students, but she also mentions her school year in the Netherlands and the extremely interactive and project-based approach to computer science she observed there, finding it "*cool but* [...] *impossible*".

Furthermore, M.D.V., despite studying computer science in his scientific high school, views coding as "*a very complicated and difficult thing*" distant from himself, but he has never actually participated in related activities, although he is aware of their existence. On the contrary, S.F., who attends a vocational school for graphic arts, feels the topic is "*quite close* [...] *because, for example, they ask us to present our personal website with all our projects at the exam*". Moreover S.F. had also had the opportunity, already during middle school, to take part in a national project for girls only, focusing on technology topics, "*in Milan and at the problem-solving Olympics in Cesena*", an experience that positively impacted her, also because "*it was great to see that not only males can do these things*".

The gender issue indeed proves to be particularly significant in influencing the perception of coding, and more broadly the field of computer science, as either close or distant from oneself. J.L.B., a graduate in computer science and currently a secondary school teacher, laments that "[...] *unfortunately, at that age, the idea that science, but especially computer science, is stuff for males has already formed*". In his opinion, it would be appropriate to work on eradicating this preconception starting from the language used, seeking greater inclusion of the female component, albeit with the awareness that "*the difficult thing is to do it in a way that is natural, and not that forced thing*". On the other hand, L.C. (an experienced student) confirms that in his pure computer science degree course, there may be about "10%" of girls, already "*much more than the* 1%" he expected; while I.C., in her experience as a trainer, estimates at most "15%" of female computer science students, who "*all — almost all* [...] *come from the scientific high school*".

B. Benefits of coding

Our research aimed to determine the level of awareness among young people in the Trentino region about the benefits of coding, as reported in literature, and whether they recognize these advantages in their own lives. Coding is known to offer multiple benefits in professional settings and in developing transversal skills for daily activities. Key benefits include enhanced mental flexibility and a strong problem-solving mindset. We found that most interviewees, regardless of their expertise in coding, acknowledged its positive impacts both professionally and personally.

A.C., a non-expert student, states, for example, that "The fact that these things are taught to children [...] already gives them extra capabilities that can be used in life, [...] not only in the IT field, but also in how to approach life in general". It is significant that such importance is recognized in the dissemination of coding skills from an early age by someone not involved in the technical-informatics field. There is a strong belief in the cognitive advantages of coding both in the digital world and in everyday life. Unconsciously, the statement of this student echoes the assumption that possessing computational thinking is "an attitude and a set of universally applicable skills that everyone, not just computer scientists, would be eager to learn and use" [5]. Therefore, mastery of this additional skill proves to be fundamental universally, precisely because "*it could help anyone*" (L.C., expert student).

At the same time, it is possible to argue that this essential skill has been applied by humans long before the recent introduction of technological tools accessible to all or the dissemination of programming languages [6]. In fact, its inherent characteristic is the problem-solving elaborative process, presumably as ancient as humanity itself, whose mastery still needs to be trained, and coding serves as an excellent gym

for it. Several of the interviews we conducted have confirmed this perspective, highlighting how coding allows individuals to acquire a specific method, of a structural and systematic nature, to identify one or more approaches to solving any problem they encounter.

Indeed, expert students and training instructors believe that this is one of the most effective benefits of applying computational thinking to everyday life. As stated by E.M., an expert student, "*In my opinion, it makes you learn to think in a certain way, because in the end, it's about solving problems formally, analyzing a problem, and solving it systematically, so it's useful*". A.M., a teacher and trainer, agrees with the young student, convinced that coding "*teaches you to see everything as a problem to solve, to optimize, to face* [...] *[to] try to improve the process, in short. You develop a logical way of thinking* [...] *[useful for] how I approach problems and* [...] *how I try to solve them* [...] *with a certain method*".

Consequently, coding can be seen as a discipline that, in parallel with the development of logical thinking, leads to the ability to solve problems through structured methods. According to Wing, who builds on Papert's ideas, this systematic nature of coding arises from the fact that programming involves step-by-step formulation of a solution to a specific question, which must then be translated into a form that can be processed by a computer, regardless of its human or technological nature [5]. In other words, coding is not characterized by the mere execution of a program by a computer, but it also involves understanding and formulating problems and solutions, encoded and translated into shared languages. Thus, if we move away from the view that sees computational thinking as firmly anchored and limited to the IT field, it becomes evident how it can be "*applied to a little bit of everything*" (F.M.B., expert student).

