JOURNAL OF ENGINEERING RESEARCH & SCIENCES

JENRS

www.jenrs.com
ISSN: 2831-4085

Volume 2 Issue 3
March 2023

Editorial

In this edition of our journal, we feature two remarkable research papers that address cutting-edge advancements in augmented reality (AR) technology and computational thinking in early education. These studies offer significant contributions to their respective fields, providing innovative insights and practical applications that can transform how we interact with technology and how we educate future generations.

The first paper explores the use of augmented reality (AR) headsets across various industries, including oil, healthcare, and the military. Despite the widespread adoption of AR technology, there has been a notable gap in research and design recommendations for presenting information effectively in AR headset displays to aid situational awareness. This study conducts two essential investigations: one on the perceptibility of visual stimuli (color, text, shapes) for critical information, and another on the effectiveness of different presentation styles (Display, Environment, Mixed Environment) for secondary textual information. The findings reveal that existing visual perception principles can be applied to AR headsets, establishing a hierarchy of salient visual features. For secondary information, the Display and Environment presentation styles significantly enhanced participants' perception and comprehension compared to the Mixed Environment style. These results provide valuable design recommendations for AR headset displays, crucial for safety-critical domains such as the military [1].

The second paper addresses the growing importance of computational thinking, often referred to as the "new English," in school curricula worldwide. Recognizing the need to introduce coding concepts at an early age, the researchers implemented a training program starting from kindergarten. The program's primary objective is to teach coding to children as young as three, leveraging their inherent logical thinking abilities. Over three years, the program has been refined and expanded, showing impressive results in children's understanding and engagement. By gradually increasing the complexity of concepts based on previously learned material, the program optimizes learning time while minimizing the required hours and resources. Moreover, the program's cost-effectiveness ensures it is feasible for any school, making it an accessible and practical solution for integrating computational thinking into early education [2].

The two papers presented in this edition exemplify the innovative and impactful research that our journal strives to publish. From enhancing situational awareness with advanced AR technology to revolutionizing early education with computational thinking, these studies provide valuable contributions to their fields. We are honoured to share these insights with our readers and anticipate that they will inspire further advancements and research.

References:

[1] J. Woodward, J. Smith, I. Wang, S. Cuenca, J. Ruiz, “Designing Critical and Secondary Information in Augmented Reality Headsets for Situational Awareness,” Journal of Engineering Research and Sciences, vol. 2, no. 3, pp. 1–15, 2023, doi:10.55708/js0203001.

[2] E. Benetti, G. Mazzini, “Coding: First Steps from Kindergarten up to Primary School,” Journal of Engineering Research and Sciences, vol. 2, no. 3, pp. 16–30, 2023, doi:10.55708/js0203002.

Editor-in-chief

Prof. Paul Andrew

JOURNAL OF ENGINEERING RESEARCH AND SCIENCES

Volume 2 Issue 3	March 2023

CONTENTS

Designing Critical and Secondary Information in Augmented Reality Headsets for Situational Awareness 01
Julia Woodward, Jesse Smith, Isaac Wang, Sofia Cuenca, Jaime Ruiz

Coding: First Steps from Kindergarten up to Primary School 16
Elisa Benetti, Gianluca Mazzini

Special Issue on Computing, Engineering and Sciences

JENRS

Received: 22 November, 2022, Revised: 10 February 2023, Accepted: 24 February, 2023, Online: 23 March, 2023

DOI: https://doi.org/10.55708/js0203001

Designing Critical and Secondary Information in Augmented Reality Headsets for Situational Awareness

Julia Woodward[1*], Jesse Smith[2], Isaac Wang[3], Sofia Cuenca[4], Jaime Ruiz[2]

[1]Department of Computer Science and Engineering, University of South Florida, Tampa, Florida, 33620, USA
[2]Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, 32611, USA
[3]Department of Computer Science, James Madison University, Harrisonburg, Virginia, 22807, USA
[4]Department of Computer Systems, Farmingdale State College, Farmingdale, New York, 11735, USA
*Corresponding author: Julia Woodward, 4202 E. Fowler Ave ENG030, Tampa, FL, 33620, USA, juliaevewoodward@usf.edu

ABSTRACT: Augmented Reality (AR) headsets are being used in different contexts (e.g., the oil industry, healthcare, military); however, there is a lack of research and design recommendations on how information should be presented in the AR headset displays, especially for aiding users' situational awareness. We present two studies: one examining if existing findings on the perceptibility of three types of visual stimulus (color, text, shapes) can be applied to AR headsets for *critical information*, and one analyzing three different presentation styles (Display, Environment, Mixed Environment) for textual *secondary information* in AR headsets. Our study on secondary information is an extension of prior work. For critical information, we found that existing visual perception findings can be applied to AR headsets; there is a hierarchy of salient visual features. Understanding that we can utilize prior work on visual features helps in designing salient critical information for AR headset displays. For secondary information, we found that having the text in the Display and Environment presentation styles assisted in participants' perception and comprehension when compared to the Mixed Environment presentation style. Based on our results, we provide design recommendations on how to present critical and secondary information in AR headset displays to aid in users' situational awareness, which is essential in safety crucial domains such as the military.

KEYWORDS: Augmented Reality, Situational Awareness, Design

1. Introduction

Augmented reality (AR) systems combine virtual elements with the real-world environment [1]. Compared to more traditional AR platforms (e.g., tablets, smartphones, computers), AR headsets are becoming more popular due to providing more mobility, hands-free capabilities, and user immersion [2,3]. AR headsets are entering the consumer market [4], and are also being employed in industrial settings [5]. However, prior research studies have focused on examining the applicability of using AR headsets in various environments and not on investigating how to design the information in the display, especially for aiding users' situational awareness. Situational awareness consists of three levels: perception and detection of elements in the environment (level 1), comprehension and interpretation (level 2), and prediction of the future status of the environment (level 3) [6,7]. Situational awareness is separate from users' decision making [7] and is essential in safety crucial domains [8]. A lack of situational awareness has been attributed to tragedies, such as aircraft crashes [8], oil spills [9], and errors in anesthesia [10].

Since AR keeps users aware of their surroundings while providing additional virtual information in real-time, AR has the capability to increase users' situational awareness. Prior work has examined applying AR headsets for users' situational awareness in a wide range of contexts, such as the military [11], maintenance [12], construction [13], and healthcare [14]. However, prior work has mainly studied the applicability of AR instead of investigating how to design the visual information for aiding users' situational awareness (e.g., [14–17]). In addition, there has been conflicting results with using AR, such as both higher and lower situational awareness [15,16,18] and cognitive workload [19,20]. Therefore, it is important to study how the design of information affects users' situational awareness, since not considering the design and simply applying AR to situations may not be beneficial (e.g., lower situational awareness).

In this paper, we examine how the presentation of visual information in AR headsets can affect users' situational awareness (i.e., level 1 perception and level 2 comprehension). In terms of information necessity, visual information can be separated into two categories: central or critical information (e.g., hazard warnings, essential information) and peripheral or secondary information (e.g., current time, nonessential information) [21]. Critical information is essential to comprehend when completing a task, while secondary information may be beneficial but not necessary. Therefore, we conducted two separate studies focusing on critical and secondary visual information. In our critical information study, we focused on *perception* through examining if existing findings on the perceptibility of three types of visual stimulus (color, text, shapes) can be applied to AR headsets. Previous work in visual perception has found a hierarchy of salient features. For instance, people can more easily detect color than shapes and text [22,23]. However, it is unclear if these findings translate to AR headsets due to these headsets possessing technological and perceptual challenges. The low resolution and loss of visual acuity in AR headsets can negatively affect legibility, object recognition, and depth perception [24,25]. Also, the environment and transparency of virtual elements can impact users' color perception in AR headsets [24,26]. Understanding if we can apply existing perception findings to AR headsets will aid in the design of salient critical information. In our experiment, participants had to monitor visual stimulus in an AR headset while completing math problems on a tablet, and press a physical button when they noticed a specific visual condition in the headset (Figure 1). The math problems were used as a cognitively demanding task for the participants to focus on. The visual stimulus was locked to the AR headset display view. We designed the stimulus to always be present because critical information should be prominent and immediately perceptible [27,28].

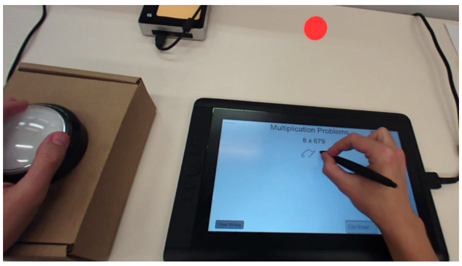


Figure 1: Study setup for critical information study: view from right-handed participant during color visual condition (color red is in AR headset display).

While visual saliency is essential for critical information, secondary information does not have this restriction, and therefore can display larger quantities of detailed information as text (date, time, descriptions, etc.). Since secondary information does not need to be as visually salient, there are more opportunities to integrate it with the environment. Therefore, for our second study, we investigated three different presentation styles for textual secondary information in the context of aiding both *perception* and *comprehension*: locked to the display view (Display), located in the environment (Environment), and a mix of both (Mixed Environment). As in the critical information study, participants had to monitor the textual information in an AR headset while solving math problems. Our study on secondary information is an extension of prior work [29]. This paper updates and expands upon the original work by adding additional literature, an exploratory study on the location of elements, the experiment on critical information, and more results from the study on secondary information.

In our critical information study, we found similar results to prior work with participants having a faster response time for color, and a slower response time and higher cognitive workload for text. In our secondary information study, we found that the Display and Environment presentation styles improved perception and comprehension of textual secondary information; participants had a higher recall of information. Our results provide a new understanding of how different types of visual stimulus for critical information and different presentation styles for textual secondary information in AR headsets can aid users' situational awareness. We contribute design recommendations on how to present visual information in AR headsets for users' situational awareness. Recognizing how to design visual information in AR headsets to improve situational awareness has a wide range of implications in safety crucial environments, such as surgery.

2. Related Work

We focus our review of prior work on two categories: (1) using AR for situational awareness and (2) examining the presentation of information in AR headsets.

2.1. AR for Situational Awareness

Prior work has started studying using AR for users' situational awareness in safety domains, such as the military [14,16,17,30,31]. The authors in [31] created an AR system (FlyAR) to support Unmanned Aerial Vehicle (UAV) flight navigation. FlyAR supports live flight supervision by overlaying the flight path onto a live video stream on a tablet PC, as well as using graphical elements to show height and distance between points. The authors in [16] also developed an AR system for UAV operators, in which flight data was overlaid onto a video stream of the flight on a computer screen. Before the AR system, operators would have to look at two separate screens. The authors found that the AR system improved the operators' situational awareness. In the security domain, [32] proposed a conceptual AR computer-based design for combining social media data (e.g., twitter posts) with contextual information (e.g., Google Maps) to increase emergency operators' situational awareness. The authors iterated on the design based on a workshop with AR and situational awareness experts but did not implement and evaluate the designs in a real-world context.

AR has also been analyzed in the context of driving [17,33,34]. In [17], the authors proposed an AR car windshield system to increase drivers' situational awareness by providing warning information to the driver. For example, it would detect another vehicle and add color depending on how close the vehicle was. The authors in [33] designed an AR driving system that provides warnings on the car windshield for pedestrian collision (e.g., a yellow outline). The authors conducted a driving simulator study and found that the AR visual cues enhanced drivers' awareness of pedestrians. In [34], the authors examined object segmentation visualizations for automated vehicles. They found that including the segmentation visualizations over the car windshield, instead of a tablet on the console, resulted in participants having lower cognitive workload and higher situational awareness, especially for color segmentations over both dynamic and static objects.

While the studies listed above investigated using AR for situational awareness, they only analyzed traditional displays (e.g., car windshields, computer screens), not AR headsets. AR headsets provide more user immersion and freedom, as well as contextual integration with the environment. Previous studies have started to examine using AR headsets for aiding users' situational awareness, such as to monitor patient information [14,15]. In [14], the authors investigated if AR headsets can aid anesthesiologists in monitoring patient information during surgery. They conducted a simulated operating environment study and found that the anesthesiologists that used the AR headset spent less time looking at the anesthesia machine and detected patient events faster. The authors in [15] analyzed if AR headsets could help nurses' patient alarm management decisions and situational awareness by showing patient vital signs. The authors found that using the AR headsets resulted in nurses having higher situational awareness, less errors in recognizing alarms, and faster alarm reaction times.

Prior work has also examined employing AR headsets in the military and security domains [11,30,35,36]. In [11], the authors developed an AR headset system that displays tactical information (e.g., navigation waypoints) for soldiers on foot. The authors in [35] analyzed using an AR headset to show real-time navigational information for the US Coast Guard. They ran a training simulation and found that the AR headset increased operator track keeping and situational awareness; however, it lowered operator responsiveness. For remote pilots of UAVs, [36] investigated using AR headsets to show telemetry details. The authors found that the AR headsets allowed the pilots to focus more on keeping the aircrafts in their field of view, instead of looking at the ground control station. In [30], the authors examined using AR headsets to provide distributed team awareness, specifically in the security domain (e.g., collecting evidence). One team member would be physically present in the environment with an AR headset while a remote team member would be watching a video stream from the headset camera on a computer screen. The remote member could add and edit virtual content displayed in the collaborator's headset (e.g., arrow pointing to specific evidence). They found that the team member wearing the AR headset had higher cognitive workload and lower alertness, while the remote team member had a higher understanding of the situation.

AR headsets are also being utilized in other domains to aid in users' situational awareness, such as firefighting [37,38] and agriculture [39]. The authors in [37] analyzed a proof-of-concept design for using an AR headset in a simulated fire scenario. In the design, the optimal path to the fire would be displayed, as well as fire extinguisher locations. They found that the proof-of-concept reduced travel distance and improved firefighting efficiency. In [38], the authors designed and built an AR headset prototype, which displayed thermal imaging and object segmentation visualizations to help firefighters see their environment during situations with limited sight (e.g., heavy smoke). For agriculture, [39] created an AR system to help farmers monitor their agricultural machines. The locations and status of the machines would be shown in the AR headset display.

These studies highlight the applicability of using AR headsets in domains that require situational awareness; however, they did not analyze *how* to present the visual information in the headsets. None of the studies compared different designs of information, which can impact users. Prior work has shown that AR headsets can result in slower completion times [40,41], higher discomfort [42], and higher cognitive workload [43] when compared to traditional methods (e.g., paper instructions). Therefore, not considering the design of information and simply applying AR headsets to different contexts may not be beneficial. We go beyond prior work by examining different types of visual stimulus for critical information and different presentation styles for secondary textual information in the context of situational awareness.

2.2. Presentation of Information in AR Headsets

Prior work has started to examine the presentation of information in AR headsets [23,44–49]. During a maintenance assembly task, [48] compared using an AR headset (3D animations vs. video instructions) to traditional paper-based instructions. The authors found an improvement in participants' task performance (e.g., faster completion time, fewer errors) when using the AR headsets compared to paper instructions. For the AR headset, while it varied between 3D animations and video instructions, the participants could always see textual instructions and an image of the current tool they needed; the text design remained the same. The authors found that 3D animations, when compared to videos, in an AR headset lowered task completion times. In [47], the authors investigated different AR headset interface designs during a warehouse job simulation (i.e., finding order parts). The designs included text-based versus graphic-based designs, as well as always-on versus on-demand information. They found that graphic-based and always-on information helped users' task performance by reducing completion times and errors. The authors in [49] examined user preferences on how to convey information in industrial AR interfaces. The study consisted of a questionnaire with mockup images of an AR interface. The 3D CAD models were the most preferred, with text being the least preferred. However, the study only focused on assembly, did not use an actual AR device, and did not examine different designs and presentations of text. In addition, prior work recommends that text should not be completely removed for task instructions [50].

In [51], the authors developed an AR headset prototype to help users understand conversations in a noisy echoic environment. The prototype distinguishes between speakers by putting a symbol above their head (i.e., a blue triangle with a white flag and number). When a speaker talks, the audio is transcribed and displayed in the AR headset. The text is shown in black on a static off-white panel at the bottom of the display along with the number associated with the current speaker. The authors did not look at different text designs, only feasibility of the prototype. In [44], the authors analyzed different text positions in an AR headset for reading. When the text was in the top-right, users had higher cognitive workload and lower comprehension when compared to the center and bottom-center locations. The center and bottom-center locations resulted in users having lower cognitive workload and higher comprehension. They also examined two presentation styles: line-by-line scrolling and word-by-word. The word-by-word style resulted in higher user comprehension when users were sitting and reading, while the line-by-line scrolling style had higher comprehension when users were walking and reading.

Previous studies have investigated text and background panel colors in AR systems. The authors in [52] found that using white text with a blue panel background was the best for user readability in AR headsets. In [53], the authors conducted a crowd-sourcing study on user preferences for colors for text and background panels in AR smartphone applications. Most of the participants preferred red or blue background panels with white text. However, prior work has also recommended transparent backgrounds [45], which was not an option for participants in [53]. The authors in [45] conducted a user study, in which participants organized items in a grocery store while viewing product information in an AR headset in two modes: see through mode (i.e., transparent) or panel overlay (i.e., opaque background). The participants preferred the product text to be displayed in the center of the headset in see through mode (i.e., no background) for readability, as well as being able to easily switch between the information and environment. In [54], the authors tested different AR headset text magnification designs for low-vision users. The authors found that the participants liked the transparent background panels and that anchoring content in 3D space can support a more natural and flexible reading experience.

These prior studies examined the presentation of information in AR headsets, but mainly focused on readability and user preferences instead of situational awareness. It is important to examine how information should be presented in AR headsets to aid in users' awareness since these devices are being used in a wide range of contexts that require situational awareness (e.g., healthcare). For our studies, we focused on analyzing how different presentation styles for textual secondary information and different types of visual stimulus for critical information in AR headsets can aid in users' perception and comprehension (i.e., the first two levels of situational awareness).

