UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

pARt bloki: Programiranje v dopolnjeni resničnosti z materialnimi bloki

(pARt blocks: Programming in AR with tangible blocks)

Ime in priimek: Karolina Trajkovska
Študijski program: Računalništvo in informatika
Mentor: izr. prof. dr. Klen Čopič Pucihar
Somentor: izr. prof. dr. Matjaž Kljun
Delovni somentor: asist. Maheshya Weerasinghe, asist. Jordan Aiko Deja

Koper, september 2022

Ključna dokumentacijska informacija

Ime in PREIMEK: Karolina TRAJKOVSKA

Naslov zaključne naloge: pARt bloki: Programiranje v dopolnjeni resničnosti z materialnimi bloki

Kraj: Koper

Leto: 2022

Število listov: 56	Število slik: 12	Število tabel: 2
Število prilog: 7	Število strani prilog: 13	Število referenc: 32

Mentor: izr. prof. dr. Klen Čopič Pucihar

Somentor: izr. prof. dr. Matjaž Kljun

Delovni somentor: asist. Maheshya Weerasinghe, asist. Jordan Aiko Deja

Ključne besede: učenje, mešana resničnost, otipljivi vmesniki, programska okolja

Izvleček:
Programiranje je kompleksen izziv za začetnike, saj zahteva razumevanje sintakse in obvladovanje logičnega razmišljanja. Mlajši začetniki, ki se bolj odzivajo na vizualne in oprijemljive metode učenja, pogosto naletijo na težave, ko se srečajo z običajnimi tekstovnimi programskimi vmesniki. Zarade tega je bilo razvitih nekaj visikonivojskih blokovnih probramskih jezikov, ki omogočajo začetnikom programirati z zlaganjem blokov. V tem diplomskem delu predstavljamo pARt Blocks, programsko okolje, ki omogoča učencem programirati z oprijemljivimi fizičnimi bloki izdelanimi iz lesa s katerimi kontrolirajo digitalno okolje prikazano v razširjeni resničnosti. S ciljem razumeti, ali uporaba razširjene resničnosti prispeva k izboljšanju učnih rezultatov, smo izvedli uporabniško študijo. Naše ugotovitve lahko služijo kot smernice za prihodnje razvijalce pri izboljšavi učnih sistemov za osvajanje kompleksnih veščin, kot je programiranje.

Key words documentation

Name and SURNAME: Karolina TRAJKOVSKA

Title of final project paper: pARt blocks: Programming in AR with tangible blocks

Place: Koper

Year: 2022

Number of pages: 56 Number of figures: 12 Number of tables: 2

Number of appendices: 7 Number of appendix pages: 13 Number of references: 32

Mentor: Assoc. Prof. Klen Čopič Pucihar, PhD

Co-mentor: Assoc. Prof. Matjaž Kljun, PhD

Working co-mentor: Assist. Maheshya Weerasinghe, Assist. Jordan Aiko Deja

Keywords: learning, mixed reality, tangible interfaces, programming environments

Abstract:
Programming is a complex challenge for beginners, as it requires an understanding of syntax and mastery of logical thinking. Younger learners, who are more responsive to visual and tangible learning methods, often encounter difficulties when faced with conventional text-based programming interfaces. Because of this, several high-level block programming languages have been developed, enabling beginners to program by stacking blocks. In this thesis, we introduce pARt Blocks, a programming environment that allows students to program using tangible physical blocks made of wood, through which they control a digital environment displayed in augmented reality. In order to understand whether the use of augmented reality contributes to improved learning outcomes, we conducted a user study. Our findings can serve as guidelines for future developers in enhancing educational systems for acquiring complex skills such as programming.

Acknowledgement

I would want to thank the HICUP Lab for allowing me to bring my vision to life, for their assistance, and for allowing me to access the lab's equipment. Furthermore, I would like to express my sincere gratitude for the help received from my mentor, co-mentor and working co-mentors. They have been of tremendous assistance to me in the process of figuring out how to do appropriate research. They provided me with a comprehensive foundation to the research world, including advice on how to approach the topics I want to examine and how to conduct the research. In addition to that, they assisted me in resolving some bugs that were present in the implementation and recruiting participants for the user study. I had the privilege of gaining knowledge from the very best.

I would also want to thank my family and Milan Milivojčević for their unwavering support during my studies. It had a profound impact on me.

Contents

1 Introduction 1
1.1 Problem Statement . . . 1
1.2 Research Questions and Objectives . . . 2
1.3 Contributions . . . 2
1.4 Thesis Overview . . . 3

2 Theoretical Background 4
2.1 Block-based Learning . . . 4
2.2 Tangible Interfaces and Learning . . . 6
2.3 Learning in Augmented Reality . . . 7

3 pARt Blocks Prototype 9
3.1 Concept Design . . . 9
3.2 Desktop (non-AR) Implementation . . . 11
3.3 HoloLens (AR) Implementation . . . 12

4 Research Method 15
4.1 Study Conditions and Design . . . 15
4.2 Participants . . . 15
4.3 Protocol . . . 15
4.4 Data Collection . . . 16

5 Results 18
5.1 Performance, Mental Effort and Time . . . 18
5.2 Learning Efficiency . . . 19
5.3 System Usability and User Experience . . . 19

6 Discussion 21
6.1 Limitations . . . 21
6.2 Design Recommendations . . . 22

7 Conclusion and Future Work 23

8 Povzetek naloge v slovenskem jeziku 25

9 Bibliography 28

List of Tables

1 Descriptive statistics for non-AR and AR condition for all dependant variables. 18
2 Learning efficiency for each participant in the non-AR and AR condition. 19

List of Figures

1 Scratch platform design [25] 5
2 Tim the Train with tangible parts [8] 6
3 Execution of the The AR Flowchart System [2] 8
4 Prototype design (a) Augmented reality (AR) interface with tangible interaction; (b) Desktop interface (non-AR) with drag and drop interaction. 9
5 Scenario design (a) Preview of the game field; (b) The corresponding blocks code. 11
6 Desktop (non-AR) prototype: Task solved 12
7 Design of AR Prototype 13
8 Preliminary implementation using paper cards in the shapes of the blocks 13
9 Later implementation using laser-cut tangible blocks 14
10 HoloLens (AR) prototype: Task solved 14
11 Desktop condition (a) Application testing; (b) NASA-TLX questionnaire. 16
12 User experience questionnaire: (a) Mean values per item/question; (b) Mean values of UEQ factors (pragmatic and hedonic) and overall. . . . 20
13 A rough sketch of the future work ideas 24

Appendices

A The Storyboard of AR Interface
B The Storyboard of non-AR (Desktop or Tablet) Interface
C Scenes from the Desktop Prototype
D Step-by-step AR Prototype
E NASA-TLX Questionnaire
F UEQ and SUS Questionnaires
G User Information Questionnaire

List of Abbreviations

i.e.	that is
e.g.	for example
PL	Programming Language
IDE	Integrated Development Environment
HMD	Head Mounted Display
HCI	Human Computer Interaction
AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
XR	Extended Reality
VPL	Visual Programming Languages
MIT	Massachusetts Institute of Technology
NASA-TLX	NASA Task Load Index
SUS	System Usability Scale
UEQ	User Experience Questionnaire

1 Introduction

1.1 Problem Statement

Conventional programming language (PL) integrated development environments (IDE) are designed in a text-based format consisting of warnings and logs, developer consoles and similar panels. This format allows more experienced developers to transition between platforms and frameworks seamlessly. However, for a novice programmer, many of these activities can be easily understood and learned through improved visuals [18]. Thus, researchers have explored Visual Programming Languages (VPL), in which visual formalisms are used rather than texts [3]. For example, 'Scratch', one of the most known VPLs designed by the MIT media lab in 2017 [19] has been used and has inspired several code blocks-based approaches to programming [13, 21].

Recent research in Human-Computer Interaction (HCI) has been exploring newer interaction methods beyond visuals, such as "tangible interfaces" to create platforms to easily-learn programming languages targeting novices, especially younger children [6, 15, 16]. Most of these interfaces are designed with electronically-tagged blocks [15], while some of them focus on AR marker-based cards [16] or cubes [6]. Their findings of these works contributed to improved enjoyment, higher motivation and better engagement. Yet, there are almost-to-none contributions observing programming comparing tangible interfaces to visual programming with drag and drop interactions that consider the novices' mental effort or learning outcomes. Further, to the best of our knowledge, no existing application combines tangible interaction with AR that runs on a head-mounted display (HMD), which supports the learning process by introducing a more immersive and engaging active learning experience [28].