Closely related to the universal utility of skills developed through coding, such as problem-solving, is the versatility of this field, which provides "*a broader idea of how certain things work*" (F.M.B., expert student). While some state that they do not consider coding as "*an essential basic skill, if that's not what you want to do*" (M.Z., non-expert student), there is an awareness that "*the further you go, the more important it will be even in the working environment*" (F.C., expert student). The interviewed teachers and trainers confirm this view, considering coding useful in "*any job*", as well summarized by J.L.B., a teacher, who exemplifies it in the professional figure of a garbage collector or a courier. When faced with a map of the city indicating the stops to reach to complete their task, this person must organize themselves logically to select the most optimal route in terms of time and safety. For example, they would prefer a route with right turns rather than left turns, as the latter are more dangerous.

"[...] *it sounds like nonsense, right?*" J.L.B. continues, "*But then you look at the data, and couriers have half the accidents compared to regular people,* [...] *there's a substantial difference. So if everyone could think in terms of analysis, evaluation, and finite reasoning, in my opinion, there would be a lot to gain*". It then becomes evident that, contrary to common belief, computational thinking does not only concern areas related to computer science or engineering but potentially any task. It represents a transversal skill that, like others, can be applied to topics of different nature and origin, and it enhances the worker in any work context.

As such, it is also multifaceted and capable of addressing the needs and situation of each individual, so that A.C., a non-expert student, defines it as particularly useful "*because it allows you to be much more independent*", while L.C., an expert student, mentions how it facilitates the approach to "*anything of a technological nature*", including the use of digital identity. Benefiting both professionals and non-professionals, the autonomy provided by coding is proven to derive, once again, from its fundamental principle of devising solutions. Specifically, the constant hands-on experimentation and the continuous trial and error are necessary difficulties in relation to the advantages they bring [7]: "*No matter how much you think and prepare, there will always be a problem* [...] *[so] another transversal aspect is that [coding] teaches you to be very patient*" (L.C., expert student).

The ideal process of learning coding indeed takes place through distinct and not at all obvious methods. D.L., a teacher involved in the volunteer movement "Coderdojo", notes that while working in various educational settings, he has realized the importance of students mastering the concept of "learning to learn" before delving into programming practices per se. This formula for knowledge transmission is actually recognized as a key skill in learning in general, and as such, it is incorporated into the current education system as stipulated by the European Parliament and the European Council through Recommendation of 2006/962/EC [8]. In teaching coding, this perspective can find particularly fertile ground as a more suitable approach for disseminating computational thinking within school curricula. Additionally, the ability to learn autonomously and consistently can be seen as a result of continuous practice of computational thinking, which, as seen before, comprises various mental skills related to organizing thoughts, processing information, and consequently, one's own education.

"*Coding, specifically, really teaches you to take each step, look at it, and think... do forward thinking*" (L.C., experienced student); it essentially trains you to visualize the different steps that lead to a result, in a systematic and organized way, and to orient your thinking forward. This ability is intuitively connected to that of planning the next steps in one's life, with a view to achieving desired professional and personal outcomes in the near or distant future. In addition to individual impact, it can also have a social effect. In the words of a training manager for Computer Scientists without Border, "*If our boys and girls know how to use it [technology] from an early age, but use it actively, we have given them an extra chance to be active and incisive citizens* [...] *in their daily lives and in society*" (I.C., teacher and trainer).

From these insights, it is clear that young people in the Trentino area recognize the advantages of coding, regardless of their prior experience with it. Despite the limited number

of interviewees, there is a consensus that coding is a valuable skill that enriches their cultural knowledge in various ways. It is particularly beneficial for young individuals, enhancing their skills and mental abilities. This learning aids in complex decision-making and organizational tasks, which are crucial as they mature into adulthood. Additionally, the informed and intentional use of technology fosters their active participation in society, positioning coding as an essential skill for the future across various fields, not just in computer science.