3. Exploratory Study: Location

We first conducted an exploratory study to determine the best location to place the visual stimulus in our experiments. We wanted to choose the location, in which the participants would have the fastest reaction time possible. For this study, participants had to press a physical button when they noticed a dot appear in a Meta 2 AR headset [55]. The study was conducted in a windowless room with consistent lighting and took 5 to 10 minutes. In the application, a white 3D cube (25.4 millimeter (mm) edge length) remained in the middle of the field-of-view, while white dots (6.35 mm diameter) would appear in different locations along the periphery. Even though prior work has shown that reaction time increases for stimulus in the periphery [56,57], we focused on peripheral locations because we wanted to examine the perception of information that would not block the users' view or distract them from their main task. The participants were instructed to focus on the 3D cube and to hit the button when they noticed a white dot appear. The dot would appear in one of sixteen locations along the periphery; there was 75 mm between each dot to create all sixteen locations. When the participant pressed the button or if the participant did not notice the dot after 2 seconds, the dot disappeared. The next dot would then appear in a different location after a random time interval (1-4 seconds). Each participant viewed 80 dots total (5 dots per location). The location order was originally randomized, and then the same order was used for every participant. Participants volunteered without compensation.

3.1. Equipment

The AR application was created using Unity [58], and was run on a Meta 2 AR headset [55]. The headset features a 90-degree field-of-view with a 2560 x 1440 resolution. The physical button had a 76.2 mm diameter. We used a rectangular cardboard box (228.6 mm x 152.4 mm x 76.2 mm) as a base for the button.

3.2. Participants

We had a total of 12 adult participants (M = 23.42 years, SD = 3.55); however, we excluded one female participant for not wearing their corrective lenses. Therefore, we had a total of 11 participants for analysis (M = 23.09, SD = 3.53). Out of the 11 participants, six participants were female, one participant was left-handed, and five participants had used an AR headset at least one time before. All of the 11 participants had normal or corrected-to-normal vision (e.g., eyeglasses).

3.3. Data Analysis and Results

For analysis, we examined the participants' response time by location. Response time was calculated as the time it took a participant to press the button after a dot appeared in the headset. We excluded the times in which the participant did not notice the dot and it disappeared. We grouped the 16 individual locations into three categories: left, center, and right. The center included the six dot locations at the top and the bottom not located on the left and right edges. A Shapiro-Wilks test found that the data was normal; however, a Mauchly Test for Sphericity showed that the data did not have equal variances ($p < 0.01$). A one-way repeated measures ANOVA (RM-ANOVA) with a Greenhouse-Geisser correction found a significant main effect of location on response time ($F_{1.17,10.54} = 9.48$, $p < 0.01$). A Bonferroni post-hoc comparison showed that the participants had a significantly faster response time for the center locations (M = 0.399s, SD = 0.086) compared to the right-side locations (M = 0.444s, SD = 0.172); this is similar to prior work, which has found lower detection accuracy for the right-side of the visual field [59]. There was no significant difference in response time between the center and left-side locations (M = 0.418s, SD = 0.138).

To further analyze the locations, we examined: (1) the corner locations compared to the remaining locations, and (2) the top locations versus the bottom locations. For both the corner locations and the top versus bottom locations, a Shapiro-Wilks test found that the data was normal, and a Levene's test showed that the data met the assumption of equal variances. A paired-sample t-test found a significant main effect of corner locations on response time ($t(9) = 2.92$, $p < 0.05$). The participants had a significantly slower response time (M = 0.452s, SD = 0.202) for the corner locations compared to the remaining locations (M = 0.408s, SD = 0.102). A paired-sample t-test found no significant difference between the top and bottom locations ($t(9) = 1.27$, n.s.).

Based on our results, we decided to place the different types of visual stimulus in the top-center of the field-of-view for our first study on critical information, to increase perceptibility. For our second experiment on presentation styles for secondary information, we placed the textual information on the left-side of the field-of-view (avoiding the corner locations). We decided to place the secondary information on the left-side because we did not want the quantity of the information to block the participant's view, and there was no significant difference in response time between the center and left-side locations. Also, people exhibit a leftward visual bias, known as *pseudoneglect* [60,61], which results in higher detection accuracy and faster motion processing for elements on the left when compared to the right [59,61,62]; even for computer screens [59]. Although pseudoneglect occurs in both right-handed and left-handed people, it is not evident in cultures that read right-to-left [63]. Therefore, it is important to keep in mind users' cultural groups and differences when placing elements in headset displays.

4. Experiment 1: Critical Information

In our critical information study, we focused on examining three different types of visual stimulus in an AR headset: color, text, and shapes. These types of visual stimulus are commonly utilized to denote information, such as in warning signs [64]. The goal was to analyze if prior results on the perceptibility of types of visual stimulus could be applied to AR headsets. Previous work in visual perception has found a hierarchy of salient features. For instance, people more easily detect color than shapes and text [22,23]. However, it was unclear if these findings translate to AR headsets due to these headsets possessing technological and perceptual challenges. The low resolution and loss of visual acuity in AR headsets can negatively affect legibility, object recognition, and depth perception [24,25]. Also, the environment and transparency of virtual elements affects users' color perception in AR headsets [24,26]. Determining if we could apply existing findings to AR headsets allows us to further understand how to design critical information.

4.1. Participants

We had a total of 37 adults participate (M = 22.19 years, SD = 5.59); however, we excluded one female participant due to equipment failure, resulting in a total of 36 participants for analysis (M = 22.22 years, SD = 5.67). Twelve participants were female, two participants were left-handed, and ten participants had used an AR headset before. We did not recruit participants who were color-blind or dyslexic, and all of our participants had normal or corrected-to-normal vision.

4.2. Method and Design

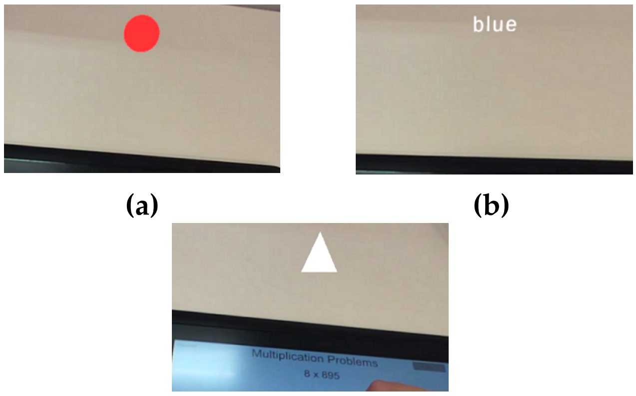
While wearing an AR headset, participants completed multiplication problems on a touchscreen tablet and different types of visual stimulus appeared in the headset: color, text, or shapes (Figure 2). The participants would only see one type of visual stimulus at a time, not a mix of all three. We placed the stimulus in the top-center of the headset field-of-view based on our results from our exploratory study.

For each visual stimulus type (e.g., color), the visual condition displayed would constantly change (e.g., switching between different colors). Participants were instructed to hit a physical button with their non-dominant hand when they saw a specific visual condition (e.g., color red). The participants took part in the study for approximately 60 minutes in a windowless room, and either volunteered without compensation or received extra credit in a course they were taking.

After consenting to participate, participants completed a demographic questionnaire. The participants then completed a 4-minute practice round of multiplication problems on the tablet without wearing the AR headset to get comfortable with the math application; which was not used in analysis. After the math practice, participants put on the AR headset and began the main study. In the main study, there were six study blocks (approximately 4-minutes each), two blocks per visual stimulus type (color, text, shapes). After completing two blocks for a visual stimulus type, the participants would take the NASA Task Load Index (TLX) [65] for that visual type, which is used to determine participants' perceived cognitive workload. The participants would then complete the next two blocks for a different visual type, the NASA TLX, and then move on to the last visual type blocks. The order of the visual types was counterbalanced across participants. After the participants completed the six study blocks, they were also asked questions about their subjective preference.

4.2.1. AR Application Design

Each visual stimulus type had a total of four different options that would constantly change in the headset in a randomized order and for a randomized duration. The four options for each type of visual stimulus included: (a) color: red, green, yellow, and blue; (b) text: "red", "green", "yellow", and "blue"; and (c) shapes: circle, triangle, star, and square. All types of visual stimulus had the same color saturation and brightness and had the same width (12.7 mm). For color, only the color of the circle changed (Figure 2a). Both the text and shapes (Figures 2b and 2c) were white, since black is transparent in AR headsets. The text height was 5 mm, which is consistent with Meta AR design recommendations [66], and in Liberation Sans font since prior work recommends using sans-serif fonts for text readability [67].



(a) (b) (c)

Figure 2: Types of visual stimulus from participant view from AR headset: (a) color, (b) text, and (c) shapes.

During each study block, the participant would view a total of 32 visual conditions (4 options x 8 occurrences). For example, if it was a color block it would constantly change between the four colors, and each specific visual condition (e.g., color blue) would appear eight times. The current visual condition (e.g., color red in Figure 2a) would remain in the headset display for a random time interval (6 to 9 seconds) before switching to the next visual condition. If the participant pressed the button it would automatically switch to the next visual condition, regardless of the amount of time left for the current visual condition. The last visual condition in that block would remain visible in the headset until the participant finished the current math problem and then both applications would end. Each study block was approximately 4 minutes. The blocks were not exactly 4 minutes because each visual condition would change after a random time interval, and the study block would not end until the participant finished the last math problem.

Since there were only two blocks for each type of visual stimulus, we had to determine the two visual conditions for when the participant would hit the button. For color, the participants hit the button when they saw the color red for one block and the color green for the other block because adults have faster reaction times for red and green colors, compared to yellow and blue [68]. For text, the participants hit the button when they saw the word "red" for one block and the word "blue" for the other block. Prior work has shown that word processing time increases as the number of letters increase [69], therefore we chose the two shortest words (out of the four options) to have the fastest reaction time possible. For shapes, the participants hit the button when they saw a circle for one block and a triangle for the other block. We chose a circle and triangle because they are used by the International Organization for Standardization (ISO) to represent mandatory actions and warnings [64].

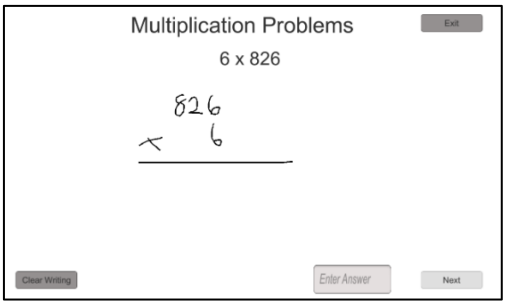


Figure 3: Screenshot from our math application.

4.2.2. *Math Application Design*

We decided for the participants to focus on math problems while monitoring the visual stimulus in the AR headset because mental math calculation uses people's working memory [70]. Working memory falls under short-term memory and is necessary for sudden perceptual cognitive tasks, such as language comprehension and reasoning [71]. By having participants focus on a task that uses their working memory, we can examine the salient properties of the different types of visual stimulus in the AR headset. We purposefully designed our math application to be more cognitively taxing (i.e., utilize more working memory) in order to keep the participants' attention on the math application, instead of the visual stimulus in the AR headset; not primarily focusing on the information in the headset is consistent with demanding real-world settings (e.g., surgery).

Our math application consisted of single-digit x three-digit multiplication problems (Figure 3). We chose multiplication problems because they have a slower solve time compared to addition problems [72]; therefore, requiring more of the participants' attention. We also implemented other design decisions to make the application more cognitively taxing, such as presenting the single-digit first because prior work has shown that adults are slower when the smaller number is first [73]. The single-digit ranged from 2 to 9 because multiplication with 0 or 1 utilizes retrieving a rule (e.g., everything multiplied by 0 equals 0) instead of a solution [74]. The three-digit also did not end in 0 or 1, and did not include three of the same digit (e.g., 444) due to the tie-effect [75]. The tie-effect states that response time for an operand pair with identical digits is faster. None of the multiplication problems repeated in the math application.

Figure 3 shows a screenshot from our math application. The current math problem would appear at the top of the screen (e.g., 6 x 826). The participants had an area to work out the problem using a stylus pen, before inputting their answer and hitting a button to go onto the next problem. The participants had to input an answer to move on, but the answer did not have to be correct. Each participant finished the number of problems they could do in the set block time. We instructed the participants to take their time and focus on getting correct answers.

4.3. *Equipment*

Both the AR application and multiplication application were created using Unity [58]. The AR application was run on the same Meta 2 headset as the exploratory location study, and the math application was run on a Wacom Cintiq Companion Hybrid tablet [76]. The tablet has a 1080 x 1920 resolution, and the screen is 13.3 inches, measured diagonally. The physical button was the same button that was used in the location study.

4.4. *Data Analysis and Results*

We analyzed the types of visual stimulus (color, text, shapes) by examining the response time and error rate for the visual stimulus, the math solve time, and the participants' cognitive workload and preference.

4.4.1. *Response Time*

We determined response time by calculating the time it took a participant to press the button after the correct visual condition appeared in the AR headset. When calculating response time, we did not include any incorrect button hits; for example, if the participant hit the button when the square shape appeared when the correct condition was a triangle. A Shapiro-Wilks test on response time per visual stimulus showed that the data was non-normal ($W = 0.97$, $p < 0.01$). We applied a log-transform [77] to the distributions and used the transformed data for analysis, but the mean response times we report are the actual measured values. A Mauchly Test for Sphericity showed that the data had equal variances. A one-way repeated measures ANOVA (RM-ANOVA) found a significant main effect of type of visual stimulus (color, text, shapes) on response time ($F_{2,70} = 34.84$, $p < 0.0001$). A Bonferroni post-hoc comparison showed that the participants had a significantly faster response time for color ($M = 1.33$s, $SD = 0.37$), than shapes ($M = 1.67$s, $SD = 0.4$) or text ($M = 1.92$s, $SD = 0.54$).

We further analyzed response time by examining all separate visual conditions (color red, color green, circle shape, triangle shape, "red" text, and "blue" text) (Figure 4). A Shapiro-Wilks test on response time per visual condition showed that the data was skewed ($W = 0.95$, $p < 0.0001$). We applied a log-transform [77] to the distributions and used the transformed data for analysis, but the mean response times reported in the paper are the actual measured values. A Mauchly Test for Sphericity showed that the data did not have equal variances ($p <$

0.05); therefore, we applied a Greenhouse-Geisser correction. A one-way RM-ANOVA found a significant main effect of type of visual condition (color red, color green, circle shape, triangle shape, "red" text, and "blue" text) on response time ($F_{3.99,139.41} = 23.03$, $p < 0.0001$). A Bonferroni post-hoc comparison showed that the participants had a significantly faster response time for color red (M = 1.17s, SD = 0.4) and a significantly slower response time for "blue" (M = 2.14s, SD = 0.68) when compared to all other visual conditions. There was no significant difference between the circle shape (M = 1.62s, SD = 0.59) and triangle shape (M = 1.69s, SD = 0.48). Altogether, participants had the fastest response time for color, more specifically for the color red, and had the slowest response time for text.

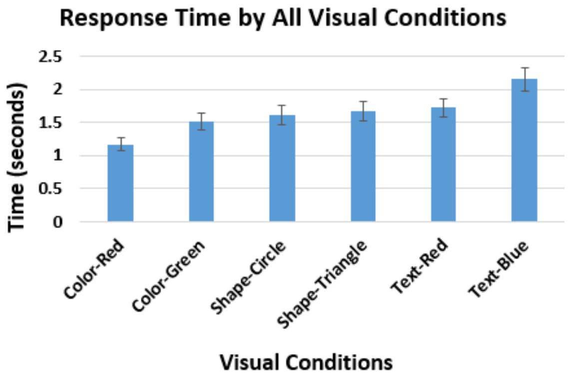


Figure 4: Average response time for all visual conditions. Error bars indicate 95% confidence interval.

4.4.2. *Error Rate*

In our study, we examined three possible types of errors: not hitting the button when the correct visual condition appeared (*missing the correct visual condition*), hitting the button for the wrong visual condition (*wrong button hit*), and overall error rate (i.e., combining the two types). A Shapiro-Wilks test on error rate per visual stimulus found that the data was severely skewed for all three types: missing the correct visual condition (W = 0.68, $p < 0.0001$), wrong button hit (W = 0.56, $p < 0.0001$), and overall error rate (W = 0.77, $p < 0.0001$). For overall error rate, a Mauchly Test for Sphericity found that the data met the assumption of equal variances; therefore, we applied the Aligned Rank Transform [78]. A one-way RM-ANOVA found no significant effect of visual stimulus type on overall error rate ($F_{2,70} = 2.89$, *n.s.*). All types of visual stimulus had a low error rate: color (1.65%), text (1.3%), and shapes (1.04%).

Next, we analyzed when the participants missed the correct visual condition. A Mauchly Test for Sphericity found that the data met the assumption of equal variances; therefore, we applied the Aligned Rank Transform [78]. A one-way RM-ANOVA found no significant effect of visual stimulus type on missing the correct visual condition ($F_{2,70} = 0.54$, *n.s.*). We analyzed the wrong hit errors (e.g., hitting the button for the text "green", when the condition was "blue"). A Mauchly Test for Sphericity showed that the data did not have equal variances ($p < 0.0001$). Since the data was not normal and did not meet the assumption of equal variances we conducted a Friedman test, which found a significant main effect of visual stimulus on hitting the button for the wrong condition ($\chi^2(2) = 9$, $p < 0.05$). A Bonferroni post-hoc comparison showed that color had significantly more wrong hit errors than shapes and text; there was no difference between text and shapes.

To examine the wrong hit errors further, we analyzed all separate visual conditions (color red, color green, circle shape, triangle shape, "red" text, and "blue" text). A Mauchly Test for Sphericity showed that the data did not have equal variances ($p < 0.0001$). Since the data was not normal and did not meet the assumption of equal variances we conducted a Friedman test, which found a significant main effect of type of visual conditions on wrong hit errors ($\chi^2(5) = 34.8$, $p < 0.0001$). A Bonferroni post-hoc comparison found that only color green had significantly more wrong hit errors than all other visual conditions. There was a total of 39 wrong hit errors, and 66.7% (26/39) of them were during the color green condition. During the color conditions, the colors red, green, blue, and yellow would cycle in the headset. We found that 96.15% (25/26) of the wrong hits during the color green condition were hit when the color was yellow. Participants frequently commented on having a hard time differentiating yellow and green. For instance, P6 stated "*Color required the least mental thought with the exception that the yellow and green are similar, and when I saw yellow I had to make sure it wasn't green.*" This is most likely due to the AR headset display quality, as it can affect users' color perception [24,26]. Generally, the three types of visual stimulus did not have a significant difference in error rate and participants discerned the correct visual condition, resulting in low error rates.