To bridge this gap, we introduce *pARt Blocks*, an innovative (AR) application designed for learning (p)rogramming, by combining visual (AR) and (t)angible interactions, offering users a more immersive and engaging educational experience. In *pARt Blocks*, the '*AR*' presents an active and playful interface for programming, integrating AR elements like a 3D animated game field, a virtual avatar, and a gamified points collection system, as illustrated in Figure 4 - a. We posit that presenting these AR elements through a Head-Mounted Display (HMD) can enhance engagement, making the pursuit of successful programming scenarios even more enjoyable. This aligns with the Montessori method, which emphasises that physical interaction with objects strongly enhances active learning [1, 14]. Thus, we designed our prototype to incorporate marker-based tangible puzzle pieces. This tangible aspect–represented by the '*t*' in *pARt Blocks*–aims to aid novices in recalling syntax and mastering logic in an immersive and efficient manner. The tangible component replaces conventional mouse clicks with tactile interactions, offering a promising avenue for enhancing the learning journey of novice programmers.

To validate our proposed approach, we conducted a between-subject study comparing two representation methods (interfaces): tangible-AR and non-AR drag-and-drop interaction (as illustrated in Figure 4 1 - b). Our findings shed light on design implications and provide recommendations that can assist teachers and instructional designers in effectively teaching complex and abstract programming concepts, including constructs, loops, and others.

1.2 Research Questions and Objectives

With the proposed study, we will try to answer the following research questions:

1. Will the AR-based tangible prototype for block-based programming have an impact on the learning outcomes?

2. Will the AR-based tangible prototype for block-based programming have an impact on the level of engagement?

This study aims to assess whether the introduction of an AR tangible interface provides improved outcomes compared to traditional Visual Programming Languages (VPLs) experienced on a 2D display. The investigation builds upon prior research that has evaluated and compared VPLs to conventional programming methods in the context of teaching fundamental programming concepts to young learners. Our application is designed to make programming more captivating for children, not only fostering improved learning capabilities but also motivating independent practice. Our objective is to shift the perception that programming is less engaging than other activities, like video games, commonly favored by young individuals.

1.3 Contributions

In summary and based on the gaps in existing literature that we have found, this thesis aims to provide the following contributions:

- A desktop-based prototype that allows players to solve programming tasks using visual programming blocks
- An HMD-based prototype that provides players a AR experience with tangible wooden blocks to solve programming task
- Results of an empirical study that involves student participants solving programming tasks using our desktop-based and HMD-based prototypes
- Design recommendations based on the results of our user studies

1.4 Thesis Overview

This thesis is structured into several chapter. The upcoming chapter delves into the theoretical foundation to provide the reader with a comprehensive understanding of our approach. Subsequently, we will detail the design of our application the method employed in the experiment. Next, we will present the results and engage in a discussion to elucidate how these findings contribute to answering our research question.

2 Theoretical Background

2.1 Block-based Learning

In this chapter, we will discuss why we chose block-based programming as the foundation of our research. As indicated in the introduction, we will conduct a comparative analysis between a 2D block-based programming interface and an enhanced AR version with tactile interaction. Importantly, both iterations support block-based programming by design, and this alignment is not coincidental. The initial design drew inspiration from Scratch, a widely recognized platform for young developers. Serving as the largest coding community for children globally, Scratch offers a user-friendly visual interface that empowers youngsters to craft digital stories, games, and animations. Motivated by the goal of bringing programming closer to children, the MIT team behind Scratch aimed to overcome challenges posed by prior efforts, such as interfaces that were difficult to navigate or concepts that failed to captivate the interest of young learners. Furthermore, previous prototypes lacked adequate guidance, rendering the learning process challenging [25].

Moreover, one of the notable advancements in block-based programming, as highlighted by Resnick, is the alignment of the visual appearance of blocks with their syntactic meaning [25]. This involves organizing the blocks in a way that visually indicates their compatibility, facilitating the proper assembly. This alignment serves as the fundamental principle of block-based programming, and we have incorporated it into pARt Blocks. The application encompasses five types of blocks, each addressing essential programming concepts required to accomplish a straightforward task, as elaborated further in the Application Design chapter. Notably, two categories include Events and Variables blocks, deliberately designed to be incompatible, signaling to children that a Control or Operator block should be employed in between.

Radek outlines additional benefits associated with the use of a block-based programming approach [23]. Notably, by minimizing the possibility of syntactical errors from the outset, they emphasize the concept of "chunking", providing youngsters with an expert-level perspective that can be comprehended without prior knowledge. Moreover, this approach significantly reduces cognitive burden, as children are not required to memorise all instructions; instead, they can select from a variety of blocks cate-

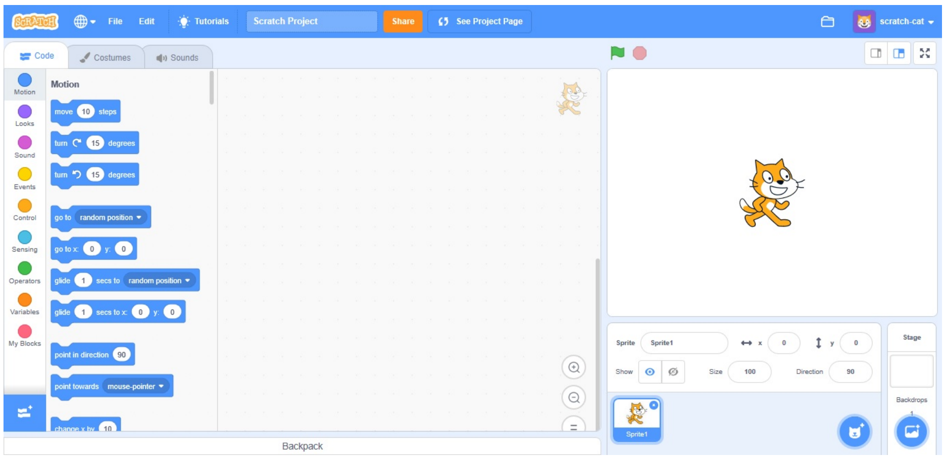


Figure 1: Scratch platform design [25]

gorised by their functionality. In pARt Blocks, we carefully curated essential blocks and included a few additional ones to present users with multiple options. For instance, even though the prototype primarily employs boolean types, it introduces users to other variable types, such as integers and strings, fostering an understanding of diverse data types.

Moreover, the adoption of block-based programming, particularly through the Scratch platform, offers significant advantages. For example, Abdularhman aimed to enhance children's participation in programming competitions [12]. In comparing technological fields, his study revealed that robotics tournaments attracted more students than algorithm-solving competitions. Similar to Resnick, he shared concerns that the current programming presentation did not captivate children enough; in their minds, constructing robots was more enjoyable. Thus, the tangible input aspect of physical blocks could be bring additional benefits. Resnick also highlighted the success of the Scratch project's collaboration with Lego, increasing children's enthusiasm for building things and stimulating their creative and computational thinking [25]. Another issue Abdularhman identified was that children were not introduced to programming at a young age. Before progressing to more advanced programming, he taught children how to use Scratch in preparation for the Syrian Olympiad in Informatics. Over 90% of the children enjoyed the preparations, and the competition took on a fresh perspective, emphasising the development of students as creative thinkers with skills applicable beyond programming. Following the competition, numerous schools and even colleges expressed interest in providing similar training or seminars for their students.

2.2 Tangible Interfaces and Learning

Tangibility aims to engage more of children's senses to enhance engagement. Beyond learning, numerous activities can become more captivating by incorporating additional senses. This aspect aligns with the focus of Human-Computer Interaction (HCI) academics and is supported by a designer who proposed the "Five Senses Theory" [17]. According to this theory, utilising various senses contributes to improved design, enhancing interactivity and enjoyment for consumers.

Several prototypes using tangible user interfaces have been explored from the educational aspect. Tim the Train [8] provides an intriguing example of incorporating sensory elements, particularly tangibility, designed for children aged 5 and above. The study found that involving various senses in learning is advantageous for children, as they benefit from tactile, visual, auditory, and olfactory stimuli. With the prototype the children can grasp concepts such as commands, alternative commands, loops, and abstraction by manipulating tetris-like pieces to load and unload train carriages. The children received chunks of instructions that create a task (an algorithm), that they must complete by adding and removing these tetris-like pieces.