C. Poor effectiveness of school teaching

What has emerged so far demonstrates that young people are aware of the importance that coding holds in their education, primarily in terms of employment prospects but also from the perspective of transversal skills. In general, computer science is considered an essential component not only by the experienced students, among whom some define it as "*the vanguard of sciences*" (L.C.), but also by the non-experienced students, who acknowledge that "*everyone knows it's important*" (A.C.) and even refer to it as "*fundamental*" (H.D.M). The interviewees widely agree on the usefulness of deepening their knowledge in this field, as "*whatever* [...] *you study, it could come in handy*" (L.C., experienced student), and "*these disciplines should be expanded* [...] *because currently, we see them in every aspect of daily life*" (R.L., non-experienced student). Despite recognizing the benefits of coding, there is a notable perception of inadequacy in computer science education in schools, a critique shared by both inexperienced students and teachers/trainers. This feedback, emerging spontaneously during interviews, was considered significant enough to warrant substantial attention in our data analysis. The importance of this critique lies in the fact that the adequacy of computer science education wasn't an initial focus of our research, but emerged as a salient issue identified by the interviewees, often expressed with criticism, irony, and discouragement. Particularly, many inexperienced students pointed out not only a sense of inadequacy but also a weak link between the computer science curriculum in schools and its practical applicability. The criticisms mainly focus on the teaching methodology, described as "boring" because it heavily revolves around passively copying steps demonstrated by the teacher on the computer, without "*ever having the opportunity to take notes, do our own things, or effectively study them*" (A.C., non-experienced student).

The same A.C., enrolled in a scientific high school in the Province of Trento, also emphasizes: "*I didn't like it [computer science] much because the teacher didn't involve us much* [...] *It was boring because we spent the whole period copying from the board*". And she admits: "*I deleted everything, except for the things I already knew how to do*". This is a thesis also supported by H.D.M., a university student, who directly refers to the teaching method: "*Explaining computer science in such a frontal and non-interactive way* [...] *I mean, okay, but for me, it's more difficult*". The inadequacy of adopting a teaching and learning method often focused solely on passing a specific exam or completing the school year is evident. The solution adopted by students in response to this type of teaching is to "learn things by heart", with an overall discouraged attitude: "*No one understood the reasoning, so we said, well, it's like this now, that's enough... So from my point of view, I never really studied computer science* [...] *I don't even know what the exams were about, if you ask me today*" (H.D.M., non-experienced student). The effectiveness of school teaching in coding and computer science is often criticized for relying on a frontal and passive teaching method. This observation aligns with the findings of Papert [9], who argues for the importance of active learning in computer education. Papert suggests that students should learn to program, rather than being programmed by computers. Essentially, the educational focus should shift from merely filling students with facts to empowering them to actively engage with and program. A very similar concept is presented by I.C., a trainer, who highlights that "*machines must be used and guided by us if we want to make them our assistant*". Referring to young people, she adds that "*if we adults help them by providing methods, giving them guidance, [the computer] becomes an ally and a tool they use, they become more and more protagonists of it*".

To further summarize this view with a provocation, "the importance of play and thinking takes precedence over the learned notions" [9]. Coding, in fact, is based on Papert's constructionist learning theory, as it promotes the construction of mental models, the development of critical abilities, and the realization of concrete and operational thought processes. It is also capable of correcting the flaws of traditional schooling, where the student is often a passive object of educational systems that make them less participative [10]. However, this cannot happen if the teaching of coding is also subjected to the aforementioned flaws, as unfortunately confirmed by the professional experience of the trainers and teachers we interviewed in the context of the research.

These ones contribute to the overall assessment of inadequacy in the teaching with particular bitterness, strongly criticizing various aspects. A heavy burden of bureaucratic work, the lack of adequate computer laboratories, and the presence of teachers who are highly unprepared in the subject are just some of their reasons supporting the criticism of the Italian computer science curricula: "*From the perspective of well-organized structural commitment, there is nothing. You can even go through the entire school path without even catching a glimpse of a computer.* [...] *I think that in the guidelines of the Ministry, it is indicated that coding should be done, but it is not mandatory, and many schools do not do it*" (D.L., volunteer CoderDojo trainer).