4.4.3. *Math Solve Time*

We examined the participants' math solve time to investigate if participants were focusing on the math application for each visual stimulus type. We were not interested in how fast the participants completed the math problems since we told the participants to take their time, but rather analyzed the math solve time to make sure they were consistently focusing on the math application between the different study blocks. A Shapiro-Wilks test on response time per visual stimulus showed that the data was non-normal (W = 0.85, $p < 0.0001$). We applied a log-transform [77] to the distributions and used the transformed data for analysis. A Mauchly Test for Sphericity showed that the data had equal variances. A one-way RM-ANOVA found no significant effect of visual stimulus type on math solve time ($F_{2,70} = 0.56$, *n.s.*). The consistency in solve time between the different types of visual stimulus corroborates that participants considered the multiplication problems their main task.

4.4.4. Perceived Cognitive Workload

The participants completed the weighted NASA TLX [65] for each type of visual stimulus (color, text, shapes). A Shapiro-Wilks test found that the data was normal, and a Mauchly Test for Sphericity showed that the data met the assumption of equal variances. A one-way RM-ANOVA found a significant main effect of type of visual stimulus on perceived cognitive workload ($F_{2,70} = 8.97$, $p < 0.001$). A Bonferroni post-hoc comparison found that text (M = 45.08, SD = 17.33) had a significantly higher perceived cognitive workload compared to color (M = 36.46, SD = 18.35) and shapes (M = 38.43, SD = 17.04); there was no significant difference between color and shapes. The participants found text to be more cognitively demanding.

4.5. Subjective Preference

At the end of the study, we asked the participants to rank the three types of visual stimulus in their order of preference: from most preferred to least preferred (1 to 3). A Friedman test found a significant relationship between preference rank and type of visual stimulus ($\chi^2(2) = 47.2$, $p < 0.0001$). A Bonferroni post-hoc comparison found that color (M = 1.31, SD = 0.52) and shapes (M = 1.81, SD = 0.52) were ranked significantly higher than text (M = 2.89, SD = 0.40). Color was highly preferred, with 72% (26/36) of the participants ranking it as their first choice. Text was overwhelmingly the least preferred, with 92% (33/36) of the participants ranking it last. The participants preferred color and shapes over text for critical information.

4.6. Discussion

For our experiment on critical information, we found that existing perception findings can be applied to AR headsets. Our results are consistent with literature on visual perception [23,79], which shows a hierarchy of salient visual features. People can more easily detect color, followed by shapes, and then text. Participants in our study frequently remarked about not having to divert their focus away from the math problems to discern the different colors. AR headsets can lead to difficulties in object recognition and legibility, as well as impact users' color perception [24]. Having an inaccurate perception of color comprises the users' interpretation of display elements which is necessary in contexts that rely on color-coding (e.g., military). Therefore, understanding that we can utilize prior work on visual features helps in designing salient critical information for AR headsets. Also, understanding how to design salient information in AR headsets is important because visual salience can help working memory [80], which is crucial during complex tasks. The AR headset system for helping farmers operate and monitor agricultural machines in [39] provided warnings in the display if something was wrong with the machines. The warning consisted of the word "ALERT" in black text on a red background in the periphery of the display. While the authors use color, it might be more beneficial to utilize a distinct shape instead of text. Also, the authors should place the warning in the center location instead of the periphery to increase saliency.

In addition, examining differences between specific conditions led to further insight into how to design critical information for AR headsets. When we analyzed the two separate visual conditions for color (red and green), we found a significant difference in response time. The participants had a faster response time for color red when compared to all other visual conditions, including color green, which is consistent with prior work [81]. Color also had significantly more wrong hit errors than shapes and text, which we found was due to the color green condition. The participants frequently confused yellow and green, which strengthens the argument for utilizing high contrast elements in AR [28], especially for critical information. Yellow and green are analogous colors, as they are grouped next to each other on the color wheel. Designers should consider color choice for critical information in AR headsets and avoid using analogous colors to denote separate information.

Shapes were the second most effective in aiding awareness for critical information. The majority of the participants (25/36) ranked shapes as their second choice, and shapes had the second fastest participant response time. One interesting factor that the participants mentioned was looking for specific aspects of the shapes to determine if it was the correct visual condition, such as corners. For example, P16 stated "*Shapes were a mix, since looking for points like a triangle could cause you to mix up shapes like stars as well.*" In addition to designers avoiding analogous colors, designers should also avoid using shapes that have similar characteristics (e.g., points).

Text was the least effective in helping awareness for critical information; the participants had the slowest response time, highest cognitive workload, and lowest preference for text. The majority of the participants (33/36) ranked text as their least preferred, due to having to pay more attention and actually read the text. We did find a significant difference in response time between the two visual conditions in which the participant had to press the button ("red" and "blue"), which aligns with prior work on processing times based on word length [69]. Our study confirms that designers should consider word length when designing for critical information in AR headsets. Depending on the context, including text in an AR headset may be necessary to effectively communicate the desired information. For instance, prior work recommends that text should not be completely removed for instructions, as it can lead to fewer errors and faster learning times [50].

We recommend that there needs to be a balance between providing the information and not cognitively overloading the user. Based on our findings, designers should incorporate more visually salient information (e.g., color, shapes) when possible, for critical information in AR headsets to aid in situational awareness.

5. Experiment 2: Secondary Information

While visual saliency is essential for critical information, secondary information does not have this restriction, and therefore can display larger quantities of detailed information. Since secondary information does not need to be as visually salient, there are more opportunities to integrate it with the environment. Therefore, we examined three textual presentation styles for secondary information: locked to the display view (Display), located in the environment (Environment), and a mix of both (Mixed Environment). We chose to study text because it is commonly used to denote information in AR headset applications and may be necessary depending on the context [14,23,30,44,46]. This experiment is an extension of prior work [29].

5.1. Participants

We had 33 adults participate in our study (M = 21.55 years, SD = 3.55); however, we excluded one male participant due to equipment failure and two participants (one female, one male) for self-reported peripheral vision loss. Therefore, we had 30 participants (M = 21.63 years, SD = 3.69) for analysis, which consisted of seventeen males, twelve females, and one non-binary participant. Two of the participants were left-handed, seventeen had prior experience with AR headsets, and all had normal or corrected-to-normal vision.

5.2. Method and Design

During the study, participants completed multiplication problems on a touchscreen tablet (same as our first study) while viewing textual secondary information in an AR headset in three different presentation styles. The participants took part in the study for approximately 60 minutes in a windowless room, and either volunteered without compensation or received extra credit in a course they were taking.

The structure of the study was similar to our first experiment. After consenting to participate, participants completed a demographic questionnaire. The participants then completed a 5-minute practice round of multiplication problems on the tablet without wearing the AR headset to get comfortable with the math application; which was not used in analysis. After the math practice, participants put on the AR headset and completed three study blocks (5-minutes each), one per presentation style (Display, Environment, Mixed Environment). After completing a block, the participants would take the NASA TLX [65] for that presentation style, and then we would ask the participants to recall the last textual information displayed in the headset. We did not explain beforehand that we would ask them to recall the last presented information in the headset, which allowed us to examine if there was a difference in perception and comprehension. We counterbalanced the order of the presentation styles across participants.

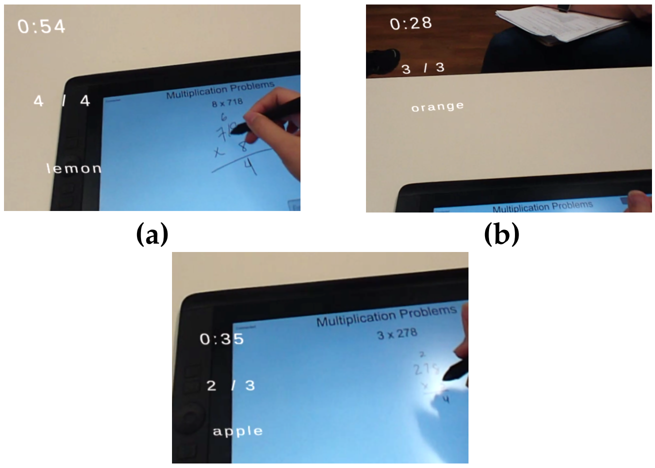


Figure 5: Presentation styles in AR headset: (a) Display, (b) Environment, and (c) Mixed Environment (Images from [29]).

5.2.1. *Secondary Textual Information*

The content of the secondary textual information included the participant's average math problem completion time, the participant's math accuracy, and a random word (Figure 5). Before the start of each study, we explained the content and how it would be presented in the headset to the participant; the information was always presented in the same order. Like in our first study, the text was in Liberation Sans font since prior work recommends using sans-serif fonts for readability [67]. Each textual element was placed 75 mm apart.

The participant's average math problem completion time was presented in minutes and seconds (e.g., "2:36"). The math accuracy was the number of correct problems over total completed (e.g., "2/3"). Both the average completion time and accuracy were updated in real-time after the participant completed a problem. For the random word, the text would randomly cycle between four words: "apple", "banana", "lemon", and "orange". Each word would remain in the headset for a random time interval (20 to 40 seconds) before switching. We used the random word as a substitute for information that may be necessary but not essential for situational awareness.

5.2.2. *AR Textual Presentation Styles*

The three different presentation styles included: Display, Environment, and Mixed Environment. For the *Display* presentation style, the textual information was locked to the left-hand side of the AR headset field-of-view and superimposed over the users' environment (Figure 5a). As mentioned earlier, we decided to place the secondary information on the left-side because we did not want the quantity of the information to block the participant's view, there was no significant difference in response time between the center and left-side locations in our exploratory study, and people exhibit pseudoneglect (i.e., leftward visual bias) [60,61]. Same as our first experiment, text height was 5 mm and white, which is consistent with recommendations [66]. The information was always present in the AR headset field-of-view, which is different than the *Environment* presentation style. For the Environment style, the text was superimposed and fixed in the environment to the left of the participant; therefore, it was more conformal to the environment. Consistent with design recommendations, the text was 500 mm away from the participant with a height of 10 mm [66]. In Figure 5b of the Environment style, the participant is looking directly at the text in the headset. If the participant looked away from the text (e.g., down or to the right) they would not be able to see the information. The *Mixed Environment* presentation style was a mix of both of the previous styles. The text was superimposed and always present in the AR headset field-of-view (same as Display) but was 500 mm away from the participant with a height of 10 mm (same as Environment). Although the text was always present in the headset, having the text 500 mm away from the participant placed the textual information more into the participant's central view (Figure 5c).

5.3. Equipment

Both the AR application and multiplication application were created using Unity [58]. The AR application was run on the same Meta 2 AR headset [55] as the previous studies, and the math application was run on the same Wacom tablet [76] as the critical information study.

5.4. Data Analysis and Results

We analyzed the presentation styles by investigating participants' accuracy of recalled information, math solve time, cognitive workload, and subjective preference.

5.4.1. Information Recall

After each study block, we asked the participants to recall the last textual information that was displayed in the AR headset (i.e., average math time, math accuracy, random word). The participants were unaware that we were going to ask this information; therefore, they were truly unsuspecting for the first study block but became aware for the remaining two study blocks. To capture this difference we split our analysis into two categories: *first response* recall (first presentation style) and *habituated response* recall (other styles). *First response* recall captures the natural perceptibility of the presentation styles, while *habituated response* recall is more aligned with real-life settings in which the users are conscious of what information they have to monitor. For both *first response* and *habituated response* recalls, we calculated the percentage of correct answers for each participant's presentation style. A participant's answer had to directly match the last information presented in the AR headset to be considered correct. A Shapiro-Wilks test showed that the data was non-normal for both *first response* recall ($W = 0.85$, $p < 0.001$) and *habituated response* recall $W = 0.87$, $p < 0.0001$). They both met the assumption of equal variances, so we applied the Aligned Rank Transform [78].

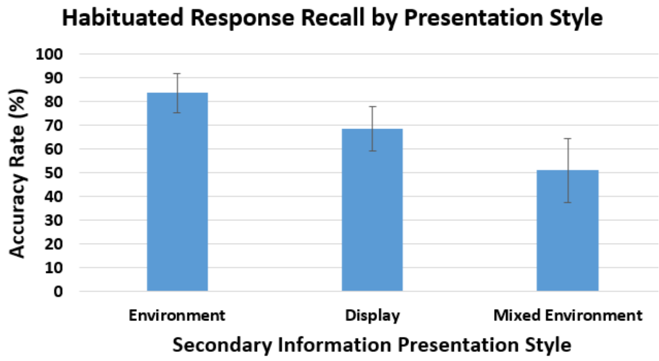


Figure 6: Habituated response recall accuracy by secondary information presentation style. Error bars represent 95% confidence interval.

For *first response* recall, a one-way ANOVA found no significant effect of type of presentation style on recall accuracy ($F_{2,27} = 1.64$, *n.s.*); there was no significant difference between the Display (M = 60.0%, SD = 21.1%), Environment (M = 67.5%, SD = 23.7%), and Mixed Environment (M = 50.0%, SD = 20.4%) styles. However, for *habituated response* recall, a one-way RM-ANOVA found a significant main effect of type of presentation type on recall accuracy ($F_{2,44.05} = 8.91$, $p < 0.0001$). A Bonferroni post-hoc test showed that participants had a significantly higher *habituated response* recall accuracy for the Environment style (M = 83.8%, SD = 18.6%) than the Mixed Environment style (M = 51.2%, SD = 30.9%) (Figure 6); there was no significant difference between the Environment and Display (M = 68.8%, SD = 21.3%) styles. We examined each secondary information separately, and only found a significant effect of presentation style on the random word accuracy ($p < 0.001$, Fisher's exact test [82]). A pairwise test of independence with a Bonferroni correction only found a significant difference between the Environment and Mixed Environment styles. The Environment style had a higher number of correct recall events for the random word than the Mixed Environment style (18 correct vs. 6 correct).

5.4.2. Math Solve Time

As in the first study, we examined the participants' math solve time to investigate if participants were focusing on the math application for each presentation style. A Shapiro-Wilks test found that the data was severely skewed ($W = 0.70$, $p < 0.0001$) and a Mauchly Test for Sphericity showed that the data did not have equal variances ($p < 0.0001$). Since the data was not normal and did not meet the assumption of equal variances we conducted a Friedman test, which did not find a significant effect of presentation style on math solve time ($\chi^2 (2) = 3.27$, *n.s.*). The consistency in solve time between the different types of presentation styles corroborates that participants considered the math as their main task.

5.4.3. *Perceived Cognitive Workload*

We analyzed the participants' perceived cognitive workload for each presentation style. A Shapiro-Wilks test on cognitive workload per presentation style showed that the data was non-normal ($W = 0.96$, $p < 0.05$), and the data met the assumption of equal variances using a Mauchly Test for Sphericity; therefore, we applied the Aligned Rank Transform [78]. A one-way RM-ANOVA found no significant effect of type of presentation style on perceived cognitive workload ($F_{2,58} = 0.12$, *n.s.*). There was no significant difference in participants' perceived cognitive workload between the three presentation styles: Display ($M = 31.5$, $SD = 17.5$), Environment ($M = 31.73$, $SD = 16.04$), and Mixed Environment ($M = 30.9$, $SD = 13.89$).

5.4.4. *Subjective Preference*

After the participants completed the study blocks, we asked them to rank the three types of presentation styles in their order of preference: from most preferred to least preferred (1 to 3). A Friedman test found no significant relationship between preference rank and type of presentation style ($\chi^2 (2) = 4.2$, *n.s.*). There was no significant difference in preference between the styles: Display ($M = 1.9$, $SD = 0.8$), Environment ($M = 1.8$, $SD = 0.85$), and Mixed Environment ($M = 2.3$, $SD = 0.75$).

5.5. *Discussion*

The only significant difference we found between the three presentation styles (Display, Environment, Mixed Environment) was for *habituated response* recall accuracy. The Environment presentation style had a higher *habituated response* recall accuracy than the Mixed Environment style. With further examination, we only found a significant effect of presentation style on the random word accuracy. During the study, participants frequently mentioned that they were more interested in the math completion time and accuracy, instead of the random word, since they were related to the main task and their performance. Therefore, the Environment presentation style resulted in higher user perceptibility, since it aided in the awareness of information that did not capture the participants' attention (i.e., random word).

For the Mixed Environment style, having the textual information further into the participants' central field of vision made it more distracting to the participants. For example, P13 stated *"The mixed environment was too distracting and put too much pressure on me to get more problems right"*, and P21 stated *"It [The Mixed Environment style] would block some of my work or the math problem."* With the Environment presentation style, the participants could look at the information when they wanted. One participant (P8) stated *"[The Environment style] didn't get in my way so I didn't have to block it out of my vision while completing the math problems. It was nice to look up at it when I felt the need to."* Although the Display style was always present in the headset field-of-view like the Mixed Environment style, participants remarked that it was easier to disregard since it was more in the periphery. Both the Display and Environment presentation styles allowed the participants to view, as well as tune out, the secondary information whenever they preferred. This resulted in a stronger focus when the participants did pay attention to the information, which led to higher recall accuracy.

Both the Display and Environment styles aided user perception and comprehension for secondary textual information. However, prior work has found that text notifications locked to the field-of-view in both virtual reality (VR) [83] and AR [84] headsets result in a higher sense of urgency. For instance, the authors in [83] found that text notifications locked to a VR headset display resulted in users viewing them as more imperative than text notifications in the environment. Also, the authors in [84] conducted a study examining different locations of notifications in an AR headset display during social conversations. The notifications placed in the direct center of the field-of-view were perceived as urgent and intrusive, when compared to notifications that were slightly offset. Secondary information should be subtle and in the background, while, as mentioned in our first experiment, critical information should be salient and require promptness [27]. Therefore, we recommend that AR headset application designers use the Environment style for supplementary secondary information and the Display style for information that requires urgency to increase users' perception and comprehension.