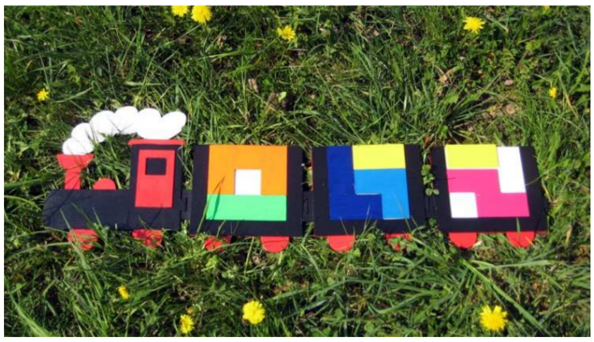


Figure 2: Tim the Train with tangible parts [8]

[32] provides further insights into tangibility, highlighting that children exhibit a strong affinity for sensory development tools and engage with physical materials with heightened attention for lengthy periods f time, leading to advancements in critical thinking. Learning through hands-on experiences, or interacting with the environment, a concept officially known as experiential learning [14, 31], proves to be more effective than traditional methods such as reading books or listening to lectures. Direct physical interaction with the world plays a crucial role in the cognitive development of children [1]. Therefore, we aim to incorporate tactile interaction in our pARt Blocks to enhance learning outcomes and foster creativity. As mentioned in the introduction, while there is existing research on tangibles for programming and AR for programming, we are not aware of any that combines both to teach programming with the added dimension of

AR through an HMD.

In tasks such as programming, maintaining children's attention is crucial. Moreover, the prototype should be captivating enough to encourage students to continue practicing. In this context, programming can be considered a skilled behavior, implying that human performance improves with experience [7]. Ideally, children should have multiple tasks and the ability to practice independently. Currently, we are assessing improvement after just one iteration of exercise. At present, our tangible blocks are constructed from a basic material, but in the future, we aim to attach 3D printed figurines to them. For the movement blocks, we would link the object in the scenario to which the movement is applied. For the control blocks, we may attach something indicating their purpose, such as adding a question mark to the 'if' block. With the AR visualisation, we anticipate that the children's enthusiasm for engaging with the app will further increase. More details on this can be found in the next subsection.

2.3 Learning in Augmented Reality

Augmented Reality offers a diverse range of advantages leveraged by numerous applications today. While in Virtual Reality, the user is completely immersed in a virtual environment, in the term Augmented reality, in the business context also referred to as Mixed Reality, pertains to an environment where real-world and virtual elements interact or coexist in users' view. This means the real world is "augmented" by additional virtual elements. Essentially, users in AR remain aware of their surroundings while interacting with virtual objects.

We anticipate that transitioning from the tablet to AR version will enhance children's engagement. The AR Flowchart System, as presented in [2], supports our expectation. The scenario involves visualising a package delivery to a residence, where the learner designs a flowchart using cards, and they observe the execution field via a tablet. The study found unanimous agreement among students that utilising this AR flowchart enhances their understanding of control structures, and they expressed satisfaction with the experience. A similar approach of visualizing programming with cards through a tablet is also explored in [16].

Another study supporting the notion that AR enhances enjoyment is the Augmented-Reality Scratch [24]. By blurring the lines between reality and fiction, this approach enables the creation of magical experiences, such as stories where otherworldly beings inhabit the space directly in front of the reader or environments where music is influenced by physical interactions. Augmented-Reality Scratch allows children to use the tablet's or computer's camera to move markers with attached sprites while programming different events (e.g., when sprites meet). The enjoyment factor of the scenario in our pARt Blocks could be significantly influenced, given that the setup allows for ample room for animation in augmented reality. The scenario involves a dog shelter that can be realistically and vividly depicted in AR, providing an engaging environment for children to interact with virtual puppies.

AR is also used to facilitate experiential learning in various educational applications beyond programming. An exemplary case is the VocabulARy [29], which focuses on acquisition of vocabulary in a foreign language. The study compares the effectiveness of a 2D tablet interface with a 3D AR interface, revealing that the AR system outperforms the tablet system in terms of immediate recall, reduced mental effort, and quicker job completion time. This research design closely aligns with the approach taken in pARt Blocks, and further details can be explored in subsequent chapters. Our emphasis on comparing the two interfaces aims to illustrate findings akin to those demonstrated by the VocabulARy study.

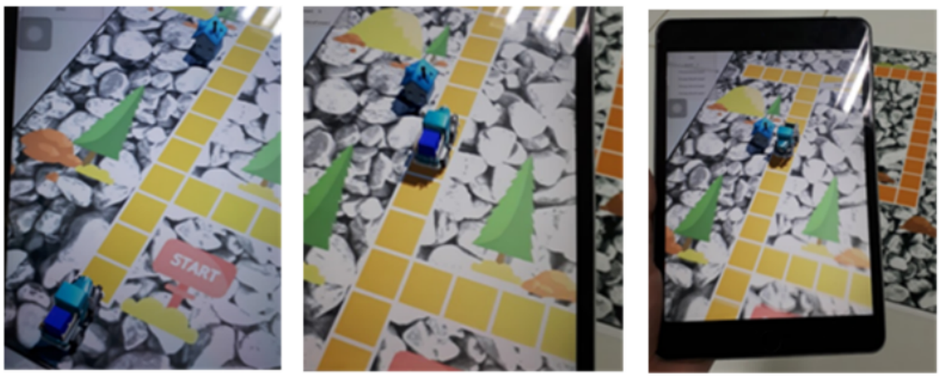


Figure 3: Execution of the The AR Flowchart System [2]

3 pARt Blocks Prototype

In this chapter we describe major considerations regarding scenario creation, implementation, and technologies being used. The identical concept has been implemented twice for two different interface configurations. One for tablet or desktop use (non-AR), and the other for Microsoft HoloLens 2 (AR). Because of the same scenario used in each condition we decided to conduct a between-subject study design, which is explained in detail in section 4.1.

3.1 Concept Design

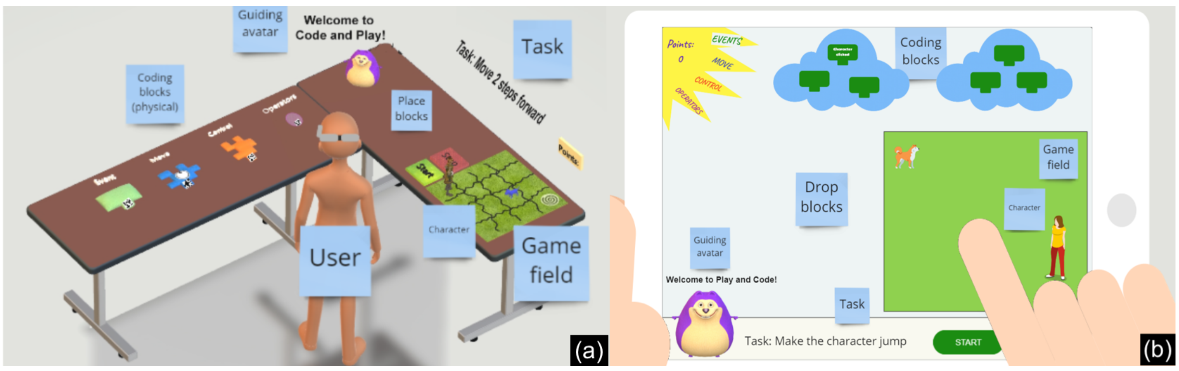


Figure 4: Prototype design (a) Augmented reality (AR) interface with tangible interaction; (b) Desktop interface (non-AR) with drag and drop interaction.

CodeCubes [6] utilized instructions and sequences of instructions, while Tim the Train [8] incorporated also if/else constructs and loops through abstracted cards. In our investigation, we aimed to explore the impact of introducing additional events and less abstracted blocks, providing a clearer representation of programming language concepts. Unlike using abstracted images, we opted for presenting the key block names using words. Essentially, our goal was to take a step further and introduce upper elementary or early high school students to authentic programming principles without directly immersing them in a specific programming language (PL). Moreover, the blocks deliberately avoid representing a particular PL syntax; instead, they feature universal names, enabling students to seamlessly transition to any programming language once they have mastered block-based programming.

The core idea was to create a programming area where blocks could be interconnected, a game field to showcase the execution, and an initial placement for the blocks. Additionally, in terms of interface design, our goal was to establish a child-friendly environment. Figure 4 illustrates these components, and detailed information about each will be provided in the respective subsections.

The goal for the scenario was to create something endearing, avoiding any suggestion of violence and instead promoting goodwill. This is why we chose to feature dogs and a character (in our case, Anna) whose primary objective is to befriend and contribute to the happiness of the dogs. The task would explicitly outline what actions the character should perform, and the programming area would control Anna's movements. In other words, everything in the programming space would be applied to what Anna is expected to achieve. The animation of the dogs is incorporated to showcase satisfaction and enhance the child's motivation to succeed in the subsequent rounds, even though this specific behavior of the dogs is not part of the task programmed by the children.