Very recently, as per the time of writing this research report, art. 24 bis of D.L. 152/2021 has mandated the introduction of computers in all schools, including kindergartens. From the 2022–2023 academic year, the National Training Plan for teachers across all school levels "prioritizes teaching computer programming (coding) and digital education", in line with Law 107/2015 and the National Recovery and Resilience Plan (PNRR). This plan aims to enhance digital learning and skills, with coding being a key focus in the Italian educational

system. As a result, starting from the 2023–2024 academic year, teachers will undergo professional development to align with these new educational objectives. starting from the academic year 2025-26, "*the development of digital skills will be pursued, also promoting computer programming learning (coding), within the framework of existing teachings*" (art. 24 bis D.L 152/2021). Even though at the current state, we must admit, it is far from achieving the hoped-for objectives.

Stimulating class curiosity and fostering active learning requires proper preparation. Volunteer movements like Coderdojo are increasingly focusing on primary and secondary school teachers, introducing them to new concepts within the Italian educational landscape. The emerging knowledge society underscores the need to develop skills for engaging with new knowledge in educational systems. It's crucial to create connections and synergies between different learnings and to use technology as a tool for amplifying human potential and enhancing traditional learning processes [11]. However, the current inefficacy in Italian education is linked to structural rigidity in curricula, evaluation methods, and more, contrasting with the dynamic nature of modern society. This makes it difficult to capture students' interest and involve them in the learning process [11]. The interviews repeatedly highlight the importance of motivation in learning; to effectively engage young people in coding, they must be adequately motivated at the school level.

Besides being an essential element in raising awareness among young individuals about the subject, motivation appears to be a key ingredient in breaking down any prejudice and convincing boys and girls that "*anyone can do coding*" (I.C., trainer). The first step to spur students' interest would be to accommodate their inclinations and personal interests: "*there's the issue of motivation, if you tell them to only copy what you, as a teacher, like, that's what they'll learn; the best thing is to say "create your own website, think of a topic you like, and develop it using the technology I explain to you". That usually makes them take a step forward*" (D.L., trainer).

According to interviewed teachers and trainers, the challenge in motivating students to study is why most coding initiatives are aimed at younger students, primarily in primary or lower secondary schools. For instance, Coderdojo organizes workshops specifically for 7 to 12-year-old. Trainers justify this choice by noting that children in this age group are generally very curious and easily engaged, making it simpler to stimulate their participation and spark their interest in activities like coding. They also learn more quickly than teenagers and are more malleable. As explained by J.L.B., a computer science teacher in primary and secondary schools in Trento, "*Children are extremely motivated, even with something trivial* [...] *if you set it right the boys get excited, and of course, the girls as well*". On the other hand, high school students seem to be a less sought-after target because motivating them appears more complex and challenging: "*[In the] first and second years of high school, you have some problems* [...] *with motivation. Apart from those two or three students who already know they will become computer scientists, engaging the others is the most difficult thing*" (J.L.B., trainer).

I.C.'s experience as the training manager of Informatici Senza Frontiere (ISF) confirms this gap between different age groups and the consequent focus on children when it comes to teaching coding. However, she also adds that, in the age group above 14 years old, "*it has been proven, I would say there is a global consensus now, that if we intervene and teach in a playful, fun way, we can engage girls and boys in using the computer*".

A paradox noted in our research is that while coding can benefit everyone if taught properly, activities predominantly target children and sometimes teachers, often bypassing high school students. Assuming that motivation is central to this issue, our research aimed to explore how to effectively engage older students in coding. We particularly focused on the use of playful activities as a starting point for this engagement.

D. Coding awareness through play

Through interviews with non-expert students, it became evident that coding is often seen as distant, complex, and unrelated to everyday life, especially for those with only a vague understanding of it. As summarized by non-expert student A.C.: "*I think it doesn't concern me much because I don't believe I am capable, I mean I don't believe I would be able to learn it...to know how to use it in the best way*". Despite not excluding interest in coding, these "non-experts" generally lack motivation or confidence in learning the skill. In contrast, expert students, teachers, and trainers emphasize its widespread benefits. A unanimous concern is the inadequate coding education at the school level, with insufficient and inappropriately targeted initiatives for the age group in question.