6. Limitations and Future Work

Our results provide insight into the design of critical and secondary information in AR headset displays for users' situational awareness (i.e., perception and comprehension); however, there are some limitations. First, we only examined three types of visual stimulus for critical information and three secondary information presentation styles. In addition, we mainly focused on perception (level 1). Future work can analyze different types of information, as well as comprehension (level 2) and prediction (level 3). Another limitation for the first study was that the visual stimulus constantly changed, which could have prompted the participants to be more aware and closely monitor the stimulus. Prior work has found that *motion* has a high perception accuracy for peripheral visual notifications on computer screens (i.e., participants were able to quickly perceive the notifications), but it can also distract from the primary task [59,85]. Although the constant change may have prompted the participants, it does not detract from our main goal of comparing different types of stimuli.

7. Conclusion

We conducted two studies on how to present critical and secondary information in AR headsets to aid in users' situational awareness: one examining if existing findings on the perceptibility of three types of visual stimulus (color, text, shapes) can be applied to AR headsets for critical information, and one analyzing three presentation styles for textual secondary information (Display, Environment, Mixed Environment). Our results showed that the Display and Environment presentation styles improved the awareness of textual secondary information; participants had a higher recall of information when compared to the Mixed Environment presentation style. For critical information, we found similar results to prior work; the participants perceived color faster, and had a slower response time and higher cognitive workload for text. We contribute design recommendations on how to present critical and secondary information in AR headset displays to aid in users' situational awareness, which is essential to understand in safety crucial domains such as the military and healthcare.

Institutional Review Board Statement

All participants consented before they participated in the studies. The studies were conducted in accordance with the Declaration of Helsinki, and the protocols were approved by the University of Florida Institutional Review Board.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This work is partially supported by National Science Foundation Grant Award #IIS-1750840 and the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1842473. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect these agencies' views.

References

[1] D.W.F. van Krevelen, R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations," *The International Journal of Virtual Reality*, vol. 9, no. 2, pp. 1–20, 2010, doi:10.1155/2011/721827.

[2] S. Vardomatski, Augmented And Virtual Reality After Covid-19, https://www.forbes.com/sites/forbestechcouncil/2021/09/14/augmented-and-virtual-reality-after-covid-19/?sh=1b886d882d97, 2021.

[3] A. Olney, Augmented Reality | All About Holograms, American Library Association: 1–16, 2019.

[4] Meta, Meta Augmented Reality, https://www.metavision.com/, Apr. 2019.

[5] Augmented Reality Is A Game Changer For Oil & Gas, https://oilprice.com/Energy/Energy-General/Augmented-Reality-Is-A-Game-Changer-For-Oil-Gas.html, May 2022.

[6] M.R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors*, vol. 37, no. 1, pp. 32–64, 1995, doi:10.1518/001872095779049543.

[7] M.R. Endsley, "Design and Evaluation for Situation Awareness Enhancement," *Proceedings of the Human Factors Society Annual Meeting*, vol. 32, no. 2, pp. 97–101, 1988, doi:10.1177/154193128803200221.

[8] N.A. Stanton, P.R.G. Chambers, J. Piggott, "Situational Awareness and Safety," *Safety Science*, vol. 39, no. 3, pp. 189–204, 2001, doi:10.1016/S0925-7535(01)00010-8.

[9] T.W. Reader, P. O'Connor, "The Deepwater Horizon Explosion: Non-technical Skills, Safety Culture, and System Complexity," *Journal of Risk Research*, vol. 17, no. 3, pp. 405–424, 2014, doi:10.1080/13669877.2013.815652.

[10] C.M. Schulz, V. Krautheim, A. Hackemann, M. Kreuzer, E.F. Kochs, K.J. Wagner, "Situation Awareness Errors in Anesthesia and Critical Care in 200 Cases of a Critical Incident Reporting System," *BMC Anesthesiology*, vol. 16, no. 4, pp. 10pp, 2016, doi:10.1186/s12871-016-0172-7.

[11] E. Gans, D. Roberts, M. Bennett, H. Towles, A. Menozzi, J. Cook, T. Sherrill, "Augmented Reality Technology for Day/Night Situational Awareness for the Dismounted Soldier," in: Desjardins, D. D., Sarma, K. R., Marasco, P. L., and Havig, P. R., eds., in *Display Technologies and Applications for Defense, Security, and Avionics*, International Society for Optics and Photonics: Article 9470, 2015, doi:10.1117/12.2177086.

[12] Z. Zhu, V. Branzoi, M. Wolverton, G. Murray, N. Vitovitch, L. Yarnall, G. Acharya, S. Samarasekera, R. Kumar, "AR-Mentor: Augmented Reality Based Mentoring System," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR '14)*, IEEE: 17–22, 2014, doi:10.1109/ISMAR.2014.6948404.

[13] M. Wallmyr, T.A. Sitompul, T. Holstein, R. Lindell, "Evaluating Mixed Reality Notifications to Support Excavator Operator Awareness," in *IFIP Conference on Human-Computer Interaction (INTERACT ' 19)*, Springer, Cham: 743–762, 2019, doi:10.1007/978-3-030-29381-9_44.

[14] D. Liu, S.A. Jenkins, P.M. Sanderson, M.O. Watson, T. Leane, A. Kruys, W.J. Russell, "Monitoring with Head-Mounted Displays: Performance and Safety in a Full-Scale Simulator and Part-Task Trainer," *Anesthesia & Analgesia*, vol. 109, no. 4, pp. 1135–1146, 2009, doi:10.1213/ANE.0b013e3181b5a200.

[15] M.T. Pascale, P. Sanderson, D. Liu, I. Mohamed, B. Brecknell, R.G. Loeb, "The Impact of Head-Worn Displays on Strategic Alarm Management and Situation Awareness," *Human Factors*, vol. 61, no. 4, pp. 537–563, 2019, doi:10.1177/0018720818814969.

[16] S. Ruano, C. Cuevas, G. Gallego, N. García, "Augmented Reality Tool for the Situational Awareness Improvement of UAV Operators," *Sensors*, vol. 17, no. 2, pp. Article 297, 2017, doi:10.3390/s17020297.

[17] B.-J. Park, C. Yoon, J.-W. Lee, K.-H. Kim, "Augmented Reality Based on Driving Situation Awareness in Vehicle," in *International Conference on Advanced Communication Technology (ICACT'15)*, IEEE: 593–595, 2015, doi:10.1109/ICACT.2015.7224865.

[18] D. Aschenbrenner, N. Maltry, J. Kimmel, M. Albert, J. Scharnagl, K. Schilling, "ARTab - Using Virtual and Augmented Reality Methods for an Improved Situation Awareness for Telemaintenance," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 204–209, 2016, doi:10.1016/J.IFACOL.2016.11.168.

[19] K.-H. Kim, K.-Y. Wohn, "Effects on Productivity and Safety of Map and Augmented Reality Navigation Paradigms," *IEICE TRANSACTIONS on Information and Systems*, vol. E94-D, no. 5, pp. 1051–1061, 2011, doi:10.1587/transinf.E94.D.1051.

[20] L. Hou, X. Wang, "A Study on the Benefits of Augmented Reality in Retaining Working Memory in Assembly Tasks: A Focus on Differences in Gender," *Automation in Construction*, vol. 32, , pp. 38–45, 2013, doi:10.1016/J.AUTCON.2012.12.007.

[21] M.-C. Chen, R. Klatzky, "Displays Attentive to Unattended Regions: Presenting Information in a Peripheral-Vision-Friendly Way," *International Conference on Human-Computer Interaction (HCII'07)*, vol. 4551, , pp. 23–31, 2007.

[22] C.G. Healey, K.S. Booth, J.T. Enns, "High-Speed Visual Estimation Using Preattentive Processing," *ACM Transactions on Computer-Human Interaction*, vol. 3, no. 2, pp. 107–135, 1996, doi:10.1145/230562.230563.

[23] Y. Ishiguro, J. Rekimoto, "Peripheral Vision Annotation: Noninterference Information Presentation Method for Mobile Augmented Reality," in *Proceedings of the Augmented Human International Conference (AH'11)*, ACM Press, New York, New York, USA: 1–5, 2011, doi:10.1145/1959826.1959834.

[24] E. Kruijff, J.E. Swan, S. Feiner, "Perceptual Issues in Augmented Reality Revisited," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR '10)*, IEEE: 3–12, 2010, doi:10.1109/ISMAR.2010.5643530.

[25] M.A. Livingston, J.L. Gabbard, J.E. Swan, C.M. Sibley, J.H. Barrow, Basic Perception in Head-Worn Augmented Reality Displays, Springer New York, New York, NY: 35–65, 2013, doi:10.1007/978-1-4614-4205-9_3.

[26] J.L. Gabbard, J.E. Swan, J. Zedlitz, W.W. Winchester, "More Than Meets the Eye: An Engineering Study to Empirically Examine the Blending of Real and Virtual Color Spaces," in *IEEE Virtual Reality Conference (VR '10)*, IEEE: 79–86, 2010, doi:10.1109/VR.2010.5444808.

[27] M.R. Endsley, "Designing for Situation Awareness in Complex System," in *Proceedings of the Second International Workshop on Symbiosis of Humans, Artifacts and Environment*, 14pp, 2001.

[28] S. Ganapathy, Design Guidelines for Mobile Augmented Reality: User Experience, Springer New York, New York, NY: 165–180, 2013, doi:10.1007/978-1-4614-4205-9_7.

[29] J. Woodward, J. Smith, I. Wang, S. Cuenca, J. Ruiz, "Examining the Presentation of Information in Augmented Reality Headsets for Situational Awareness," in *ACM International Conference on Advanced Visual Interfaces (AVI '20)*, ACM Press: 1–5, 2020, doi:10.1145/3399715.3399846.

[30] S. Lukosch, H. Lukosch, D. Datcu, M. Cidota, "Providing Information on the Spot: Using Augmented Reality for Situational Awareness in the Security Domain," *Computer Supported Cooperative Work (CSCW '15)*, vol. 24, no. 6, pp. 613–664, 2015, doi:10.1007/s10606-015-9235-4.

[31] S. Zollmann, C. Hoppe, T. Langlotz, G. Reitmayr, "FlyAR: Augmented Reality Supported Micro Aerial Vehicle Navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 560–568, 2014, doi:10.1109/TVCG.2014.24.

[32] J. Fromm, K. Eyilmez, M. Baßfeld, T.A. Majchrzak, S. Stieglitz, "Social Media Data in an Augmented Reality System for Situation Awareness Support in Emergency Control Rooms," *Information Systems Frontiers*, vol. 25, no. 3, pp. 303–326, 2021, doi:10.1007/S10796-020-10101-9.

[33] Minh Tien Phan, I. Thouvenin, V. Fremont, "Enhancing the Driver Awareness of Pedestrian Using Augmented Reality Cues," in *IEEE International Conference on Intelligent Transportation Systems (ITSC '16)*, IEEE: 1298–1304, 2016, doi:10.1109/ITSC.2016.7795724.

[34] M. Colley, B. Eder, J.O. Rixen, E. Rukzio, "Efects of Semantic Segmentation Visualization on Trust, Situation Awareness, and Cognitive Load in Highly Automated Vehicles," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '21)*, Association for Computing Machinery: 11pp, 2021, doi:10.1145/3411764.3445351.

[35] A. Rowen, M. Grabowski, J.P. Rancy, A. Crane, "Impacts of Wearable Augmented Reality Displays on Operator Performance, Situation Awareness, and Communication in Safety-Critical Systems," *Applied Ergonomics*, vol. 80, , pp. 17–27, 2019, doi:10.1016/J.APERGO.2019.04.013.

[36] J. Coleman, D. Thirtyacre, "Remote Pilot Situational Awareness with Augmented Reality Glasses: An Observational Field Study," *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 8, no. 1, pp. 10pp, 2021, doi:https://doi.org/10.15394/ijaaa.2021.1547.

[37] H. Chen, L. Hou, G. (Kevin) Zhang, S. Moon, "Development of BIM, IoT and AR/VR technologies for Fire Safety and Upskilling," *Automation in Construction*, vol. 125, no. 103631, 2021, doi:10.1016/J.AUTCON.2021.103631.

[38] M. Bhattarai, A.R. Jensen-Curtis, M. Martinez-Ramon, "An Embedded Deep Learning System for Augmented Reality in Firefighting Applications," in *IEEE International Conference on Machine Learning and Applications (ICMLA '20)*, IEEE: 1224–1230, 2020, doi:10.1109/ICMLA51294.2020.00193.

[39] J. Huuskonen, T. Oksanen, "Augmented Reality for Supervising Multirobot System in Agricultural Field Operation," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 367–372, 2019, doi:10.1016/J.IFACOL.2019.12.568.

[40] E.R. Velamkayala, M. V. Zambrano, H. Li, "Effects of HoloLens in Collaboration: A Case in Navigation Tasks," *Human Factors and Ergonomics Society Annual Meeting*, vol. 61, no. 1, pp. 2110–2114, 2017, doi:10.1177/1541931213602009.

[41] J. Blattgerste, B. Strenge, P. Renner, T. Pfeiffer, K. Essig, "Comparing Conventional and Augmented Reality Instructions for Manual Assembly Tasks," in *Proceedings of the International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '17)*, ACM Press, New York, New York, USA: 75–82, 2017, doi:10.1145/3056540.3056547.

[42] S. Wang, M. Parsons, J. Stone-McLean, P. Rogers, S. Boyd, K. Hoover, O. Meruvia-Pastor, M. Gong, A. Smith, "Augmented Reality as a Telemedicine Platform for Remote Procedural Training," *Sensors*, vol. 17, no. 10, pp. Article 2294, 2017, doi:10.3390/s17102294.

[43] D. Datcu, S. Lukosch, H. Lukosch, "Comparing Presence, Workload and Situational Awareness in a Collaborative Real World and Augmented Reality Scenario," *IEEE ISMAR Workshop on Collaboration in Merging Realities (CiMeR '13)*, pp. 6pp, 2013, doi:https://research.tudelft.nl/.

[44] R. Rzayev, P.W. Wozniak, T. Dingler, N. Henze, "Reading on Smart Glasses: The Effect of Text Position, Presentation Type and Walking," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'18)*, ACM Press, New York, New York, USA: 9pp, 2018, doi:10.1145/3173574.3173619.

[45] A. Albarelli, A. Celentano, L. Cosmo, R. Marchi, "On the Interplay between Data Overlay and Real-World Context using See-through Displays," in *Proceedings of the Biannual Conference on Italian SIGCHI Chapter (CHItaly'15)*, ACM Press, New York, New York, USA: 58–65, 2015, doi:10.1145/2808435.2808455.

[46] J. Orlosky, K. Kiyokawa, H. Takemura, "Managing Mobile Text in Head Mounted Displays: Studies on Visual Preference and Text Placement," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 2, pp. 20–31, 2014, doi:10.1145/2636242.2636246.

[47] S. Kim, M.A. Nussbaum, J.L. Gabbard, "Influences of Augmented Reality Head-Worn Display Type and User Interface Design on Performance and Usability in Simulated Warehouse Order Picking," *Applied Ergonomics*, vol. 74, , pp. 186–193, 2019, doi:10.1016/J.APERGO.2018.08.026.

[48] D. Ariansyah, J.A. Erkoyuncu, I. Eimontaite, T. Johnson, A.M. Oostveen, S. Fletcher, S. Sharples, "A Head Mounted Augmented Reality Design Practice for Maintenance Assembly: Toward Meeting Perceptual and Cognitive Needs of AR Users," *Applied Ergonomics*, vol. 98, , pp. 103597, 2022, doi:10.1016/J.APERGO.2021.103597.

[49] M. Gattullo, L. Dammacco, F. Ruospo, A. Evangelista, M. Fiorentino, J. Schmitt, A.E. Uva, "Design Preferences on Industrial Augmented Reality: A Survey with Potential Technical Writers," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR '20)*, IEEE: 172–177, 2020, doi:10.1109/ISMAR-ADJUNCT51615.2020.00054.

[50] N. Irrazabal, G. Saux, D. Burin, "Procedural Multimedia Presentations: The Effects of Working Memory and Task Complexity on Instruction Time and Assembly Accuracy," *Applied Cognitive Psychology*, vol. 30, no. 6, pp. 1052–1060, 2016, doi:10.1002/ACP.3299.

[51] K. Sekiguchi, A.A. Nugraha, Y. Du, Y. Bando, M. Fontaine, K. Yoshii, "Direction-Aware Adaptive Online Neural Speech Enhancement with an Augmented Reality Headset in Real Noisy Conversational Environments," in *International Conference on Intelligent Robots and Systems (IROS '22)*, 9266–9273, 2022, doi:10.1109/IROS47612.2022.9981659.

[52] S. Debernardis, M. Fiorentino, M. Gattullo, G. Monno, A.E. Uva, "Text Readability in Head-Worn Displays: Color and Style Optimization in Video versus Optical See-Through Devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 1, pp. 125–139, 2014, doi:10.1109/TVCG.2013.86.

[53] J.J. Dudley, J.T. Jacques, P.O. Kristensson, "Crowdsourcing Design Guidance for Contextual Adaptation of Text Content in Augmented Reality," in *SIGCHI Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery: 14pp, 2021, doi:10.1145/3411764.3445493.

[54] L. Stearns, L. Findlater, J.E. Froehlich, "Design of an Augmented Reality Magnification Aid for Low Vision Users," in *International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS' 18)*, ACM: 28–39, 2018, doi:10.1145/3234695.3236361.

[55] R. Brown, Meta 2: Full Specification, https://vr-compare.com/headset/meta2, May 2022.

[56] S. Ando, N. Kida, S. Oda, "Central and Peripheral Visual Reaction Time of Soccer Players and Nonathletes," *Perceptual and Motor Skills*, vol. 92, no. 3, pp. 786–794, 2001, doi:10.2466/pms.2001.92.3.786.

[57] G. Berlucchi, W. Heron, R. Hyman, G. Rizzolatti, C. Umiltà, "Simple Reaction Times of Ipsilateral and Contralateral Hand to Lateralized Visual Stimuli," *Brain*, vol. 94, no. 3, pp. 419–430, 1971, doi:10.1093/brain/94.3.419.