An example task could involve initiating the execution, where Anna is required to visit each dog in the shelter. If the dog is hungry or cold, Anna should feed the dog and provide a blanket. If the dog is only cold, then Anna should offer a blanket. The task involves understanding and implementing the following concepts:

- On start (event) - when execution begins
- Repeating action (loop) - visit each dog
- Conditional assignments (if / else if /else construct, logical operators, variables)

In light of this, we came up with the following building blocks:

1. Event blocks: specify when something should occur.
2. Control blocks: create loops and if/else expressions.
3. Operator blocks: specify the logical and comparison operators.
4. Variables blocks: preset variables with no input option to reduce confusion for users. However, several types of variables have been defined. The prototype uses boolean variables.
5. Move blocks: define the movements of the character (Anna)

When discussing the design, it is crucial to highlight the incorporation of gamification elements. Weerasinghe provides the distinction between game-based learning and gamification [30]. To put it simply, game-based learning involves the creation of

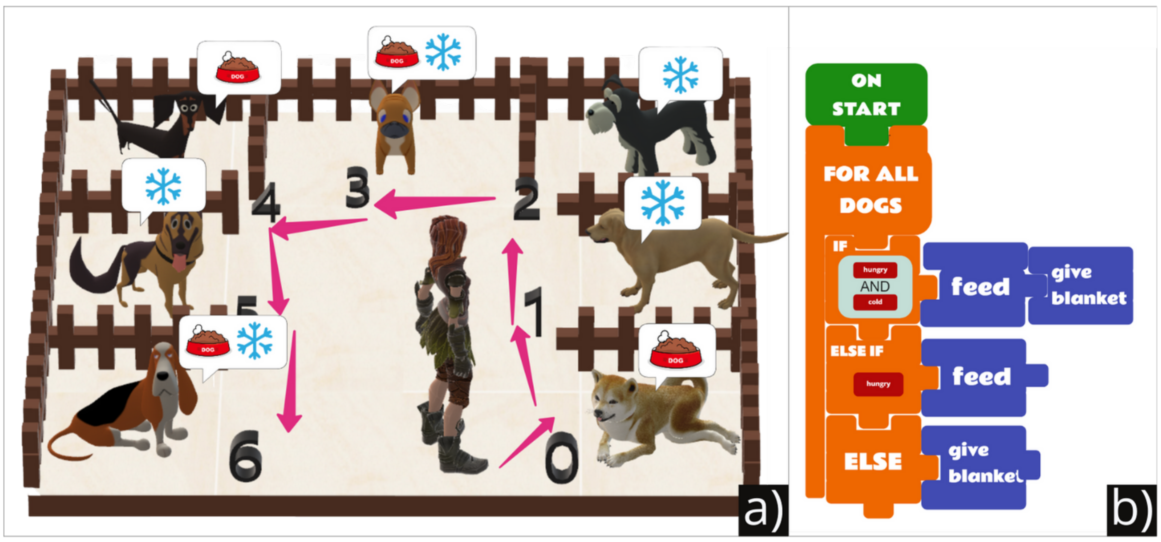


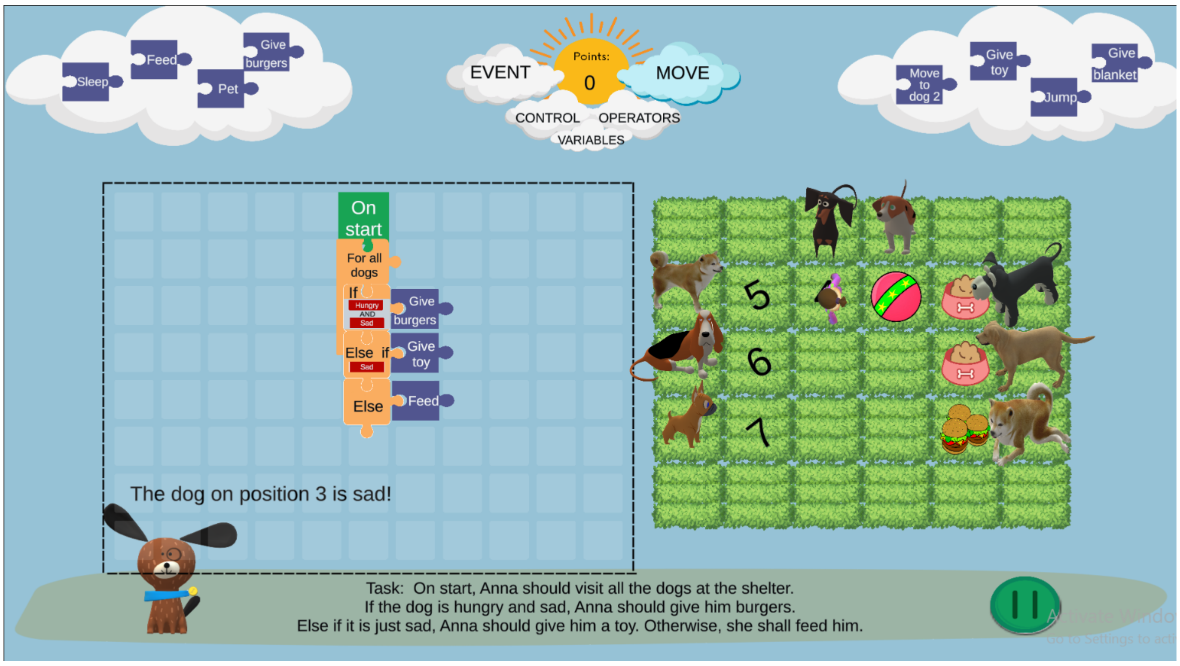
Figure 5: Scenario design (a) Preview of the game field; (b) The corresponding blocks code.

educational activities with game features, essentially crafting a game for educational purposes. Conversely, gamification involves integrating specific game ideas and components into our learning environment to enhance user engagement. The latter approach elucidates our choice to incorporate gamification. We believe that by introducing features such as a guiding avatar and awarding points for each task, we can heighten engagement and make the application more appealing to children. The outlined features are evident in both prototypes, as illustrated in Figure 4.

3.2 Desktop (non-AR) Implementation

The 2D prototype was initially intended for a tablet computer; however, during testing, it became apparent that the blocks appeared too small for tablet use. As a result, the testing was conducted on a desktop computer. Nevertheless, for future development, this aspect needs refinement, as elaborated further in the subsequent sections. The desktop version took precedence in implementation, introducing the logic for the first time. Developed in Unity, all elements are situated on a canvas, characterised by a distinctive design in terms of placement and functionality. Canva Pro facilitated the creation of several sprites.

The arrangement of objects predominantly adhered to the original concept. The sun is encircled by small clouds housing buttons for altering the types of blocks. Figure 6 exemplifies one instance of this functionality: when a button is pressed, the corresponding blocks emerge, floating in from both sides. The programming area, also



The dog on position 3 is sad!

Task: On start, Anna should visit all the dogs at the shelter.
If the dog is hungry and sad, Anna should give him burgers.
Else if it is just sad, Anna should give him a toy. Otherwise, she shall feed him.

Figure 6: Desktop (non-AR) prototype: Task solved

visible in Figure 6, where the blocks are dragged and dropped, is displayed on the right. Items in the programming field help to recognise the sequence of the blocks inserted. The blocks also have a simulated magnet effect, so one block cannot be in between and still click properly with the rest of the blocks.

At the bottom is a bar featuring the avatar, the task description, and a play button. Upon pressing the play button, the execution starts if the program is correctly programmed by the user by placing the blocks in the correct way. The dogs, randomly generated with states of hunger, cold, or both, are then visited by Anna. The numerical indicators in front of the dogs, resembling an array, transform into food, toys, or burgers based on the assigned task. The execution unfolds on the right side of the screen and can be paused at any moment. Upon completion, 100 points are awarded.

Moreover, to enhance the user testing presentation and ensure a smoother overall flow, three additional scenes were incorporated, as depicted in Appendix D.

3.3 HoloLens (AR) Implementation

The AR prototype follows a similar organisation, but with the utilisation of physical blocks. To recognise them and understand how they are connected (positioned), we used the Vuforia Engine in conjunction with Unity3D. The decision to use an HMD over a handheld device allows users to move freely around and manipulate the physical blocks. The types of blocks are also the same as in desktop prototype. Another crucial

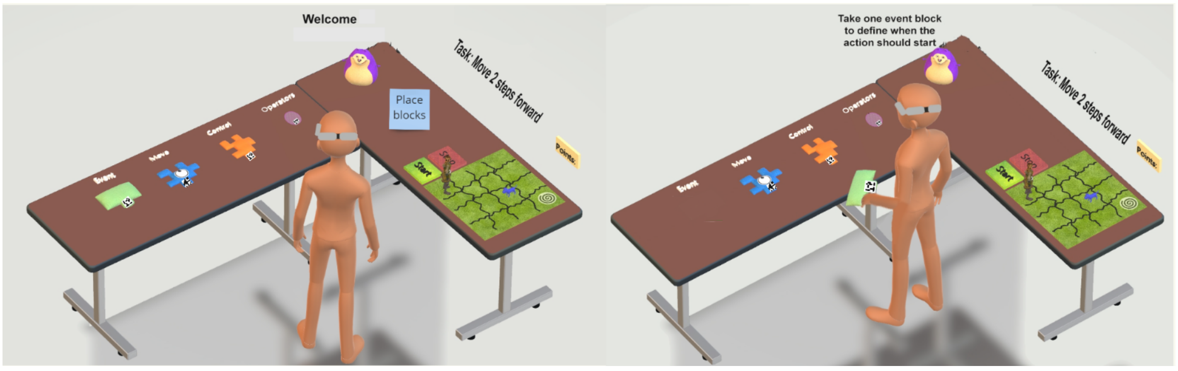

Welcome
Place blocks
Task: Move 2 steps forward
Event
Move
Control
Operators
Start
Points
Take one event block to define when the action should start

Figure 7: Design of AR Prototype

aspect is the introduction of finely animated dogs that can evoke emotions and capture the user's interest (Figure 10), illustrated as important in the Theoretical background chapter.