Based on these findings, our research delved into how to approach and motivate young people, particularly those who view coding as distant, to learn coding. The idea that coding can be learned and is useful to anyone is best conveyed through active motivation, such as playful tools, as supported by experiences and opinions from expert students and trainers. For example, computer science student L.C. was introduced to coding as a child through a "little game" using blocks, where "*depending on how you put the blocks together, you could make things happen, make them repeat certain actions, particular actions*". F.C., a student of communication and meta-electronics engineering, suggests starting with foundational concepts through a game, then progressing to "*start with concepts and foundations, perhaps starting with a game to then make* [...] *them understand what is underneath it all*".

The importance of play in learning has been extensively analysed by various scholars and educators between the 19th and 20th centuries, including Maria Montessori, Rosa Agazzi, John Dewey, and Friedrich Froebel. Their studies highlight the child's ability, through playful activity, to develop creativity and strengthen acquisitions, while also promoting the learning of "emotional, relational, and cognitive skills" [12], [13]. According to the pedagogue Jean Piaget, playful activity also

guides a complete development of the individual as it facilitates socialization and the development of intelligence [14].

Among the interviewees, there is a certain knowledge of different games that can serve precisely this purpose in the field of coding. D.L., a trainer, reports that "*There are games for the general public that have programming elements in them*". Three trainers and one experienced student specifically mention the free programming environment of Scratch, characterized by a graphical programming language, as a game to successfully engage and sensitize young people to the world of coding. Scratch is, in fact, "both a programming language and an online community where kids can program and share with others [...] their interactive multimedia creations like stories, games, and animations" [15]. The teacher and trainer A.M. endorses its highly educational qualities, allowing for quickly achieving visible and concrete results with strong motivational capacity: "*I know that from a certain point of view, it's for kids... but it is also used by adults, and I really like it* [...] *And you learn a lot, you learn geometry, logic, the Cartesian plane, etc. You do a lot of things, and you get results*". The possibility of obtaining real results through Scratch is also observed by the experienced student D.F. during the focus group, who describes it as "*a language, more like a puzzle, where you combine basic components* [...] *and* [...] *you manage to get something real, like making the little character move or things like that*".

As stated by A.M., it is, moreover, a flexible tool that can be adapted to the learning needs of children and adults alike, making it accessible to all ages, proving that "*anyone can do coding*". Equally committed to the cross-cutting expansion of computer science learning through gaming methods, albeit specifically in the school environment, are organizations like Code.org, which aims to provide every student in every school with access to computer science education. The organization's website offers various digital gaming activities focused on programming, designed for young people of all ages. Additionally, there is a section explicitly dedicated to high school students "and beyond" [16], a target that, as seen previously, is rarely taken into consideration in coding education. In addition to increasingly advanced online courses provided by well-known institutions like Harvard University and LinkedIn, here, older students can explore careers related to computer science and opportunities for mentoring, internships, and scholarships [16].

Similarly, the aforementioned Coderdojo movement is mentioned by three trainers and two expert students for its specific goal of introducing children and young people to the world of coding through playful activities. This "*international network of computer clubs where children and teenagers can meet and freely learn to write computer programs, develop websites, applications, games, and much more*" (Coderdojo Verona, n.d.) provides further evidence of the importance and effectiveness of learning coding in an active, enjoyable, and proactive way, quite different from what is typically offered in Italian schools. In the words of trainer D.L., in fact, "*Coderdojo does not have a school-based approach* [...] *Participants come, and they are children from 7 years old and up* [...]*[and] they try to build a little program... usually, the ideal is to create something fun, we teach them to create a video game*".

Another playful dimension that, according to four of the expert interviewees, would raise awareness of coding is the game of Lego bricks. I.C., a trainer, connects them to programming, saying that "*for example, Lego Mindstorms can be programmed with Scratch and similar tools. And then, it always refers to the concept of blocks* [...] *and being able to construct a sequence of bricks to achieve something that is at an exponentially higher level of difficulty compared to the individual brick you have*". The teacher and trainer A.M. takes the correlation even further, stating that "*those who created Lego Mindstorms* [...] *are people who have been dealing with computational thinking for a lifetime. In a way, the idea is that it gives you a mental openness that is associated with a creative game, so to speak*". As previously mentioned, computational thinking and coding are inherently linked, as the latter can develop the former and, in relation to it, the problem-solving activity that falls among its primary benefits.