[58] Unity, https://unity3d.com/, 2022.

[59] A. Mairena, C. Gutwin, A. Cockburn, "Peripheral Notifications in Large Displays: Effects of Feature Combination and Task Interference," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'19)*, ACM Press, New York, New York, USA: 12pp, 2019, doi:10.1145/3290605.3300870.

[60] D. Bowers, K.M. Heilman, "Pseudoneglect: Effects of Hemispace on a Tactile Line Bisection Task," *Neuropsychologia*, vol. 18, no. 4–5, pp. 491–498, 1980, doi:10.1016/0028-3932(80)90151-7.

[61] N.A. Thomas, O. Schneider, C. Gutwin, L.J. Elias, "Dorsal Stream Contributions to Perceptual Asymmetries," *Journal of the International Neuropsychological Society : JINS*, vol. 18, no. 2, pp. 251–259, 2012, doi:10.1017/S1355617711001585.

[62] S.D. Christman, C.L. Niebauer, "The Relation Between Left-Right and Upper-Lower Visual Field Asymmetries: (Or: What Goes Up Goes Right While What's Left Lays Low)," *Advances in Psychology*, vol. 123, , pp. 263–296, 1997, doi:10.1016/S0166-4115(97)80076-3.

[63] A.K. Smith, I. Szelest, T.E. Friedrich, L.J. Elias, "Native Reading Direction Influences Lateral Biases in The Perception of Shape From Shading," *Laterality*, vol. 20, no. 4, pp. 418–433, 2014, doi:10.1080/1357650X.2014.990975.

[64] ISO 7010:2011 Graphical symbols, https://www.iso.org/standard/54432.html, 2011.

[65] S.G. Hart, "Nasa-Task Load Index (NASA-TLX); 20 Years Later," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006, doi:10.1177/154193120605000909.

[66] P. Mealy, Planning Your Augmented Reality Project, John Wiley & Sons, Inc. : 159–161, 2018.

[67] L. Rello, R. Baeza-Yates, "How to Present More Readable Text for People with Dyslexia," *Universal Access in the Information Society*, vol. 16, no. 1, pp. 29–49, 2017, doi:10.1007/s10209-015-0438-8.

[68] D.J. McKeefry, N.R.A. Parry, I.J. Murray, "Simple Reaction Times in Color Space: The Influence of Chromaticity, Contrast, and Cone Opponency," *Investigative Opthalmology & Visual Science*, vol. 44, no. 5, pp. 2267–2276, 2003, doi:10.1167/iovs.02-0772.

[69] V. Kuperman, H. Stadthagen-Gonzalez, M. Brysbaert, "Age-Of-Acquisition Ratings for 30,000 English Words," *Behavior Research Methods*, vol. 44, no. 4, pp. 978–990, 2012, doi:10.3758/s13428-012-0210-4.

[70] R.H. Logie, K.J. Gilhooly, V. Wynn, "Counting on Working Memory in Arithmetic Problem Solving," *Memory & Cognition*, vol. 22, no. 4, pp. 395–410, 1994, doi:10.3758/BF03200866.

[71] A. Baddeley, "Working Memory," *Science*, vol. 255, no. 5044, pp. 556–559, 1992, doi:10.1126/SCIENCE.1736359.

[72] K. Miller, M. Perlmutter, D. Keating, "Cognitive Arithmetic: Comparison of Operations," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, no. 1, pp. 46–60, 1984, doi:10.1037/0278-7393.10.1.46.

[73] J.I.D. Campbell, D.J. Graham, "Mental Multiplication Skill: Structure, Process, and Acquisition.," *Canadian Journal of Psychology*, vol. 39, no. 2, pp. 338–366, 1985, doi:10.1037/h0080065.

[74] P. Lemaire, M. Fayol, "The Role of Working Memory Resources in Simple Cognitive Arithmetic," *European Journal of Cognitive Psychology*, vol. 8, no. 1, pp. 73–103, 1996.

[75] G.J. Groen, J.M. Parkman, "A Chronometric Analysis of Simple Addition," *Psychological Review*, vol. 79, no. 4, pp. 329–343, 1972, doi:10.1037/h0032950.

[76] Wacom, Professional Hybrid Creative Tablet User's Manual About the Cintiq Companion Hybrid, 1–93, https://cdn.wacom.com/f/manual/0x0307/UM-EN.pdf, May 2022.

[77] O.N. Keene, "The Log Transformation is Special," *Statistics in Medicine*, vol. 14, no. 8, pp. 811–819, 1995, doi:10.1002/sim.4780140810.

[78] J.O. Wobbrock, L. Findlater, D. Gergle, J.J. Higgins, "The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*, ACM Press, New York, New York, USA: 143–146, 2011, doi:10.1145/1978942.1978963.

[79] C. Healey, J. Enns, "Attention and Visual Memory in Visualization and Computer Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 7, pp. 1170–1188, 2012, doi:10.1109/TVCG.2011.127.

[80] M.S. Fine, B.S. Minnery, "Visual Salience Affects Performance in a Working Memory Task," *Journal of Neuroscience*, vol. 29, no. 25, pp. 8016–8021, 2009, doi:10.1523/JNEUROSCI.5503-08.2009.

[81] A. Batra, D. Jeph, S. Vyas, "Color Difference on Simple Visual Reaction Time in Young Volunteers," *International Journal of Clinical and Experimental Physiology*, vol. 1, no. 4, pp. 311–313, 2014, doi:10.4103/2348-8093.149771.

[82] H.-Y. Kim, "Statistical Notes for Clinical Researchers: Chi-squared Test and Fisher's Exact Test.," *Restorative Dentistry & Endodontics*, vol. 42, no. 2, pp. 152–155, 2017, doi:10.5395/rde.2017.42.2.152.

[83] R. Rzayev, S. Mayer, C. Krauter, N. Henze, "Notification in VR: The Effect of Notification Placement, Task, and Environment," in *Computer-Human Interaction in Play (CHIPlay'19)*, ACM Press: 199–211, 2019, doi:10.1145/3311350.3347190.

[84] R. Rzayev, S. Korbely, M. Maul, A. Schark, V. Schwind, N. Henze, "Effects of Position and Alignment of Notifications on AR Glasses during Social Interaction," in *Proceedings of the Nordic Conference on Human-Computer Interaction (NordiCHI '20)*, ACM Press, New York, NY, USA: 11pp, 2020, doi:10.1145/3419249.

[85] C. Gutwin, A. Cockburn, A. Coveney, "Peripheral Popout: The Influence of Visual Angle and Stimulus Intensity on Popout Effects," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*, ACM Press, New York, New York, USA: 208–219, 2017, doi:10.1145/3025453.3025984.

Received: 21 December 2022, Revised: 09 February 2023, Accepted: 16 February 2023, Online: 27 March 2023
DOI: https://dx.doi.org/10.55708/js0203002

Coding: First Steps from Kindergarten up to Primary School

Elisa Benetti*, **Gianluca Mazzini**
Lepida ScpA, Via della Liberazione 15, 40128, Bologna, Italy
*Corresponding author: Elisa Benetti & elisa.benetti@lepida.it

ABSTRACT: Computational thinking is now featured in many school curricula around the world. It is in fact defined as the "new English", emphasizing its universally recognized indispensability. Despite this, the subject is almost never addressed until primary school where, however, hours dedicated to it are often too limited. Our first training proposal, including basic coding concepts in kindergarten, led to better results than expected in terms of children's understanding and involvement. Our field training has led to a refinement and expansion of the program in these past three years. The primary objective is to begin the study of coding at the age of three, when the foundations of logical thinking are actually already present, due to get to the writing of the first programs in pseudocode and analysis of programming languages at the end of elementary school. All methodologies used are chosen on the basis of the possibility of following a single logical trend, which gradually increases the concepts to be learned and their difficulty, but always starting from already known bases, previously addressed. This allows to optimize learning times by minimizing the necessary hours and human resources and still obtaining the desired results. In addition to not burdening the number of hours available, a further firm point was not to burden schools economically either: costs were in fact always achievable without any problems. Having no impact either on the budget, or on the number of hours, or on the required staff makes this program easily feasible for any school.

KEYWORDS: Coding, Kidergarten, Primary School

1. Introduction

The concept of computational thinking was first introduced by Seymour Papert in the book Mindstorms [1], published in 1980, where in his theory of learning, known as Constructivism and based on the LOGO language he invented, he states how the computer is an important new medium for learning. Indeed, it is seen not only as a machine with which you can process information, but a tool for building, manipulating, learning, discovering and even making mistakes. A very important point of his theory for the purposes of the work presented in this article, in fact, is that error is not seen in a negative light, but as a constructive aspect of the learning process. To err is to explore in search of alternative solutions to the problem. More recently, in 2006, this concept has been taken up by computer scientist and MIT professor Jeanette Wing, who defines it as follows, "Computational thinking is a process of formulating problems and solutions in a form that is executable by an agent who processes information." [2] Indeed, we know that computers are used to solve problems, but even before solving a problem, it is important to understand by what means it can be solved, and it is computational thinking that enables us to do this.

The revolution of this concept is to note that it is not only important to solve problems, but more importantly to understand them, in order to formulate a process that leads to its resolution. This process can be performed by an agent or executor, figures that we will see in the exercises used, who implements instructions in a mechanical and unconscious manner, replicating human thinking. Computational thinking has been used in programming for the longest time, but recently also in coding, as well as in educational robotics. Tools used are not only technological and related to these disciplines, but also in normal life situations involving the decomposition of a problem, just to emphasize its importance in any aspect of life. Every day, without even realizing it, we find ourselves deciding on the expression of a solution by instructions and the execution of those instructions: finding the shortest route to a place, executing a recipe, assembling an object, assembling constructions. All of those listed are processes that involve computational thinking and presuppose a set of precise, orderly, clear, and repeatable instructions that will enable an effective solution to be reached by whoever is executing them. These instructions represent a de facto algorithm, that will certainly lead to the solution and can be applied to another identical problem with the same result, just as is the case with the algorithms behind any type of programming: from making decisions in a video game to performing an Internet search to managing interpersonal relationships on a smartphone.

Thus, computational thinking can be described by three main stages:

1. Abstraction: formulation of the problem;
2. Automation: expression of the solution;
3. Analysis: execution of the solution and evaluation.

All this is done by starting with the ability to break down a complex problem into several parts so that it can be tackled more easily. In this, coding is similar to mathematics: it is the logic of everything that works in a programmable way as mathematics is the logic of numbers and figures.

The International Society for Technology in Education [3], too, further highlights its importance in school education, pointing out that computational thinking allows people to:

1. Represent problem data through specific models;
2. Organize problem data in a logical manner;
3. Formulate and analyze problems so that they can be solved by a performer, computer or human;
4. Segment solutions into sequences of ordered, accurately described steps, or automate them through algorithms;
5. Identify possible solutions to implement the one that is the most effective and efficient in terms of effort and resources;
6. Abstract such processes for solving similar problems;

Computational thinking therefore is not only closely related to computer science and it is essential to develop it from an early age. One of the most effective ways to do this is to cultivate it through the use of coding tools, the process of writing languages and instructions intended for machines. Coding also constitutes a practical and immediate way to apply the theory of computational thinking and its previously illustrated steps and tools, while having fun, leaving room for creativity and imagination, while learning a new language and a new way of seeing problems and situations and expressing one's ideas and solutions clearly and effectively, thus also developing one's intelligence and critical thinking.

Coding, is not the only way to develop, or apply, computational thinking, but it has proven to be particularly effective because of the immediacy, interactivity, variety, availability and versatility of available tools. For this reason, its presence in school curricula is now worldwide recognized as indispensable as also explained in [4]. However, the training solutions and proposals available to date have been designed in limited areas. In fact, they are often focused only on the specific skills of an age: a lot of work has been done to exploit cognitive skills already in the pre-school context. For example, [5] shows how a coding course has led to an objective increase in problem solving skills and cognitive abilities in 4 and 5 year old children. On the other hand, children of various elementary schools in [6] used the Code.org site showing how, after following 8 coding activities, also in this case cognitive abilities increased, and not only that: children began to spend more time to planning, with increased ability to solve standardized planning tasks and in many cases even led to an inhibition of overbearing responses. In other researches, such as [7], the age analyzed is broader but the tool used is very specific, in this case apps. Both [8] and [9] searched in several elementary schools with a wider time range but in both cases using only the Scratch application. Therefore, in the existing literature, there are no proposals that lead to a general use for a long period. Our proposal therefore has the objective of choosing some of the tools available, designed for the different abilities relating to growing ages but sufficiently similar to allow the development of a fluid and continuous coding educational path. New complexities are, in fact, gradually added along the way, following this list of concepts gradually taught, starting from 3 years of age, reaching 11 years: instruction and sequence, coding through colors and symbols, algorithm, constraints, transcoding, programming a robot, programming on tablets or devices analogues, programming a hardware device using block programming, first approaches to a real programming language.

2. Dissemination of coding at school

The spread of coding in schools around the world has been gradual. Many countries have already implemented coding as a subject in primary education programs for years, recognizing its importance, and we can see the most striking examples, summarized in table 1.

Table 1: Summary of coding dissemination in the world

Singapore	2014
France	2014
Denmark	2014
Spain	2015
UK	2015
Estonia	2015
Slovakia	2015
Philippines	2015
Australia	2015
Belgium	2016
Finland	2016
Poland	2016
Portugal	2016
United Arabian Emirates	2017
Qatar	2020
South Africa	2020
Italy	2021
Kenya	2022

Australia as early as 2015 has noticed a growing need for technology talent recognizing that the future of work is toward technology, as seen in [10]. In order to properly train its younger generation by providing the technological knowledge necessary for their future, it started early to allocate significant economic sums. To reach kindergarten and basic education on programming, the Australian government spends 64 million dollars to fund school-based STEM (Science, Technology, Engineering, and Mathematics) and early learning initiatives under the Inspiring All Australians in Digital Literacy and STEM measure. As early as 2018, coding teaching from elementary school onward is mandatory in Australia.

In Asia, on the other hand, Singapore, which adopted computer science education in 2014 as shown in [11], quickly made it a compulsory subject, as early as 2020 and giving it a lot of space in school curricula: coding, in fact, is usually done for 10 hours per week. The Singapore government, in 2017, released 3 million dollars allocated for the distribution of 100,000 coding pocket gadgets to school children before the start of compulsory coding and spent annual budget allocations for the program. Malaysia, Thailand, Vietnam, and Indonesia have already been investing economically for years in their turn for the dissemination of coding. In contrast, an Asian country that has more recently introduced programming as a subject in primary and secondary schools is the Philippines. Although the government, unlike those previously mentioned, spends much less on programming education, pupils show much interest in programming and are willing to learn programming in school. Interviews from 2015 already showed that about 97 percent of students in the Philippines were interested in learning about programming and 96 percent wanted programming to become a core subject in their schools.

South Africa was the first African country to adopt coding education at primary and secondary levels [12]. In 2020, it began by providing programs for teachers to learn how to teach programming as a first step in order to be able to pass the same knowledge on to students. On the other hand, Software Engineer turns out to be the most in-demand job in South Africa to date, and for this reason the government in South Africa is paying a lot of attention to technical education, understanding how necessary it is to equip the younger generation with relevant technical skills to adapt when it comes to future jobs. In the recent August 2022, the Kenyan government also announced the inclusion of programming as a subject in its educational curriculum for primary and secondary school pupils, becoming the second African country to adopt programming education at primary and secondary levels.

More than 90 percent of parents in the United Arabian Emirates wanted their children taught programming in schools as early as 2017, and in year 2020, about 35 percent of schools in the country have begun implementing programming courses for their students. The Arab country has immediately begun to transform the entire education system by adopting the use of e-books, robotics, and other emerging technologies in teaching and learning. Other countries have also more recently begun efforts to prepare new generations for the technological revolution that will shape future jobs, for example, Qatar since 2020 has been restructuring its education system to include programming.

In Europe, the first effort aimed at the importance of coding across a broad spectrum, is undoubtedly codeweek, [13], launched in 2013. The European Commission supports European Programming Week as part of its Digital Single Market strategy. In its Digital Education Action Plan, it especially encourages schools to join the initiative. European Programming Week is an event therefore aimed primarily at schools but not only, celebrating creativity, problem solving and collaboration through programming and other technology activities. The idea is to make programming more visible, show young people, adults and older people how to bring their ideas to life with programming, explain these skills and bring motivated people together to learn. The latest statistics regarding the event that took place in 2021 show that 4 million people from more than 80 countries around the world participated in European Programming Week. The average age of the participants was 11 years old, and 49 percent of them in 2021 were women or girls. Eighty-eight percent of European Programming Week events took place in schools, showing that efforts to strengthen teachers, during the 2021 campaign, were successful. Another experience is provided by the web site All you need is C<3DE , through which the European Coding Initiative has supported thousands of teachers across Europe in their efforts to integrate programming and coding teaching with a collection of open online courses, teaching materials, tools and lesson plans. To understand the uptake in compulsory schools it helps the JRC, which, in March 2022, published a new report [14] which examines the integration of computational thinking in compulsory schools in 29 countries, European and non-European: 18 european and 7 non-european countries have already made the teaching of coding compulsory, of the remaining 4 Denmark is carrying out a pilot initiative, while Italy, Slovenia and the Czech Republic have policies in this direction.

In fact, even Italian Parliament finally seems to have become convinced of the need and urgency, necessary requirements for law decrees, to also include coding as a basic learning skill The National Recovery and Resilience Plan(PNRR) is the plan approved in 2021 by Italy to revive its economy after the COVID-19 pandemic in order to enable the country's green and digital development. Among the measures foreseen with regard to schools can be found that, as of the school year 2025/2026, it will be mandatory in schools of all grades and levels to pursue the development of digital skills, including by fostering the learning of computer programming (coding), within the existing teachings. Albeit, quoting the text here, "with the human, instrumental and financial resources available under current legislation and in any case without or greater burdens on public finance". These resources, both human and financial, are far from substantial in Italy. In proposals regarding coding, therefore, it must also be taken into account that they should not be economically costly and should not weigh excessively on available school hours either. That is, programs must be proposed that optimize the number of hours needed with respect to the skills acquired by students, using material that is as low-cost as possible.