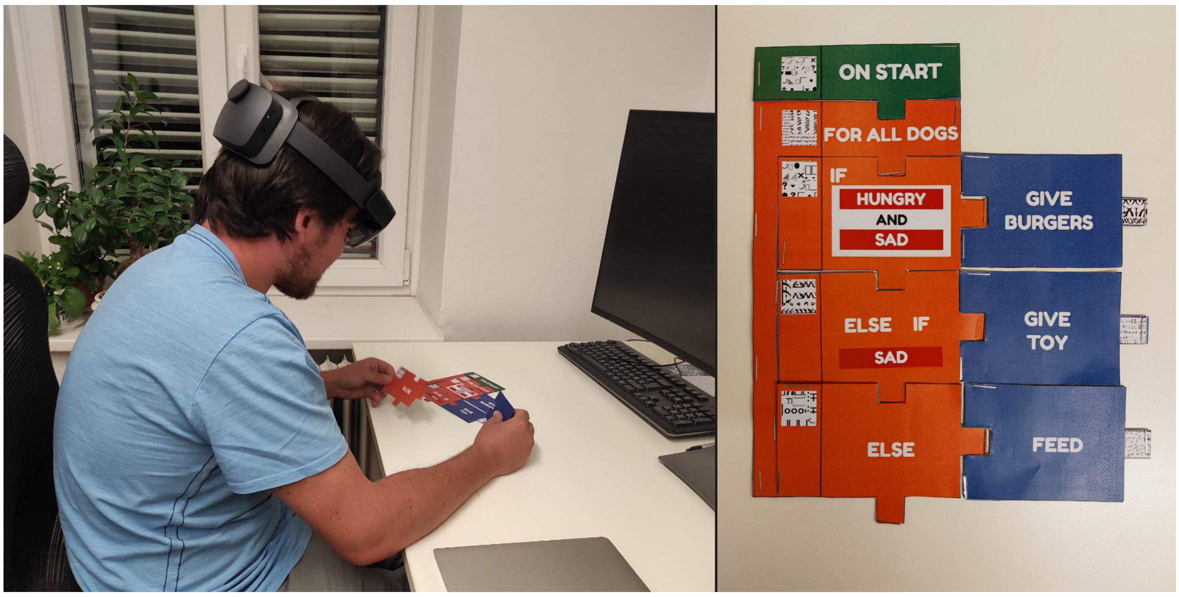


Figure 8: Preliminary implementation using paper cards in the shapes of the blocks

For the implementation of the logic and the game field, we initially used temporary paper cards with the shapes of the blocks (Figure 8). Once the code was functioning correctly and the game field was animated accordingly, we replaced the paper cards with physical blocks that were laser-cut from wood. They were much easier to assemble. The wooden blocks have paper glued on top, based on which they can be recognized both by the user and the Vuforia Engine. The Computer Vision system of the Vuforia Engine performed well in detecting the blocks, even without the use of AR marker detection, since the blocks differed in shape, colour, and text written on them. As in the desktop implementation, upon pressing the play button, the execution starts if the program is correctly programmed by the user by placing the blocks in the correct way. Otherwise, the character Anna reports an unsuccessful attempt.

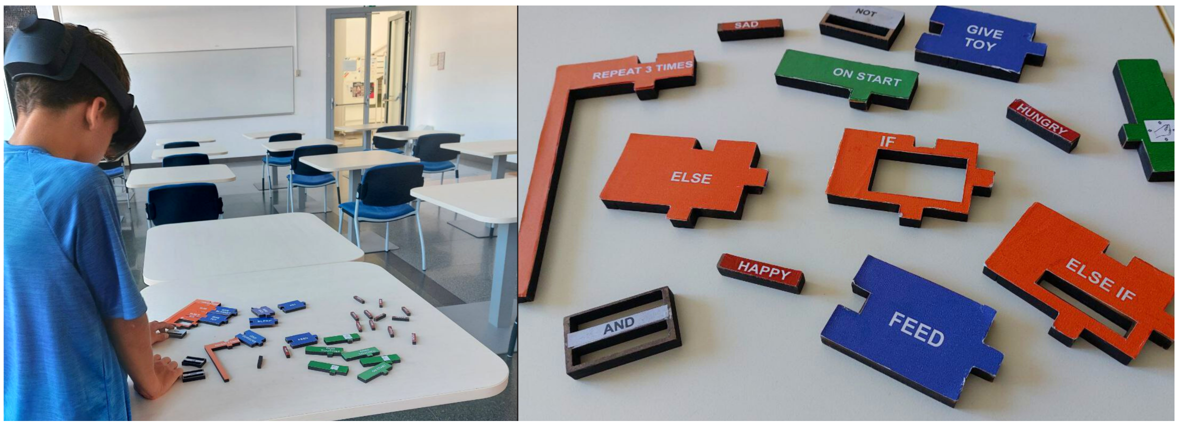


Figure 9: Later implementation using laser-cut tangible blocks

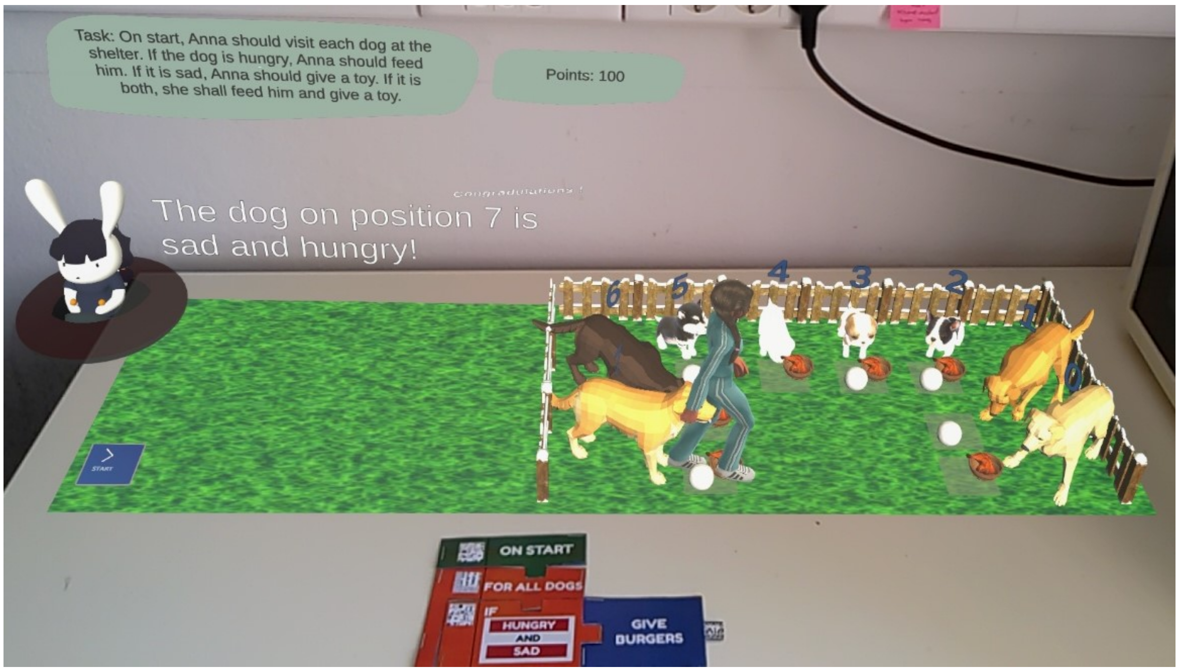

Task: On start, Anna should visit each dog at the shelter. If the dog is hungry, Anna should feed him. If it is sad, Anna should give a toy. If it is both, she shall feed him and give a toy.
Points: 100
The dog on position 7 is sad and hungry!
ON START
FOR ALL DOGS
IF
HUNGRY
AND
SAD
GIVE BURGERS

Figure 10: HoloLens (AR) prototype: Task solved

4 Research Method

This chapter explains the study conditions, design, participant recruitment, study method, data collection, and analysis.