Role-playing games are also mentioned as playful activities that can be related to these three areas. This is primarily because they are characterized by "*many rules [that] represent a formal system of execution* [...] *that somehow resembles* [...] *that of informatics*", as explained by A.M. "*And then [for] the fact that within this system, however, you have ways to solve problems and, in some way, this is associated with the ability to act in a complex manner, and therefore to find a different solution to a problem of some kind*". The ability to playfully develop coding skills, computational thinking, and problem-solving is similarly recognized in board games, such as Risiko (F.B., expert student) and Scrabble (F.C., expert student). Their affinity with the world of coding, more or less apparent depending on the type of game, once again lies in the presence of precise rules, the need to adopt a personal strategy and planning, and the strong connection to problem-solving [17].

Playful activities, therefore, seem to represent an optimal dimension for raising awareness among young people of all ages about learning coding elements. Through fun and stimulating activities offered by various physical and digital games, designed more or less specifically to lay the foundations of this skill, even those who do not consider themselves endowed with the necessary characteristics to approach the subject can be more easily motivated and passionate about it. This will reduce the perceived distance from this reality, hopefully leading to a greater general awareness of the transversal scope of skills associated with coding.

IV. The local context

By entering the terms "coding" and "Trento" in any search engine, it will take less than a second for various results related to teaching and initiation to computational thinking in the Trentino capital to appear. At the top of the page, for example, you will find the local section of CoderDojo, (https://www.coderdojotrento.it/). Following this, you will see a small series of laboratory events organized in recent years by individual

schools, the University of Trento, and local associations and museums such as Computer Scientists without borders, an Italian association that works to bridge the digital divide, and Muse, the Museum of Science in Trento. These initiatives are often framed within the broader "Hour of Code" movement, also known as "Code Week" (https://hourofcode.com/it;http://www.codeweek.it/). Finally, there are some online training opportunities on "Coding and Computational Thinking" accredited by the italian Ministry of Education, University and Research (MIUR), intended for teachers of all levels of public education.

By further researching and expanding the search, we can add Voxel and DeltaInformatica to the list: the former is a transqueer-inclusive community in Trento for all women who want to become coders and software developers or improve in these fields (https://www.voxel.community/it); the latter is a company that provides various types of training in the field of business informatics. In June 2017, DeltaInformatica received the Family Audit certification issued by the provincial agency for family, birth rate, and youth policies of the Autonomous Province of Trento (https://www.deltainformatica.eu/formazione/). After that, the relevant results will conclude, unless there are any updates from individual schools, the university, or local associations and museums, sometimes in collaboration with each other through funding opportunities, competitions, or similar initiatives.

The Trentino region, while not devoid of coding opportunities, lacks a comprehensive and structured approach to make these opportunities widely and universally accessible. For school-aged children and teenagers, access to coding significantly depends on their environment, such as a family member's or teacher's interest in the subject. This discrepancy leads to a notable variation in coding awareness and its perceived benefits among young individuals from the same city, as our research findings highlight.

At the same time, the analysis of data collected in 2022 presents a concerning picture. Individuals of all ages who have engaged with coding acknowledge its benefits across various sectors, a stance supported by increasing academic research. Yet, the school system, particularly at advanced levels, struggles to modernize its curriculum on coding. It is often criticized for being ineffective, insufficient, and inadequate in teaching programming fundamentals. Most notably, higher education institutions overlook the significant potential of incorporating playfulness into learning. This playful approach could spark curiosity and interest in coding among students who otherwise feel alienated from it.

As seen before, the conducted interviews also confirm the presence of a generalized gap in providing training and introduction to coding for young adolescents, with a preference for children in the age group of 6-12 years, considered more suitable for achieving effective long-term results. Trainers and teachers interviewed report that, unlike adolescents, children are more easily stimulated to learn, engaged in the activities proposed by teachers, and inclined to understand computational thinking. Similarly, they observe a clear gender gap during adolescence that is absent in earlier stages: in their experience, it is during this phase of growth that boys and girls begin to define their interests, and too often, they do so by following stereotypes that associate technical-scientific subjects as predominantly male and humanities as predominantly female. Female students over the age of thirteen are thus considered "lost" by trainers in the field of computer science, and they, in turn, start to perceive computer science as distant from them, regardless of whether they have experienced it or not.