We have therefore seen how the importance of coding in the school curriculum is recognized worldwide and both strategic and economic plans are being implemented everywhere for its diffusion. Australia has been among the first to leave, as early as 2015. In Asia, Singapore in 2014, the Philippines in 2015 and the United Arab Emirates in 2017 started early, with a subsequent slowdown. However, since the Code for Asia project [15] was born in 2021, other countries are quickly aligning, such as Malaysia, Thailand, Vietnam and Indonesia. In Europe, the beginning was given by the codeweek in 2013, following which, between 2014 and 2017, most European countries began to promote initiatives for the diffusion of coding. Italy was among the last to join, in 2021. The continent that moved last was Africa, where South Africa was the first to introduce coding into primary and secondary schools in 2020, followed last year by

Kenya. In 2022, the AltSchool digital campus was born, with a purely technological curriculum, also attracting interest in Nigeria, Ghana, Uganda and Botswana. CodingAfrica [16] was also born in 2022, with the aim of promoting tech literacy in the rest of Africa. Our effort in trying to introduce coding already in preschool age started in 2019, with the first year of a pilot project in a kindergarten in the province of Bologna. These almost four years of proactive field experience have allowed us to already have a concrete and easily implementable solution in any school, even without any previous experience since teachers are provided with both the list of necessary materials and a series of lessons, ideas, exercises and software/hardware creations to copy or draw inspiration from. This opportunity is today very important in our territory, given the mandatory nature of coding in Italian school curricula by 2025/2026.

2.1. Gender Gap in STEM

We devote a final space for reflection to how the introduction of coding in education can also help overcome existing gender gaps.

In Iraq, for example, coding has been, since 2020, a tool used for a dual purpose: to also help the gender gap present in education. Indeed, there are still to this day both strong regional differences within the country and more widespread structural, social and cultural barriers that prevent girls from fully and equally accessing and completing their education, thus making it more difficult for them to participate in the employment sector and also in society as a whole. To address this problem, Mercy Hands for Humanitarian Aid, in partnership with Mercy Hands Europe and with support from the Canadian Fund for Local Initiatives, has implemented an innovative project to strengthen girls' technology skills in Basra through computer and coding courses, benefiting both female students and teachers. The goal of the project is to improve girls' IT and programming skills by giving them more employability while training teachers on IT and coding in public schools in Shatt al-Arab.

In Europe there is a similar gender gap related to STEM (Science, Technology, Engineering and Mathematics) subjects. In fact, as explained in [17] and [18], there is a critical gender gap in STEM areas at all levels of education and the labor market. Various research and statisticsshow that globally, women obtain 53 percent of STEM university degrees, but in the EU only 34 percent of graduates in the field are women. Moreover, in 2018, only 41 percent of EU scientists and engineers were women, and only five EU member states had more women scientists than men: Lithuania, Bulgaria, Latvia, Portugal, and Denmark. Finally, note that there is also a "gender equality paradox," whereby women are less likely to obtain STEM degrees in wealthier societies with greater gender equality, such as Finland and Sweden.

Particularly in Italy we are at the bottom of the European rankings for female participation in the digital economy and society. To make up for this, fortunately already for a number of years there have been many organizations dedicated to promoting a better image of science and technology subjects and building a real sisterhood among girls who engage in these areas. Notable among them is definitely Girls who code [19], whose mission is to change the stereotypical image of the programmer. Girls who code, instead, offers several free courses and classes on programming and women working in the technology area. Another successful initiative is Coding Girls [20], born in 2014 and supported by the U.S. Diplomatic Mission in Italy, the Ministry of Education, University and Research, Roma Capitale and Microsoft. In subsequent editions, the project has grown to shape itself as an augmented educational program to train the next generation in STEAM, but more importantly, to help young female students gain confidence in science and navigate the careers of the future. There are also Girls Code it better project clubs, which organize workshops at various secondary schools in grades I and II. Girlstart, projectCSGIRLS and Technovation girls also carry out similar operations.

3. Coding in kindergarten and primary school

There are many reasons why primary school children should be taught programming, some of which have already been discussed in the preceding paragraphs. Among first reasons is the great ability of children in being able to learn new notions quickly, which is why from an early age they are induced to learn new foreign languages. If we consider the enormous influence that technological revolution is having in the world of work, learning coding and programming language allow children to have a greater understanding, from an early age, of how computers and technology work, skills that are indispensable today in the world of work as well as in everyday life. Benefits are not only technical: an additional one comes precisely from the possibility of developing, through educational coding programs, important general social and relational skills such as working in teams. Recognizing an error in a solution process helps to understand how making mistakes can be an opportunity for improvement and collaboration to reach the solution, also in an increasingly optimized way. Specifically, on the other hand, the final learning of programming languages is a great exercise for children in learning what can be called in effect a new language. Coding has proven in our experience to be an easily applicable tool as early as kindergarten because the basics of logical thinking, already broadly understood by age 3, can be taught very simply through fun games.

As we explained earlier, coding in schools started to spread as early as 8 years old, and in recent years many tools have been proposed that can be used in programs to be carried out in school and in specific events. Those chosen for this project were decided on the basis of two basic components, which are derived from the previously quoted sentence of the PNRR:

1. Possibly they must be no cost, where not possible they must be limited cost and allow with limited purchases ample opportunity for use.

2. They must fit the relative abilities of the age of the children to whom they are proposed while trying to create a uniform path with a smooth transition from one tool to the next more advanced one, all the way

from age 3 to 11, thus minimizing the hours needed to learn coding.

We then go on to illustrate first for kindergarten and then for primary school, all proposed exercises and tools chosen for this pathway.

3.1. Constraints and choices for preschool

The most obvious constraint in preschool is that children are unable to read and write. Tools used must therefore rely on colors and symbols. The need to make lessons playful is high, especially in the early approach. Finally, although devices such as smartphones and tablets are now used from the earliest years of life, the concepts of programming and computer science as they are understood in the working world are far from the minds of children of this age. It therefore becomes essential to demonstrate how logical thinking is applicable, and often unconsciously already applied, in what they do every day. In addition, in order to minimize the hours needed for coding, the proposed exercises are integrated with other activities and topics covered in the school year.

3.2. Sequences and encodings

The first approach with 3 and 4 year old children is done using two types of cards: codyfeet free and codycolor. As explained in [21] this decision was made in order to separate the concepts of sequencing and coding so that they can be learned gradually before combining them at a later stage. Interaction with children is crucial to maintain attention and the introduction is an example of this, starting by asking them if they want to guess what coding is, then suggesting that it is related to the word code. Already this first stimulus to reasoning has always brought out the link to passwords and secret codes. Asking what secret codes they know leads to the unlock code on their parents' tablet or smartphone. This already leads to the next level where teachers notice together with children that the codes are of several different types: some parents use numbers, some use signs, some use biometric data such as fingerprint or face recognition. Also when asked what they ask to unlock these devices for, the main reasons are: viewing videos on social channels or video games. This leads to the explanation that behind both there is a code that is called programming language, which programmers use. Programming is not only needed for games and applications but it can be found everywhere and the theacher can look with the children in what they do during the day or what they would like to do when they grow up, where the programming or coding is found. The amazement and interest in seeing that it exists in everything increases their interest: if they want to fly a plane they will use programmed controls, if they want to be police officers they will certainly know the coding of road signs, when they watch television it is a program that follow one cartoon to another at the same times but with different episodes, when they are in the car with mom and dad it is the programming of a "strange card called a control unit" that makes a sound to remind parents to add fuel or put on the seat belt. Once we get their attention, we return to coding related to computers and explain the first concept, that of sequence. Computer, in fact, executes a series of instructions that are given to it one after another. We must therefore as a first step decide on the right sequence of instructions to get to what we want.

To make children understand that instruction is nothing more than an action that you want them to perform, ask them to perform some elementary actions, e.g. 'Raise your hand' 'Clap your hands' ' Do no with your head'. Finally ask 'come to me jumping like a kangaroo' then having them analyze what movements they did: they stood up then performed N kangaroo jumps forward. This is a sequence. Often, without our realizing it, we are asked to do something that requires a sequence of instructions in order to do it, and we then illustrate the cody-feet free tiles that will be used to demonstrate this. The tiles are of 3 types, as shown in the Figure 1: beginning/instruction/end, and are recognizable by the way they can fit together like puzzle pieces, forming a long line of tiles, that is, a sequence. On top of the instruction tiles is a piece of velcro and it is explained that it is used to have fun playing different games depending on what you stick on it.

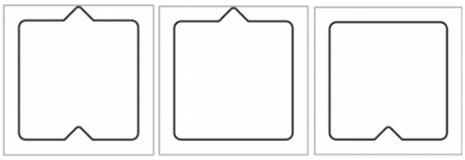


Figure 1: CodeFeetFree Cards

We can then play the first game with these tiles to explain the importance of putting the instructions in the right order in a sequence. We tell the children that their parents woke up particularly sleepy and have to get them dressed for school. As we tell in what order the clothes are put on, we create the sequence by sticking the clothes on the instruction velcro. The sequence will obviously be wrong, and we show at the end a drawing of how the parents would have sent them to school: with the underwear over the pants,the tank top over the vest, and the socks over the shoes! What should the correct sequence look like? By reasoning aloud with the group of children, they independently manage to dress correctly.

The second proposed game, on the other hand, combines coding and movement: each tile represents a movement to do, as shown in the Figure 2. Education tiles with the movements on them are made available to the children. They take turns choosing their favorite movement and, putting them in sequence, create a dance to try all together. This type of exercise can also be proposed during motor skills hours by changing the symbols as desired to create motor pathways for the children.

All the material used so far is paper-based and can safely be printed by the school at little cost. Not only that: exercises can also be suggested to be done at home with the relevant material. In fact, at the end of the illustrated lesson, suggestions of games to be played together at home were sent to parents, complete with cards, obviously of a small size, that can be printed, cut out and used. First suggested exercises were:

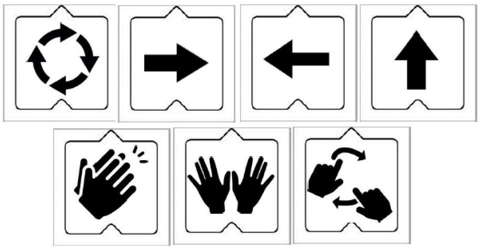


Figure 2: Dance with CodyFeetFree Cards Example

1. Practice discovering the sequences of instructions hidden in what we do: try to think together with the children about a job you do often and what is the right sequence of actions. For example to set the table, brush teeth, cook pasta. You can draw the actions together on instruction tiles and put them in sequence, then have fun shuffling them around to see how much the end result changes if you don't perform them in the correct sequence: if you drained the pasta before you turned on the stove, what would happen?
2. Dance battle: each family member proposes his or her own dance using the tiles already known from the lesson at school. The others will vote to see who is the best choreographer in the family.
3. Storytelling Inventory: tiles with good characters, bad characters, places and objects are also provided. The game is to create a sequence and invent a fairy tale by following it. A similar exercise can also be offered at school in storytelling hours, preparing tiles concerning the fairy tale that will be read and asking children to put them in the right sequence to recreate the story.

In a second lesson the term coding is introduced, starting with the explanation that it is nothing more than a simple way to explain an instruction to be executed. In the previous lesson, symbols were used, as they are also used for example on devices and remote controls: square for stop and trinagle for play are already known at age 3. It is not only symbols that are used for coding of course, and one of the most frequently used ways is certainly the use of colors, which can easily be explained to them by showing them a drawing of a traffic light. Color coding will also be used in later exercises, through the new codycolor tiles, explained in [22]: these have only one color but no interlocks. Tiles are large so they can walk on them. On gray they will step forward, on red they will turn to the right and step forward, and on yellow they will turn to the left and step forward. Since the concept of left and right is not yet clear to all children, 2 yellow and red bracelets are used to help them i understand which hand to turn toward. A first volunteer child is then sought to put the bracelets on and an object is placed on the floor: the other children in turn will have to choose the right tiles for the child to reach for the object. The teacher places the first one and the child stands on it, then it is decided where to spot in the room to go further, and which tile needs to be placed in front of the child's feet to reach the goal. The paths to take can be guided by imagination or by stories read at school, for example, one can imagine that the dragon is coming and a princess-doll needs to be collected and then placed safely in a container-castle. A variation may be, instead, to place an object in the center of the room, arrange some obstacles on the floor and form two teams that, starting from opposite corners of the room, will have to try to reach the object first.

The third and final lesson introduces the chessboard into the games. A large 5x5 chessboard is placed on the floor and they try to create a path together like the ones on the floor, but this time putting the tiles on the squares of the board. The first tile is placed by the teacher. When the path comes out of the chessboard then the teacher introduces the last two new tiles to the children, namely the start and end tiles of the path by putting the triangle before the first tile of the path and the circle at the end. Start and end now are put in retrospect, the following year they will instead become constraints to be respected: they will be set at the beginning and the children will have to make sure to create a path between the two. All the necessary information and tools for the next level have then been given.

The experience with 3 and 4-year-olds has been carried out with 3 different classes of children year after year. Every year it has been confirmed that through continuous interaction, playful exercises, and the presence of movement, the children's attention and interest always manages to remain high, often with requests from them to extend the lesson. After the first trial year, in which many parents asked for information following their children's enthusiastic stories, explanations and exercises that could be done at home were introduced. Obviously interest is subjective, but over the next two years 30 percent of parents shared with us fairy tales and dances invented together with their children or fun times when dad shaved his beard following the sequence specially scrambled by his own child. We initially expected this feedback to come from families in which at least one of both parents worked in the computer field or similar, and had clear concepts related to coding. Instead, we feel it is important to make explicit that the feedback was always related solely to the child's interest, and the parents who participated often asked for additional guidance because, completely distanced from the concept of coding and programming in their own work, they found themselves intrigued in turn.

3.3. *Consolidation of acquired knowledge and introduction of Constraints and cycles*

This second part is offered to 5 and 6-year-olds. In the first lesson we start immediately with the chessboard but using tiles that combine the two concepts of sequence and coding, cody-feet cards. As explained in [21] and [23] these have both point and wedge shapes to fit together like a puzzle, and color, with the same coding as the codycolor tiles. The main change from the previous level of difficulty is that you do not create the path before, but during. The start and end tiles are placed a priori, then 3 children proceed at a time, each with a specific task: one of them will be the executor, that is, he will only follow the instructions given to him; one will be the analyst, who will decide on the sequence of instructions to achieve the goal; and one will be the programmer, who, starting from the solution identified by the analyst, translates it into code to be executed. As much as this process has always been followed easily and naturally by children, we can see how complex it is: it involves doing teamwork while managing to keep each person in his or her specific role.

The first challenge to propose is to try to get the children to create an increasingly shorter path, until they achieve the shortest possible one. This explains the importance in achieving a goal with as few instructions as possible: it is achieved in less time and the performer gets less tired. The link to scheduling in the business world is becoming more and more evident, but there are still generalizable motivations: for example, planning better for a long journey allows one to arrive at the destination sooner, less tired and spending less money on fuel. Some planning concepts can then be explained through simple games, such as creating a tree of depth 5 by keeping the same starting point but, by choosing tiles from a predefined set, reaching different end points. By drawing each path on an A4 sheet of paper with the 5x5 checkerboard reproduced and going over it on tissue paper, the various paths found, when overlaid, will show just such a tree. Again, is sent a document to their parents, with an explanation of what was done, together with tiles and chessboards that they can print out to play with their children on A4 sheets. Some challenges proposed for home, and shown in Figure 3,are a first approach to the concept of constraint, which will later be explained at school. Using a 3x3 chessboard, how could it be filled using any tiles? How could it be filled if one had no yellow tiles? What if one had only two grays?

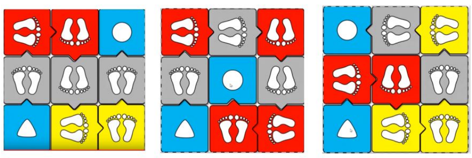


Figure 3: First Approach to Constraints Excercises

The concept of constraint in class is introduced with a google maps satellite map where they can see their school from above. Different groups of children are asked to draw the route they would take to reach a house nearby where they are having a big party. They may choose different routes, but instinctively they will follow the constraint of only being able to drive a car on one road: none of them go through a farmer's field or a park, even though doing so might take a shorter route! During exercises on the chessboard, constraints will be obstacles of various kinds to be avoided in order to reach the goal. Again, they develop concepts through themes they are dealing with in school or by linking them to a particular time of year. In this three-year project, for example, a game was proposed where the child had to program a robot to make it clean a room of garbage by managing a separate collection of items: paper, plastic and organic. As shown in the Figure 4 in order to clean up everything it will be necessary to start with one type of objects and then move on to the others.

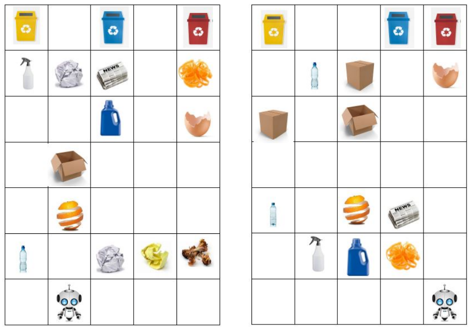


Figure 4: Garbage Collection Excercise

Both at school and at home, collection challenges have been proposed: the Easter Bunny's Easter egg collection, shown in Figure 5 where the child has to impersonate the rabbit and return to his burrow trying to put as many eggs in his basket as possible. Similarly, for the holiday season, a tale of a blizzard has been proposed at home, that dropped a lot of presents from Santa's sleigh, which he must now try to retrieve while being careful not to fly into high buildings.