4.1 Study Conditions and Design

We had two conditions, each using a distinct interface of the programming environment: (i) a 2D desktop interface featuring drag-and-drop interaction with blocks—non-AR condition, and (ii) a 3D AR interface incorporating tangible puzzle blocks—AR condition. Both conditions are illustrated in Figure 4. The study a between-subject design.

4.2 Participants

For the non-AR condition eleven (11) participants were voluntarily recruited from the Slovenian scout event that took place in Cerkno, Slovenia (from 29 July to 7 August, 2023). The sample included two (2) female participants (18.2%). For the AR condition four (4) participants were voluntarily recruited in Koper (all male). None of the participants had any programming experience beforehand. The participants were aged between 10 and 24 years, with the mean of $\overline{x} = 16.3$ and $SD = 3.41$.

4.3 Protocol

Upon assignment to their respective conditions, participants were provided with a consent form to sign. For those below the age of 18, the form was forwarded to their parents for permission. Before proceeding with the task and scenario, participants were instructed to watch a 2-minute tutorial video. The tutorial included a brief overview of programming, an introduction to object positioning, and a demonstration of a specific task. Various types of blocks were also highlighted. Following the completion of the first scenario, students filled out the NASA Task Load Index (NASA-TLX) questionnaire [10]. Subsequently, they completed a post-test questionnaire aimed at assessing their immediate recall skills.

The post-test involved an assignment with the same programming principle but in a new context, requiring participants to write pseudocode, with the definition of pseudocode provided. Following the completion of the experiment, participants were given two more standard questionnaires: a System Usability Questionnaire (SUS) and a User Experience Questionnaire (UEQ). Another short post-questionnaire was given to participants at the conclusion of the study, which asked about their demographics, their prior experience with AR technology, and any visual issues they were aware of. The entire experiment took from 30 to 45 minutes.

Figure 11: Desktop condition (a) Application testing; (b) NASA-TLX questionnaire.

4.4 Data Collection

In order to measure the task completion time, the time stamp data (start time and end time) were logged by the system. The NASA Task Load Index (NASA-TLX) [10, 20] was used to measure participants' subjective level of mental effort. Participants rated five of its six dimensions (mental demand, physical demand, temporal demand, effort and frustration) on a 20-point scale ranging from 0 (very low) to 20 (very high). The endpoints of the sixth dimension (own performance) were success and failure. Finally, the overall workload/mental effort was calculated across these six dimensions.

In the performance questionnaire, participants were asked to write a pseudocode based on a similar scenario to the task that they learnt using the system. This was undertaken immediately after interacting with the prototype (Instant Recall).

Learning efficiency was determined based on the ratio of performance to the difficulty of the learning task as proposed in [22]. The performance of each study condition was based on the recall test participants obtained after completing the task. The difficulty of the task was based on the mental effort they invested during the learning phase Performance and task difficulty data were then standardised using Formula 4.1 where z = z-score, r = raw data score, M = population mean, and SD = standard deviation.

$$z = \frac{r - M}{SD} \tag{4.1}$$

Next, the learning efficiency (E) was assessed for each of the two study conditions using the Formula 4.2 [5, 9, 22] where E = Learning efficiency, z_P = Average performance in Z-scores, and z_M = Average task difficulty in Z-scores. Note that square root of 2 in this formula comes from the general formula for the calculation of distance from a point, $p(x, y)$, to a line, $ax + by + c = 0$.

$$E = \frac{z_P - z_M}{\sqrt{2}} \tag{4.2}$$

To measure the usability of the system, we used the System Usability Scale (SUS), a ten question questionnaire originally created by Brooke, 1996 [4], on a five-point Likert scale, ranging from “Strongly” agree at 1 to “Strongly disagree” at 5. For measuring the user experience we used the short version of the User Experience Questionnaire (UEQ-S) [26, 27] with eight items/questions, reported on a 7-point Likert scale. The first four represent pragmatic qualities (Perspicuity, Efficiency and Dependability) and the last four hedonic qualities (Stimulation and Novelty) [26].

5 Results

The results and analysis are based on the 15 participants who had completed all the parts of one of the two conditions, i.e. the basic training, the programming task and the post-questionnaires including mental effort and instant recall test.

Table 1: Descriptive statistics for non-AR and AR condition for all dependant variables.

	Condition	**Mean**	**Median**	**SD**	**Minimum**	**Maximum**
Mental effort	Desktop	40.2	39.2	16.19	20.0	67.5
	AR	27.7	32.1	9.61	13.3	33.3
Performance	Desktop	72.5	70.0	32.84	0	100
	AR	85.0	85.0	17.32	70	100
Time in seconds	Desktop	522.0	600.0	159.60	160.9	600.0
	AR	474.5	512.5	157.64	273.0	600.0
SUS Score	Desktop	70.3	73.8	12.50	42.5	82.5
	AR	80.0	82.5	18.71	55.0	100.0

5.1 Performance, Mental Effort and Time

The descriptive statistics of the performance, the mental effort and the task completion time for both conditions are shown in Table 1. Results indicate that the participants' average performance score for non-AR condition is $\overline{x} = 72.5\%$, $SD = 32.84$ while the average mental effort invested in the task is $\overline{x} = 40.2$, $SD = 16.19$. Further, the participants' average task completion time for the non-AR condition is $\overline{x} = 522.0$s, $SD = 159.60$s.

For the AR condition we can see improvement in the results where the participants' average performance score is $\overline{x} = 85.0\%$, $SD = 17.32$, average mental effort invested on the task is $\overline{x} = 27.7$, $SD = 9.61$ and the average task completion time is $\overline{x} = 474.5$s, $SD = 157.64$s.

5.2 Learning Efficiency

The learning efficiency for each participants in both conditions are shown in Table 2. The definition and the calculation of learning efficiency is described in Section 4.4 Data Collection.

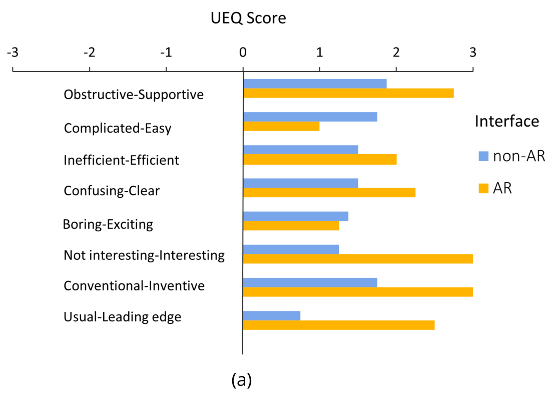
Table 2: Learning efficiency for each participant in the non-AR and AR condition.

Interface	**ParticipantID**	**Raw Data**		**Z-Score**		**Learning Efficiency**
		Perfomance	Mental Effort	Perfomance	Mental Effort	
non-AR	1	70	55.83	-0.16	1.07	-0.87
	2	100	20.00	0.88	-1.07	1.39
	3	70	43.33	-0.16	0.32	-0.34
	4	70	13.33	-0.16	-1.47	0.93
	5	70	67.50	-0.16	1.77	-1.36
	6	0	20.00	-2.59	-1.07	-1.07
	7	100	53.33	0.88	0.92	-0.02
	8	70	29.17	-0.16	-0.53	0.26
	9	100	36.67	0.88	-0.08	0.68
	10	70	40.83	-0.16	0.17	-0.23
	11	100	37.50	0.88	-0.03	0.65
AR	1	70	32.50	-0.87	0.50	-0.97
	2	70	31.67	-0.87	0.41	-0.90
	3	100	33.33	0.87	0.59	0.20
	4	100	13.33	0.87	-1.50	1.67

5.3 System Usability and User Experience

The answers to System Usability Scale (SUS) questions/items are reported on a 5-point Likert scale. The SUS scores are calculated as follows: for each of the odd numbered questions, subtract one from the user response, while for each of the even numbered questions, subtract their response from five and, add up the converted responses for each user and multiply the total by 2.5. This converts possible values to the range of 0 to 100 instead of 0 to 40. These adjustments are kept throughout the rest of the analysis.

The results of the SUS are given in Table 1. For the non-AR condition we have the mean $\overline{x} = 70.3$ and the standard deviation $SD = 12.50$. For the AR condition the result is $\overline{x} = 80.0$, $SD = 18.71$.