Also, from the identification of these age and gender-related issues, therefore, the following research stems. The primary objective is to spread awareness that "*it is never too late*" to learn the basics of computational thinking (I.C., trainer), and that it "*helps you a lot*" (J.L.B., teacher), and should be known by everyone regardless. It is indeed important to advance an initiative aimed at students, regardless of gender and the educational path they have chosen, avoiding prejudices that women may not be suited for computer science or that programming would be inaccessible to those studying in certain fields. Additionally, this study and the proposed and implemented initiatives aim to raise awareness that coding is not necessarily an activity exclusively related to pure computer programming. It will promote transversal skills and competencies that can be applied in all fields.

Our research emphasizes the critical importance of teaching coding to young people aged ten and above, particularly in secondary schools. At this pivotal age, students begin to identify their interests and make decisions that will shape their educational and career paths, regardless of their initial inclination towards or away from computer science and coding. Unfortunately, their choices are frequently based on incomplete or misleading information, shaped by stereotypes and trends. During the crucial last three years of school, when significant life decisions are made, students often rely solely on the advice of adults in their lives.

The backbone of the entire proposal and litmus test of its success will necessarily be inclusivity: to ensure that the initiative remains consistent with the context analysis conducted and the data collected, we find essential that it does not perpetuate any form of discrimination and is aimed at all categories of boys and girls, without distinctions. While creating an activity that equally engages students of different coding experiences presents challenges, concentrating only on those with limited prior or future opportunities to explore coding seems misguided. We argue that this approach could inadvertently emphasize existing disparities rather than diminish them. Moreover, it could hinder efforts to showcase coding's broad applicability across disciplines.

The widespread belief, found among the students we have referred to as "non-experts", that coding is something unattainable, complex, and distant from their lives, is instead the myth that we aim to debunk with what we are proposing. Our hope is to intrigue and engage the participants in the topics we promote, so that if they wish, they can then independently explore them as they see fit. Therefore, the activities we will

present from chapter 5 are not motivated as a direct response to an explicit need, but rather as a desire to fruitfully stimulate young people to learn a skill that, for the most part, they have not yet had the opportunity to appreciate its multifaceted relevance. We believe that practices of coding awareness and digital prototyping through play deserve more attention at the institutional as well as extracurricular level, especially in high schools. We will aim to seize the opportunities offered by innovative and playful learning approaches, encouraging creative, logical, and computational thinking.

V. Conclusions

The transition from the era of knowledge to the era of skills is happening ever more rapidly [18]. Proficiency in transversal, basic, or specialized technical skills is fundamental for the positioning of workers at different levels of organizations within the complex and articulated reality known as the "learning society".

In the context of a liquid society, where "flexibility is the slogan" [19], knowledge is continuously transforming, and the ability to perform professionally has been replaced by the ability to act professionally, where competence assumes a generative capacity for ever-new performances [20]. This is evidenced by the relevance attributed, in professional action, to tacit knowledge and competencies, such that today work is no longer reducible to the exercise of skills considered necessary for its performance but examines the protagonism of the individual with their cognitive-behavioral resources [18].

The preliminary research presented has considered coding precisely as a skill deemed fundamental for the education of new generations, to the extent that it has been defined as a "national priority" by the recent PNRR (art. 24 bis D.L. 152/2021).