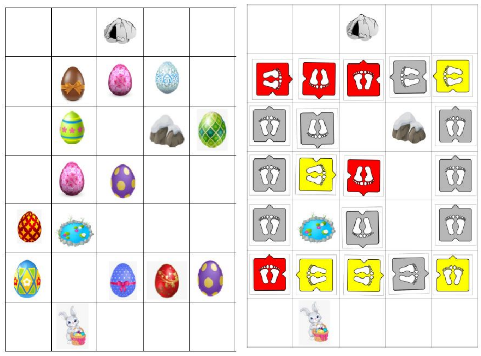


Figure 5: Easter Game Example with Solution

During the third year of the piloting of this program, the 2021/2022 school year, one section of the school was following a parallel experiment in which the section teacher, from the morning reception until the end of lunch, communicated with children only in english language. We then took advantage of this for a joint, more in-depth lesson on exercises related to one of the cycles of programming. The english teacher previously explained the key words IF THEN ELSE to the children. Next, taking a cue from the story of a bee that served as a thread for the various exercises on the board, children are divided into 3 groups. In the first group the bee, which cannot fly because it has injured a wing, will have to reach the hive while avoiding lakes and stones. In the second group the bee will be given a bulldozer and will be able to go over the boxes with stones. In the third group the bee with the boat will be able to cross the lakes. At the end of each of the three paths, the tiles used on the small checkerboards on the A4s are redrawn and overlaid on tissue paper. This time overlaying the 3 tissue papers will show the decision tree. The first real pseudocode is then made for the children to write. An A4 has already written IF <drawing bulldozer> then <free space> ELSE IF <drawing boat> THEN <free space> ELSE <free space>. In the three free spaces, the children will draw the sequence of colors they used in the three paths. This simple pseudocode got them so excited thinking they were becoming real programmers that they wanted to do it again, with the results shown in the Figure 6.

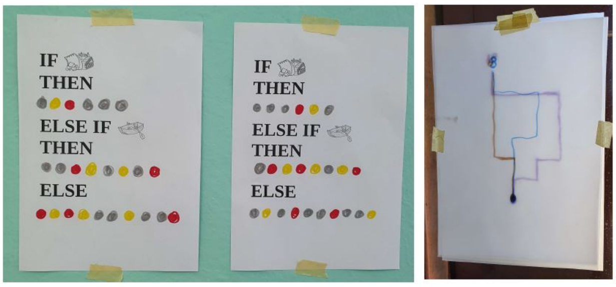


Figure 6: If Then Else Pseudocoding Example

This second phase of teaching tended to have less feedback at home: 20 percent of parents reported examples of exercises done with the children. In contrast, there was a noticeable increase during lockdown periods when schools were not open: coding was gladly exploited at home to teach in a playful way. During these periods as many as 60 percent of parents shared paths, challenges, and even proposed new ideas followed by other parents in the school. The simplicity of these tools that can be used to stimulate computational thinking was certainly confirmed by showing how it is accessible to everyone and easily transformed into fun games that can be proposed even to very young children.

4. First approaches to technology

One of the main tools of coding is visual or block programming: this type of programming offers an intuitive approach, reducing syntactic rules to simple interlocking between blocks of complementary shape. In short: the program code does not have to be typed. Even for children as young as 5 or 6, who still need to learn to read and write, visual programming thus allows them to immediately experience the effect produced by the colored blocks on the characters, called sprites, that animate the story or game being created. As children play and invent stories, they have to work hard to figure out which colored blocks to choose and fit together to recreate what they have in mind. As they do this, they unknowingly write lines of computer code. Block programming is shown as a first approach to technology.

4.1. Block coding Example

Harking back to the previous lesson, we begin by showing, on old smartphones lent by parents and given to various groups of children, a game in which, using exactly the pseudocode written with them in lesson three, the bee makes the three different paths. The mblock platform was used for this purpose, as it easily allows one to create ad-hoc blocks and visually reproduce exactly the sequence that the children see on the sheets they filled in. In figure 7 we can see on the upper part the game and on the lower the code, with the color of the instructions that the children had chosen in the different paths.

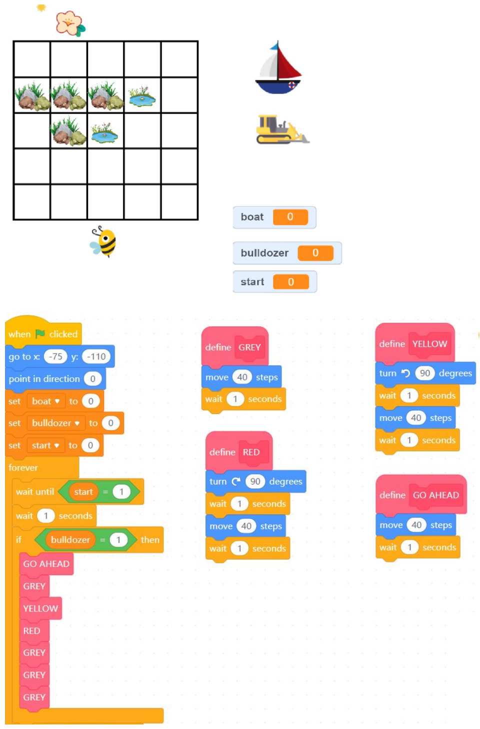


Figure 7: If Then Else Implementation with mBlock

4.2. Programming a robot

Certainly finishing kindergarten by programming a real robot was the winning choice. As explained extensively in [21], it is explained that depending on the object we want to program the coding to be used changes. In fact there are so many programming languages and often a programmer is faced with the need to transcode one of his sequences. Transcoding is necessary to transform the paths taken on the large chessboard into a sequence of keys to be pressed on a robot to make it take the same path on a chessboard of a suitable size for the length of the robot's step. The focal point of teaching at this last stage is actually the error. Easily children can make mistakes in the sequence on the robot, and at the first mistake you will cheer up the child by letting him know when the mistake is important to understand in order to correct and improve the path-it is nothing more than the debugging that every programmer does for a very high percentage of his working time! It is important to look very carefully at each step performed by the robot: when does it not do what we expect? That is exactly where there is the error that needs to be fixed!

It is also very interesting to introduce voluntary errors at 3 different points in the same sequence: at the beginning, towards the middle and almost at the end. The earlier we introduce an error the farther it will take us from the desired result.

Fortunately, also the robot was chosen not only for its easy transcoding but also for its decidedly low cost: this not only helped the school but also the parents. In fact, each year 25 to 30 percent of the seniors were so proud of the programming achievement that they asked their parents for the robot they used at school as an end-of-school gift.

5. Primary School

By primary school, manual dexterity has markedly improved, and all children are already familiar with smartphones, tablets, and interactive whiteboards at school. Until they are yet able to read and write, a less paper-based and more technological approach to coding can still be given through online applications.

Despite this, a first approach that summarized the previously assimilated concepts proved to be effective. We had further confirmation of the easy adaptability of the tools chosen in this case as well: a specific school, for example, had chosen a ship of little pirates as the theme that would also act as a leitmotif in the textbooks. During the first year it was easy to organize courses related to this area, for example:

- drive the ship between rocks and sea monsters making it arrive at the treasure island
- follow a map and reach the treasure by passing through marked key points

Adding motor skills is still important especially in the first year, therefore both courses of this type and exercises via applications on tablets or interactive whiteboards have been proposed, in parallel.

Scratch is a programming language developed by MIT (Massachusetts Institute of Technology) and made freely available. It is a block programming environment used for coding that aids in logical reasoning. In this environment there is no need to type any lines of code, but simply drag and drop blocks. The block system allows the implementation of a series of commands by simply arranging the blocks in a particular order. Each block corresponds to a command and they are executed in the order in which they were placed, from top to bottom. These features make Scratch undoubtedly one of the most popular programming languages for children.

Scratchjr is a declination dedicated to younger children and can therefore be approached in first grade. In this case blocks do not have written description of the related instruction but explain it with an intuitive picture. Available blocks are divided into 6 groups according to their functions:

1. the yellow group contains all the possibilities for starting a sequence: when you press start, when you touch a character, when a message comes to the character, and so on;
2. the blue group contains all the movements the character can do;
3. the purple group allows you to perform certain actions such as making the character talk, making him zoom in or out, making him disappear or appear;
4. the orange group contains cycles and timings;
5. the red group only indicates whether the sequence at the end should be repeated in a loop or not.

During the first year ScratchJR is then used to learn how to use blocks in sequence.

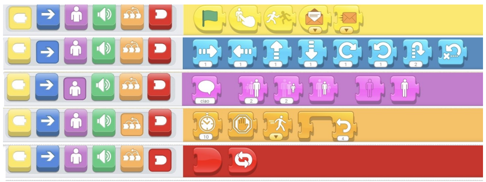


Figure 8: ScratchJR Blocks

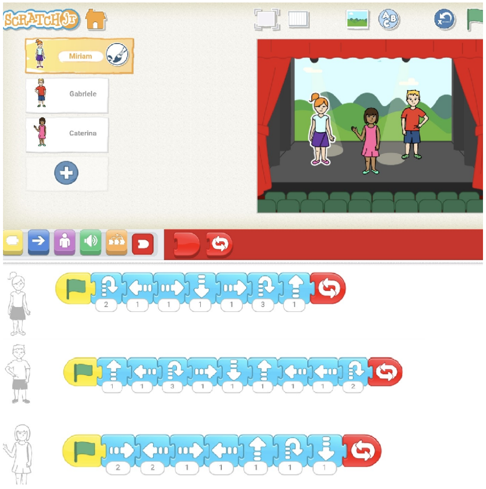


Figure 9: School Dance with ScratchJR

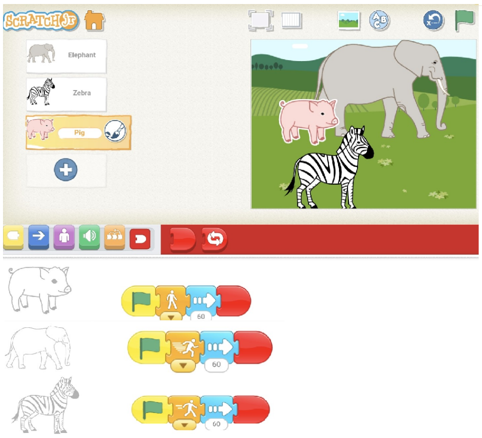


Figure 10: Race with ScratchJR

Exercises are aimed at recognizing the groups of blocks and using them creatively. The first one is, in fact, dedicated to the blue group that will lead the babies to organize the school dance. Each child chooses his character and his 10 moves. An example of an implementation performed by 3 children can be seen in Figure 9. Then, exploring the orange block, students can play a game of guessing who comes first among the chosen animals. One child in turn will choose 3 different animals, decide the speed of each one to the other children will decide which animal to bet on. In the example proposed in Figure 10 the child had the heavy elephant come first, the fast zebra second and the lazy piglet third. The level of difficulty is increased in subsequent lessons by proposing the creation of real stories. For example, this school year's class decided to fulfill the math and science teacher's dream by arranging for her to travel to the moon.

As can be seen from the Figure 11 sequence, movement, action and timing blocks were used.

As a last exercise they are stimulated in thinking of a more complex story with at least three characters interacting with each other. For example, one class proposed the following story: the child comes out of school and his mother takes him to his best friend's house to play together. Together with the teacher, the children first have to identify interactions, that is, when one character's action starts another character's sequence. In this case, the following interactions were identified:

Figure 11: Travel to the Moon with ScratchJR

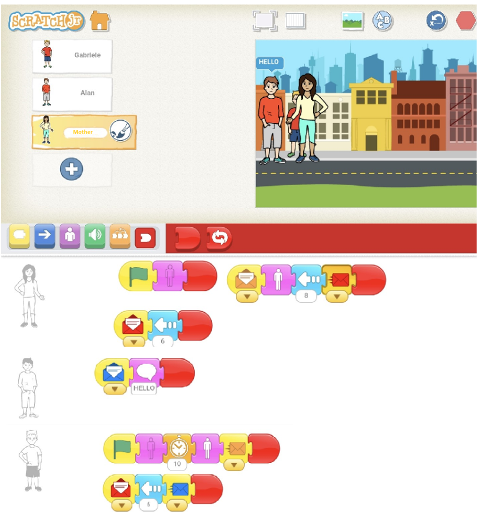


Figure 12: More Complex Interactions with ScratchJR

1. when the child appears outside school mom walks from home to school.

2. when mom comes to the child they go together to his friend's house, that is both child and mother start walking.

3. when they arrive the friend immediately says "hello".

In Figure 12 we see the implementation on scratchJR of the story. All interactions are expressed by sending colored letters, the receipt of which is the start of another sequence.

5.1. *Deepening of block programming*

With reading and writing skills established, the mBlock platform can be used in third and fourth grade. Makeblock, or mBlock [24], starts with scratch 3.0 and expands its opportunities by tying it to different types of hardware devices and the language C code hidden behind each block. Children can then approach this transitional version between the block world and real programming code. As on Scratch, blocks on mBlock also have the description of the corresponding instruction written in text. The first exercises proposed are similar to those done on scratch,for example implementing again the trip to the moon with mBlock. If the school where this program is being followed is in the same territory as the preschool and therefore there are many children who have followed the previous path, it is interesting to analyze with the newly acquired skills, the coding that the teacher had written for the exercise on the if,then,else cycle. The groups of available blocks, compared to scratchJR, obviously have many more instructions in them, and groups of operators and sound actions are added. The connection with programming languages is evident in the group of loops in which we find all the most frequently used ones.

5.2. *Exercises with advanced circuits*

Fourth grade concludes with an actual project, including hardware. The harware used can be an Arduino UNO or Elegoo UNO, there are kits for both including LEDs, sensors, displays and many other devices that can satisfy the children's creativity, at a very low cost, such that a school can buy enough of them for children to work on in small groups. The project involves developing a kit that lights up the Italian flag and plays the anthem at the push of a button and was explained specifically in [25].

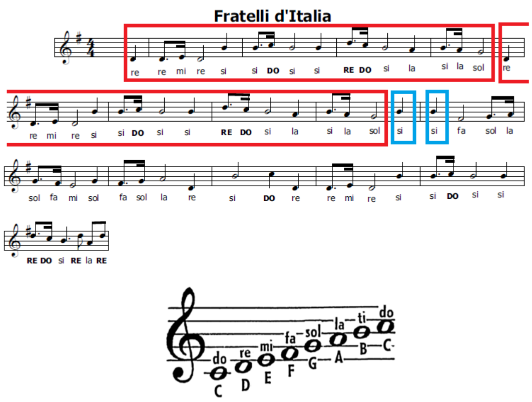


Figure 13: Looking for Repetitions and Notation Transformation

It first involves the music teacher, with whom the score of the anthem is analyzed looking for repetitions of groups of notes, as shown in Figure 13. Next comes a thorough understanding of the notes and their encoding in Anglo-Saxon notation, necessary for programming, through the transformation table in the same figure.

In Figure 14 we see the final real implementation of the project, thoroughly explained in [25]. Programming the hardware is done precisely through mBlock, already widely used by children. The Arduino UNO device is simply added to the software, and by connecting the Arduino to the PC, the program written through the blocks is loaded to it.

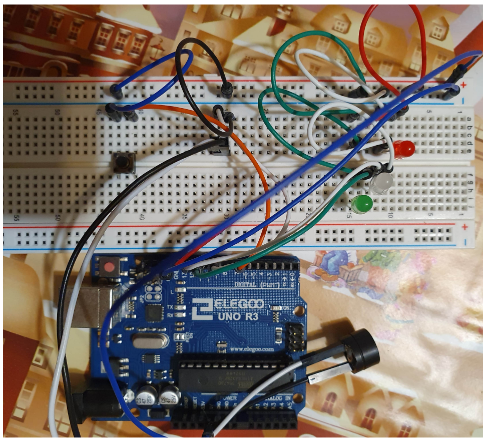


Figure 14: Real Implementation of the Anthem Project

Fifth grade is dedicated to creativity. The goal is to collaborate on the creation of an interactive landscape. The first part is devoted to analyzing what kits provided to the school include, getting inspired. The various proposals are then evaluated together, both from the point of view of feasibility and difficulty, deciding the number of people of each team that will develop the chosen ones. This project is still developing, enthusiasm is high, and many ideas have emerged, from which some choices have been made according to feasibility.

The simplest projects made by groups of two children will be as follows: The starry sky: several white and yellow LEDs with different on and off timings will be placed behind a blue veil, giving the idea of stars shining. A light sensor will be placed upstream so that the led circuit will be activated only if the sensor is in the dark or is dimmed. Design drawing and first draft of programming are respectively on the left and right side in Figure 15

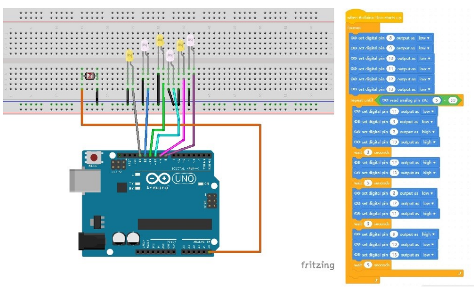


Figure 15: Starry Sky Project

Traffic management at an intersection: programming two traffic lights by reasoning about the timings needed to operate the 'intersection properly without causing accidents. The design drawing is in the left side of Figure 16, and, in the draft scheduling on the right part of the same figure, we see the timings of the two traffic lights studied through some simulations to see if they were suitable for cars to pass through without accidents.

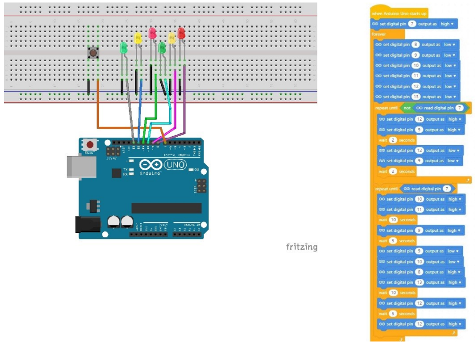


Figure 16: Traffic Lights Project

More complicated projects were then chosen to be carried out by larger groups. A group of three children will design an advertising panel in which LEDs are turned on in different ways to show different figures every minute. The main work was to understand the matrix of the LEDs and, by reproducing the designs by coloring squares on a checkerboard the size of the display used, translate them into block code.In addition, it is necessary to find the correct extension for managing the led matrix through blocks. On the left side of Figure 17 we see the design and, in Figure 18 on the right two examples of programming to make the image of a heart and an up arrow.