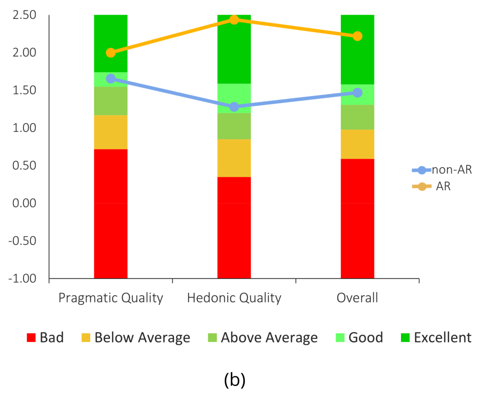


Figure 12: User experience questionnaire: (a) Mean values per item/question; (b) Mean values of UEQ factors (pragmatic and hedonic) and overall.

The UEQ-s questionnaire provides a benchmark to compare user experience between different systems [11]. It measures pragmatic qualities of a system (including efficiency, perspicuity and dependability) and hedonic qualities (including stimulation and novelty). We adapted the standard method as suggested in [26, 27] for calculating the scale means for each factor individually (efficiency, perspicuity, dependability, stimulation and novelty) and to obtain values for pragmatic quality, hedonic quality and overall user experience for both conditions.

The calculated mean for every item is shown in Figure 12b. All the evaluations are positive for both conditions, with the AR condition scoring higher. The benchmarks for an excellent score is: pragmatic > 1.73, hedonic > 1.55, overall > 1.58 (dark green in Figure 12b). According to this we see that the non-AR score is in the “good” area while the AR score in in the “excellent” area.

6 Discussion

Despite time constraints, and other limitations discussed below, the initial findings from the study are promising. Based on current data, both conditions (i.e., 2D desktop interface—non-AR and Tangible AR interface—AR) foster creativity and aid in the learning of programming, with the AR condition showing a more pronounced effect. While acknowledging certain limitations in our work, we will discuss them alongside our plans for future research. Additionally, we offer several design recommendations for upcoming studies.

6.1 Limitations

We recognise the importance of expanding the sample pool to enhance the robustness of our study and balance the number of participants in each group. By increasing the number of participants and generalising the sample, we aim to derive more meaningful results and draw conclusions that accurately reflect potential significant differences. In addition, as most of the participants at the scouts camp who were willing to participate the Desktop interface were males, there is a gender bias in the study conducted.

Moreover, the initial goal was to conduct the study with subjects ranging in age from 10 to 15 years old. However, we had difficulties in recruiting participants in the limited time frame. As a result, we decided to broaden this age range to 18-year-old or older students with no prior programming experience.

The time constraints also impacted the refinement of the prototypes, including enhancing magnetic effects in the Desktop interface and improving animations in the AR interface. Further details can be explored in the Design Recommendations subsection. However, this aspect will be addressed in future studies, where the participant pool will also be expanded.

We were additionally constrained by the performance of one of the machines used to test the desktop version. Consequently, some participants encountered latency issues.

6.2 Design Recommendations

In addition to aforementioned future plans, we also plan to implement changes to the interfaces to enhance user-friendliness. Specifically, we will address the size of the pieces used for the tangible element in the AR prototype. Enlarging these blocks and enabling them to resize in relation to their neighbors when dragged can improve usability. Additionally, as our 2D prototype couldn't be tested on a tablet as initially intended and was instead used on a laptop, we will redesign the new 2D version to support tablets. Choosing a guiding avatar that is more visually attractive and animated is also a viable option.

Furthermore, we plan to redesign the appearance of the blocks in the 2D version. As the forms require precise alignment with one another, this redesign is currently underway. Additionally, we will introduce colour coding for the blocks. Currently, the green colour of the event blocks is associated with the notion of "start". We aim to conduct a study to determine suitable colours for all block types, ensuring that users can interpret the colour as indicative of a specific meaning or activity.

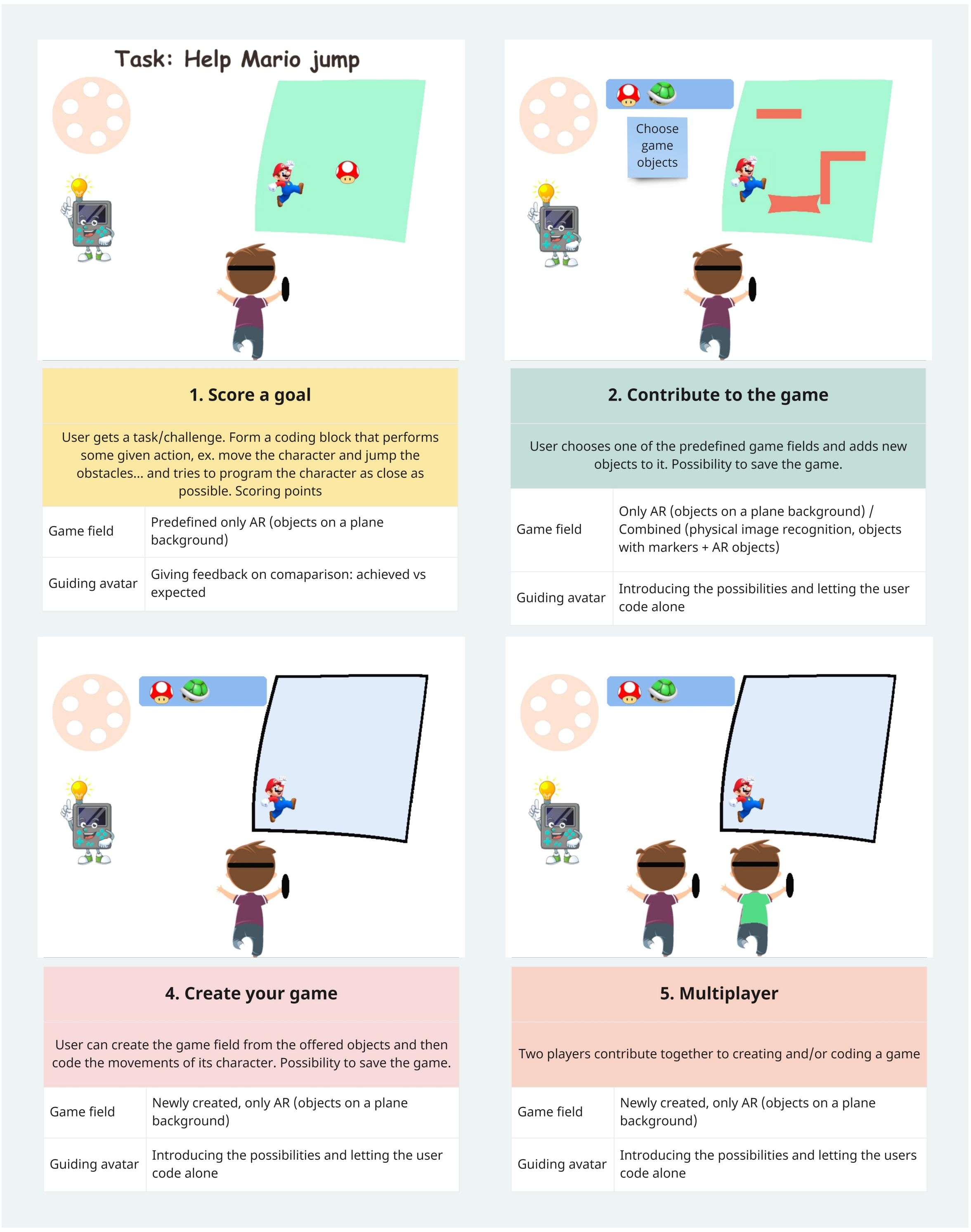
Furthermore, cognitive load may be added as a new variable, and we could test with children of various age groups who would be given simpler and more challenging situations encompassing more programming topics. The right cognitive load per age group might therefore be estimated.

7 Conclusion and Future Work

pARt Blocks is a project aimed at facilitating children's learning, enjoyment, and harnessing the benefits of technology. Despite limitations, the initial findings are quite compelling and the theoretical backdrop support them. Based on the results of the user experience, we can see that students perceive these approaches as both creative and conducive to their learning. According to the System Usability Scale (SUS) and User Experience Questionnaire (UEQ) scores, the current AR interface outperforms the non-AR interface. Participants indicated that the AR interface is more inventive, interesting, supportive, and clear (Figure 12-a). The mean performance is higher in the AR condition, while the mental effort is lower. However, further conclusions can be drawn with a t-statistic after recruiting more participants.

The prototype could undergo various enhancements. Future iterations should place a stronger emphasis on the gamification aspect. Additional game objectives could be incorporated, introducing a scenario with additional activities structured into stages, allowing players to earn points for each level completed. A virtual store could be introduced, providing users with the opportunity to purchase items using accumulated points. To enhance interactivity, players could be given the ability to design their own game field through a 2D interface or create a physical game field using objects associated with a marker. This approach would instill a sense of building and programming a personalised game. Moreover, the prototype could be further personalised by integrating tailored guidance and increased involvement from the avatar serving as a guide throughout the experience.