Considering this theoretical framework, the sociological analysis carried out demonstrates that young people from Trentino are indeed aware of the recognition coding receives in contemporary society, both in terms of hard skills and transversal competencies. However, paradoxically, the benefits associated with coding collide with a prejudice concerning their perception of themselves as ignorant and the impression of coding being excessively difficult. Moreover, especially from the words of teachers and trainers, the issue of the gender gap emerges, a relevant topic in sociological tradition, made evident in this case by the under-representation of women in the study of informatics and STEM disciplines in general. The gender issue poses itself as one of the most interesting themes, also in the perspective of possible future research lines, particularly with the purpose of focusing attention on the opinions, feelings, prejudices, and plans of boys and girls regarding coding and the gender gap in this field. The collected data then pointed to a widespread perception of inefficacy of the Italian education system regarding the teaching of computer science, a sentiment confirmed primarily but not exclusively by non-expert high school students as well as university students. This issue, although initially not identified as the focus of the research and not considered in the construction of the interview script, turned out to be central to defining the activities of the focus group. To overcome a teaching methodology that often discourages those who should instead be encouraged and sensitized to the study of computational thinking, the data suggested addressing students' motivation, primarily through the use of playful activities. To experiment with what was learned in this regard, a moment of play dedicated to solving puzzles and logical enigmas was included during the focus group. The enthusiastic and interested response of all participants to the planned activity and the subsequent discussion confirmed the results of the analysis regarding sensitization to coding through playful activities and strengthened our belief in the validity and reasonableness of the activities and projects developed among our research group.

References

[1] F. Fiore, "A constructionist approach for the future of learning," Ph.D. dissertation, TRENTO, 2023.

[2] Anief. (2021, Nov.) Decreto pnrr, tra gli emendamenti al vaglio della camera l'avvio di istituti innovativi e del coding. [Retrieved: 05/01/2022]. [Online]. Available: https://anief.org/stampa/news/37282-decreto-pnrr,-tra-gli-emendamenti-al-vaglio-della-camera-l%E2%80%99avvio-di-istituti-innovativi-e-del-coding

[3] CRUI. (2021) Vademecum per l'elaborazione del gender equality plan negli atenei italiani. [Retrieved: 10/01/2022]. [Online]. Available: https://www.crui.it/archivio-notizie/vademecum-per-l%E2%80%99elaborazione-del-gender-equality-plan-negli-atenei-italiani.html

[4] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education*, 8th ed. Routledge, 2018.

[5] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006, [Retrieved: 10/01/2022].

[6] A. Yadav, H. Hong, and C. Stephenson, "Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in k-12 classrooms," *TechTrends*, vol. 60, pp. 565–568, 2016.

[7] G. S. Stager. (2009) Eight big ideas behind the constructionist learning lab. [Online]. Available: https://stager.org/articles/8bigideas.pdf

[8] European Parliament and Council of the European Union, "Recommendation 2006/962/ec on key competences for lifelong learning," Official Journal of the European Union, 2006.

[9] S. Papert, *The children's machine: rethinking school in the age of the computer*. New York: BasicBooks, 1993.

[10] M. Resnick, *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT Press, 2017.

[11] G. Olimpo, "Società della conoscenza, educazione, tecnologia," *TD-Tecnologie Didattiche*, vol. 50, pp. 4–16, 2010.

[12] S. L. Martinez and G. S. Stager, *Invent to Learn: Making, Tinkering, and Engineering in the Classroom*, 2013.

[13] R. Sgambelluri, "Il gioco come strumento di cura educativa: cenni storici e codici pedagogici a confronto," *Formazione & Insegnamento*, vol. 13, no. 2, pp. 73–80, 2016.

[14] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., 1980.

[15] Scratch. For parents. [Online]. Available: https://scratch.mit.edu/parents/

[16] Code.org. Chi siamo. [Retrieved 28/12/2021]. [Online]. Available: https://code.org/international/about

[17] M. Education. (2020, Oct.) Dai giochi da tavolo al coding — viviana laura pinto. Retrieved: 28/12/2021. [Online]. Available: https://www.youtube.com/watch?v=ZK6uMcgAi0c&t=1460s

[18] J. Delors, I. Al Mufti, I. Amagi, R. Carneiro, F. Chiung, B. Geremek, W. Gorham, A. Kornhauser, M. Manley, M. Padrón Quero, M.-A. Savané, K. Singh, R. Stavenhagen, M. Won Suhr, and Z. Nanzhao, *Learning: The Treasure Within: Report to UNESCO of the International Commission on Education for the Twenty-First Century*. Paris: UNESCO Publishing, 1996.

[19] Z. Bauman, *Liquid Life*. Polity Press, 2005.

[20] B. Rey, *Ripensare le competenze trasversali*. Milano: Franco Angeli, 2003.