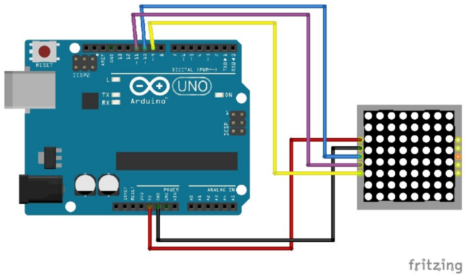


Figure 17: Advertising Panel Project

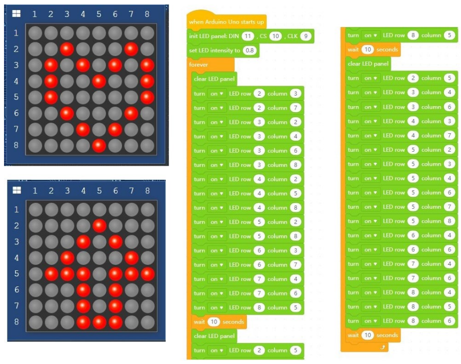


Figure 18: Matrix Programming Examples

Groups of four children will work on projects concerning the school that will be included in the landscape. The first is an automatic cooling system based on a temperature sensor that when a given threshold is exceeded activates a fan. In addition to finding the right extension to have suitable blocks to handle the sensor, this project was assigned to the children who had shown more interest in the electronic part.

In fact, of course with great help from the teacher, it was necessary to include other elements: a resistor, a diode and a transistor. In Figure 19 on the left we see the design and on the right the block programming. The second is a pad with security code for school entry. For ease of resolution, the code is a single digit. If the correct digit is pressed, the green LED lights up, otherwise the red one.

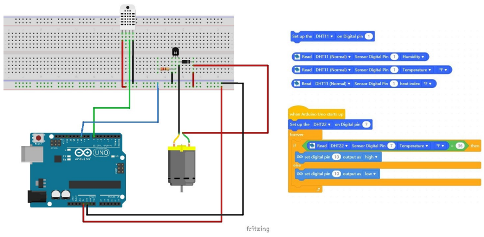


Figure 19: Cooling System Project

The main difficulty is connecting the pad and initializing it on mBlock, as well as having to find the correct extension to handle the pad here as well. Again we can see in Figure 20 on the left the design and on the right the programming.

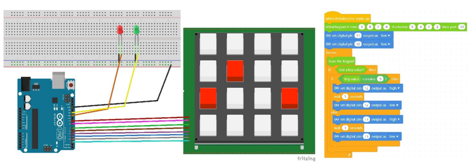


Figure 20: Security Code Pad Example

The low-cost kit proposed to the school contains many other devices not used here but which can lead to further projects that can be implemented in a primary school but with a great final functional impact. For example, it could be possible to automatically turn on the external lights of the school that is part of the project, using a proximity sensor. The cardboard model representing the school is equipped with some LEDs connected to a proximity sensor and an ultrasonic sensor is placed on the ground. When you place a hand at a predefined distance or less, all LEDs will light up for a certain number of seconds. The circuit diagram is shown in Figure 21. In Figure 22 we see a last example of a level crossing. In an infinite loop, for example every 5 minutes, the servomotor is activated so that the rod attached to it, colored to remember the level crossing, moves 90 degrees centigrade to close going into a horizontal position, lighting up the red LED parallel to the start of the movement. After 1 minute the servomotor will move 90 degrees in the opposite direction, so as to reopen returning to the vertical position, and finally the LED will turn off.

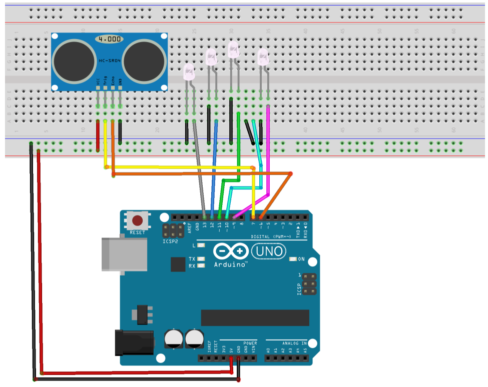


Figure 21: Switching on of Lights with Proximity Example

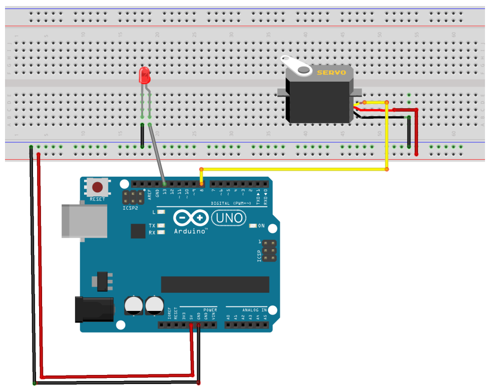


Figure 22: Level Crossing Example

The final step will be to show the programming code related to the blocks they used on mBlock and try to analyze it together, so as to have a first approach to the actual programming. The code in C language related to the hymn implementation blocks, being very simple and repetitive, lends itself well to being a first approach in which the structure of a program written in code is studied. As can be seen from the extract of Figure 23, the following can be highlighted: the inclusions of external libraries in the head of the file, the subdivision into functions, the initial setting of the variables, the main while and for loops,several calls to functions which will give an output relative to the variables that are passed as input. A more advanced analysis can instead be carried out thanks to the code relating to the exercise on the security access code. In Figure 24 students could, first of all, understand how the Pad matrix and its configuration on the hardware side are instantiated, so as to be able to uniquely encode each key pressed. The final challenge that can be proposed is to analyze the code to find the right point, shown in Figure 25, which must be changed to change the security code that should be pressed on the Pad.

```
Arduino C

#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

float button = 0;

void _delay(float seconds) {
  long endTime = millis() + seconds * 1000;
  while(millis() < endTime) _loop();
}

void setup() {
  pinMode(7,OUTPUT);
  pinMode(7,INPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(9,OUTPUT);
  while(1) {
      digitalWrite(7,1);
      while(!(!(digitalRead(7))))
      {
        _loop();
      }
      digitalWrite(11,1);
      digitalWrite(12,1);
      digitalWrite(13,1);
      for(int count=0;count<1;count++){
          tone(9,294,0.5*1000);
          delay(0.5*1000);
```

Figure 23: Analysis of the C Language of the Hymn

```
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = { 9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = { 5, 4, 3, 2}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

Figure 24: Analysis of the Pad implementation in C

```
    while(1) {
        key_value = keypad.getKey();
        if(key_value != NO_KEY){
            if(String(key_value).indexOf(String("5")) > -1){
                digitalWrite(12,1);
                _delay(1);
                digitalWrite(12,0);

            }else{
                digitalWrite(11,1);
                _delay(1);
                digitalWrite(11,0);

            }

        }
```

Figure 25: Modification of the Security Code

Each group will have its own degree of interest and understanding of the programming code, so it will be decided from time to time what type of analysis to propose in terms of difficulty and depth.

6. Conclusion

Economic investments and initiatives aimed at introducing coding into schools are present, as we have seen, in all continents: the first started in 2014, the last in 2020 but the diffusion is now widespread and is slowly becoming even mandatory in most part of the countries. In Europe, Italy is one of the last to have confirmed its willingness to invest in this topic at a political level and our aim was to move proactively with the first pilot projects to arrive at a complete and structured proposal, easily applicable in every school, in the moment in which the obligation would also arrive in our country. To our great satisfaction we have reached our goal early: compulsory education in Italy will start from the 2025/2026 school year. This work, which continues to be ongoing and expanding to this day, has always aimed to define a project that can be used in schools to incorporate coding into the school curriculum. The proposed solution respects the need to have neither additional funding nor impacts on current hours and human resources, constraints imposed by the Italian government. The choices made, in fact, took into account many key aspects that we summarize here. First of all, continuity: the main idea is a program that starts in kindergarten with 3-year-old children and continues until the end of compulsory schooling. It can be seen that all tools chosen for teaching are: on the one hand optimized for the age-appropriate skills of the students, and on the other they have a common thread that leads naturally to the next level of learning, which always has points in common with what was used previously.

The concepts of instruction, its coding through colors or symbols, the algorithm seen as a sequence of instructions that allows to reach an objective are proposed first to 3 and 4 year olds. The computational complexity of proposed exercises already increases in children aged 5 and over by adding constraints, transcoding and using symbols for programming a robot. The change of school at the age of 6 starts with a recovery of the concepts seen, through exercises on the chessboard, however combining them with their implementation of increasing complexity, on applications for touch devices, thus introducing block programming. This same block programming is finally linked to the hardware, to give life to ever-changing projects, which start from the ideas of the children themselves, and lead to the creation of something functional and interactive. The discovery of the C language code hidden behind the blocks, its analysis and modification allows students to reach the end of primary school with a suitable preparation for learning real programming languages.

Although this work is focused on kindergarten and elementary school, the same tools can be used to continue teaching in the following years, moving to the actual programming language, analyzing it, testing it, fixing it and gradually abandoning the use of the block language.

These 3 years of experimentation have brought various results:

- student interest has always been high throughout the process;
- a minimization of hours needed has been obtained, since all the necessary basic skills are already present at each change in difficulty level and teaching is dedicated only to novelty in the strict sense;
- even in the primaries in which the project was not implemented, there was evidence of how the basic notions of coding already received in kindergarten had made it possible to immediately move on to more advanced coding;
- the appropriateness of the increasing difficulty pro-

posed in parallel with increasing age was confirmed by the students' understanding;

- paper instruments or electronic devices chosen, have always been tools with a negligible cost and easily accessible by public schools;
- it was possible to consolidate a list of already tested materials to offer to schools, accompanied by lessons, course examples, application exercises and examples of Arduino hardware that can be replicated or used only as a starting point.

LepidaScpA, among all its objectives for the citizens of the Emilia Romagna region, has always had computer literacy, aimed at schools but not only, and ensuring all schools broadband connectivity to better exploit the current media technologies. The channel of communication towards schools and citizens is therefore already open on these topics and this has led us to think, as a work for the near future, of providing a portal for the exchange of information on the topic of coding. Sharing is thought of at different levels:

- publicity of training and information events by the municipalities;
- sharing initiatives in schools;
- exchange of ideas, material and examples between the teachers of the schools themselves, making the material deriving from our training proposal available first;
- exchange of information and clarifications with the students' parents, trying to involve them in the process;
- exchange of ideas, projects and collaborations between students, both from the same school and from different schools.

The basic concept of the portal is that the exchange of opinions and ideas, in this first phase of the approach that Italy is having towards coding, leads to an increase in interest with a consequent natural generalized enrichment among participants, making it a facilitating tool in the diffusion of teaching coding in schools of Emilia Romagna Region. In the hope that it will be an inspiration for the whole national territory.

Conflict of Interest All authors declare that they have no conflicts of interest.

Acknowledgment Author Elisa Benetti thanks Lepida ScpA and Gianluca Mazzini, as general manager of the company, for giving her the opportunity to teach firsthand the coding program proposed here in a pilot school. The authors thank the Primi Giochi preschool in San Pietro Capofiume, Bologna, for their willingness during the first three years of testing the program.

References

[1] S. Papert, *Children, Computers and Powerful Ideas*. New York: Basic Books, 1990. DOI: 10.1095592.

[2] J. M. Wing, "Computational Thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33-35, Mar. 2006. DOI: 10.1145/1118178.1118215.

[3] S. Boss and J. Kraus, *Reinventing Project-Based Learning: Your Field Guide to Real-World Projects in the Digital Age*. International Society for Technology in Education, 2022.

[4] S. Bocconi et al., "Developing Computational Thinking in Compulsory Education," *European Commission, JRC Science for Policy Report*, 2016, p. 68. DOI: 10.2791/792158.

[5] C. S. Iftci and A. Bildiren, "The Effect of Coding Courses on the Cognitive Abilities and Problem-Solving Skills of Preschool Children," *Comput. Sci. Educ.*, vol. 30, no. 1, pp. 3-21, 2020. DOI: 10.1080/08993408.2020.1728600.

[6] B. Arfè, T. Vardanega, and L. Ronconi, "The Effects of Coding on Children's Planning and Inhibition Skills," *Comput. Educ.*, vol. 148, p. 103807, 2020. DOI: 10.1016/j.compedu.2020.103807.

[7] S. Papadakis, "Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age," in *Mobile Learning Applications in Early Childhood Education*, IGI Global, 2020, pp. 101-121. DOI: 10.4018/978-1-7998-1486-3.ch006.

[8] J. M. Sáez-López, M. Román-González, and E. Vázquez-Cano, "Visual Programming Languages Integrated Across the Curriculum in Elementary School: A Two-Year Case Study Using Scratch in Five Schools," *Comput. Educ.*, vol. 97, pp. 129-141, 2016. DOI: 10.1016/j.compedu.2016.03.003.

[9] A. Wilson, T. Hainey, and T. Connolly, "Evaluation of Computer Games Developed by Primary School Children to Gauge Understanding of Programming Concepts," in *Proc. Eur. Conf. Games-Based Learn.*, Academic Conferences International Limited, 2012, p. 549.

[10] F. Heintz, L. Mannila, and T. Farnqvist, "A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education," in *2016 IEEE Frontiers in Education Conf. (FIE)*, 2016, pp. 1-9. DOI: 10.1109/FIE.2016.7757411.

[11] P. Seow et al., "Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore," in *Computational Thinking Education*, Springer, Singapore, 2019, pp. 345-361. DOI: 10.1007/978-981-13-6528-7_18.

[12] V. Lin and O. Shaer, "Beyond the Lab: Using Technology Toys to Engage South African Youth in Computational Thinking," in *Proc. 2016 CHI Conf. Extended Abstracts Human Factors Comput. Syst.*, 2016, pp. 655-661. DOI: 10.1145/2851581.2851634.

[13] Codeweek site, https://codeweek.eu/ Accessed: 2022-12-21.

[14] S. Bocconi et al., *Reviewing Computational Thinking in Compulsory Education*. Joint Research Centre (Seville site), 2022.

[15] Code for Asia Site, https://www.codefor.asia/ Accessed: 2023-02-07.

[16] Coding Africa Site, https://www.codingafrica.org/ Accessed: 2023-02-07.

[17] A. Garcìa-Holgado et al., "Trends in Studies Developed in Europe Focused on the Gender Gap in STEM," in *Proc. XX Int. Conf. Human Comput. Interact.*, 2019, pp. 1-8. DOI: 10.1145/3335592.3335607.

[18] A. Garcìa-Holgado et al., "Gender Equality in STEM Programs: A Proposal to Analyse the Situation of a University About the Gender Gap," in *2020 IEEE Global Eng. Educ. Conf. (EDUCON)*, 2020, pp. 1824-1830. DOI: 10.1109/EDUCON45650.2020.9125334.

[19] Girls Who Code Site, https://girlswhocode.com/ Accessed: 2022-12-21.

[20] Coding Girls Site, https://www.coding-girls.com/ Accessed: 2022-12-21.

[21] E. Benetti and G. Mazzini, "Coding Training Proposal for Kindergarten," in *2020 Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, 2020, pp. 1-5. DOI: 10.23919/SOFTCOM50211.2020.9238265.

[22] L. C. Klopfestein et al., "CodyColor: Design of a Massively Multiplayer Online Game to Develop Computational Thinking Skills," in *Extended Abstracts Annu. Symp. Comput.-Human Interact. Play Companion Extended Abstracts*, 2019, pp. 453-458. DOI: 10.1145/3341215.3356303.

[23] A. Bogliolo, "Unplugged Language-Neutral Card Games as an Inclusive Instrument to Develop Computational Thinking Skills," in *Proc. 9th Int. Technol., Educ. Dev. Conf. (INTED)*, 2015, pp. 7609-7615.

[24] mBlock Site, https://mblock.makeblock.com/en-us/ Accessed: 2022-12-21.

[25] E. Benetti and G. Mazzini, "Coding Training Proposal from Kindergarten to High School," in *2021 Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, 2021, pp. 1-5. DOI: 10.23919/SOFTCOM52868.2021.9559112.

ELISA BENETTI was born in Portomaggiore (FE) in 1981. In October 2006 she obtained a bachelor's degree in Computer and Automation Engineering from the University of Ferrara. In October 2008 she obtains her bachelor's degree at the same university with a thesis entitled "Efficient Systems with Authentication for Digital TV." The study of MHP services led her, from November 2008, to a one-year collaboration contract at Lepida SpA, within the LepidaTV project where she deepened her studies in the field of Digital Terrestrial and Audio/Video. In January 2009, he began his PhD in Engineering Sciences at the University of Ferrara, receiving his degree in January 2012 with a thesis entitled "Architectures, Services and Multimedia in New Media." From 2009 to the present, she has authored and coauthored some 20 publications in the areas of telecommunications, DataCenter infrastructure and coding dissemination in public schools. In January 2010 she is hired in LepidaSpA in the Ideation and Prototypes Area. Since 2014 she also started working in the Design and Development - DataCenter and Cloud area where to date she works full-time as a systems engineer.

GIANLUCA MAZZINI was born in 1968. He received his Master Degree and PhD from the University of Bologna in 1992 and 1996, respectively. Since 1996 he had a permanent position at the University of Ferrara. Since 1995 he taught courses on Telecommunications. He served as a tutor for 150 theses and supervised 15 PhD students. He published more than 280 international and peer-reviewed papers. He served has IEEE Associate Editor for 9 years. He managed 40 research projects, with the role of coordinator for 20. He served in 60 Technical Program Committee. He is an IEEE Fellow. He was member in the Scientific Council of: CNIT, GARR, Cento Technopole, Telematic Strategy for the Emilia-Romagna Region, ASTER, Iperbole Wireless for the Municipality of Bologna, Guglielmo Marconi Foundation. He served as R&D Director in LepidaSpA, as CEO in Cup2000ScpA, as member of the Board of Directors in LepidaSpA and as member ofthe Board of Directors in CNIT. Since 2009 he is CEO of LepidaSpA, reformed as LepidaScpA in 2019. Since 2019 he is a member of the Board of Directors of the Guglielmo Marconi Foundation.