A multiplayer concept in which the children construct the code jointly may be investigated. This may be especially intriguing in the tactile interface, where they would have to share the pieces of blocks and think aloud. While testing was being conducted at the scout camp, the youngsters exhibited a strong desire to work together to solve the problem, but we needed to separate them in order to get reliable results for the present research. However, this might be leveraged as a benefit and a new research element.



1. Score a goal

User gets a task/challenge. Form a coding block that performs some given action, ex. move the character and jump the obstacles... and tries to program the character as close as possible. Scoring points

Game field	Predefined only AR (objects on a plane background)
Guiding avatar	Giving feedback on comaparison: achieved vs expected

2. Contribute to the game

User chooses one of the predefined game fields and adds new objects to it. Possibility to save the game.

Game field	Only AR (objects on a plane background) / Combined (physical image recognition, objects with markers + AR objects)
Guiding avatar	Introducing the possibilities and letting the user code alone

4. Create your game

User can create the game field from the offered objects and then code the movements of its character. Possibility to save the game.

Game field	Newly created, only AR (objects on a plane background)
Guiding avatar	Introducing the possibilities and letting the user code alone

5. Multiplayer

Two players contribute together to creating and/or coding a game

Game field	Newly created, only AR (objects on a plane background)
Guiding avatar	Introducing the possibilities and letting the users code alone

Figure 13: A rough sketch of the future work ideas

8 Povzetek naloge v slovenskem jeziku

Integrirana razvojna okolja (IDE) standardnih višjenivojskih programskih jezikov (PL) uporabljajo besedilne vmesnike, ki vključujejo prostor za pisanje programa, opozorila, dnevnike, razvijalsko konzolo in druge podobne elemente. Ta način omogoča izkušenim razvijalcem hiter razvoj in preprost prehod med različnimi platformami in ogrodji. Vendar imajo lahko predvsem otroci začetniki s takim načinom programiranja težave in bi se ga s pomočjo vizualnih elementov lažje naučili [18]. Raziskovalci so zato razvili programske jezike za vizualno programiranje (VPL), kjer se namesto besedila uporabljajo vizualni formalizmi [3] kot so vizualni bloki, ki predstavljajo dele kode. Jezik Scratch je eden najbolj prepoznavnih VPL-jev, ki so ga leta 2017 razvili v Medijskem laboratoriju MIT [19]. Scratch je navdihnil še druge pristope k programiranju, ki temeljijo na blokih [13, 21].

Poleg učenja z vizualnimi elementi na zaslonu, raziskave na področju interakcije človek-računalnik (HCI) raziskujejo še druge metode učenja programiranja s podporo tehnologije, kot je na primer učlenje z uporabo “oprijemljivih vmesnikov”. Ti vmesniki omogočajo upravljanje digitalne vsebine s pomočjo fizičnih predmetov in so ponavadi namenjeni mlajšim otrokom [6, 15, 16]. Nekateri do sedaj predstavljeni prototipi uporabljajo oprijemljive fizične elemente, kot so kocke ali karte, ki vključujejo elektroniko za njihovo prepoznavanje [6, 15], nekateri pa temeljijo na markerjih razširjene resničnosti (AR) [16].

Nekaj raziskav je torej že bilo opravljenih na področju oprijemljivih vmesnikov pri učenju programiranja. Ugotovitve kažejo, da tak način učenja prispeva k večjim užitkom, povečani motivaciji in boljši angažiranosti. Vendar raziskav, ki bi opazovali učni napor in učne rezultate pri učenju programiranja, pri čemer bi primerjali oprijemljive vmesnike s standardnim vizualnim programiranjem v načinu “povleci in spusti” na zaslonu, skorajda ni. Poleg tega, nobena obstoječa raziskava ne združuje oprijemljive interakcije in dopolnjene resničnosti z uporabo naglavnih prikazovalnikov (HMD). Slednji namreč podpirajo večjo potopitveno izkušnjo in s tem privlačnejši učni proces, saj imajo uporabniki HMD v primerjavi z drugimi napravami, kot so telefoni, proste roke [28].

Da bi odpravili omenjeno vrzel, smo izdelali prototip *pARt Blocks*, za učenje programiranja, ki združuje oprijemljiv vmesnik in vizualne elemente v dopolnjeni resničnosti (AR), vključno z virtualnim avatarjem in igrifikacijo preko zbiranja točk, kot je prikazano na Sliki 4 - a. Elementi AR, vidni skozi HMD, povečajo angažiranost in uporabnika spodbujajo k “zmagi” s pravilno zloženim programom. Po metodi Montessori taka interakcija s fizičnimi predmeti izboljša aktivno učenje [1, 14].

Da bi raziskali prednosti izdelanega prototipa, smo izvedli uporabniško študijo med dvema skupinama. Ena je uporabljala oprijemljiv vmesnik z AR elementi, druga pa standardni vmesnik na standardnem zaslonu, ki uporablja “povleci in spusti” način interakcije (kot je prikazano na sliki 4 1 - b). Naše ugotovitve lahko pomagajo učiteljem in oblikovalcem učnih pripomočkov pri učinkovitem načrtovanju učenja kompleksnih in abstraktnih konceptov programiranja, vključno s konstrukti, zankami in drugimi elementi, preko oprijemljivih uporabniških vmesnikov.

Z uporabniško študijo smo želeli odgovoriti na naslednji raziskovalni vprašanji:

1. Ali bo oprijemljivi vmesnik z uporabo AR vplival na učne rezultate pri učenju programiranja?
2. Ali bo oprijemljivi vmesnik z uporabo AR vplival na stopnjo angažiranosti pri učenju programiranja?

V raziskavi smo tako primerjali običajno VPL programiranje na zaslonu z oprijemljivim vmesnikom, ki je omogočal enake funkcionalnosti. Raziskavo smo zasnovali tako, da so uporabniki morali napisati program, ki je virtualni lik (deklico Anno) premikal po igralnem polju. Igralno polje je predstavljalo zavetišče za pse, kjer je vsak posamezen pes bil v svoji ogradi. Anna se je morala sprehoditi od enega psa do grugega in glede na to, ali je bil posamezen pes lačen, ga je zeblo ali oboje, mu je morala dati hrano, odejo ali oboje. Pri uspešno opravljeni nalogi, so udeleženci dobili točke kot je to navada v videoigrah. Pri tem so uporabniki morali uporabiti naslednje koncepte:

- Začetno dejanje (dogodek) - ko se začne izvajanje
- Ponavljajoče se dejanje (zanka) - obiščite vsakega psa
- Pogojne dodelitve (konstrukcija if / else if /else, logični operaterji, spremenljivke)

Za izvedbo programa pa so uporabniki imeli na izbiro naslednje bloke, ki so jih sestavljali kot kocke v program:

1. Bloki dogodkov: uporanik jih uporabi za določitev, kdaj naj se nekaj zgodi.
2. Nadzorni bloki: uporabnik jih uporabi za zanke in izraze if/else.

3. Operatorski bloki: uporabnik jih uporabi za logične in primerjalne operatorje.

4. Bloki spremenljivk: uporabnik jih uporabi za prednastavljene spremenljivke.

5. Bloki premikov: uporabnik jih uporabi za določitev gibanja lika (Anna)

Kljub določenim omejitvam pri naši raziskavi (manjše število udeležencev, neenakomerna porazdelitev, več moških), so začetne ugotovitve precej spodbudne, podprte pa so tudi s teoretičnim ozadjem predstavljenim v drugem poglavju. Rezultati uporabniške izkušnje kažejo, da so udeleženci nalogo dojemali kot ustvarjalno in spodbudno. Na podlagi rezultatov ocene uporabnosti (SUS) in ocene uporabniške izkušnje (UEQ) omenjeni oprijemljivi vmesnik presega vmesnik na standardnem zaslonu. Udeleženci so označili oprijemljiv vmesnik kot inovativnejši, zanimivejši, in jasnejši (slika 12-a). Ravno tako je bila učinkovitost višja pri oprijemljivem vmseniku, medtem ko je mentalni napor nižji.

9 Bibliography

[1] H. Bienen. *The Montessori Method.* Routledge, Forbes Boulevard, Lanham, USA, 2017. *(Cited on pages 2, 6, and 26.)*

[2] S. Boonbrahm, P. Boonbrahm, C. Kaewrat, P. Pengkaew, and P. Khachorn-charoenkul. Teaching fundamental programming using augmented reality. *International Journal of Interactive Mobile Technologies (iJIM)*, 13(07):pp. 31–43, Jul. 2019. *(Cited on pages VIII, 7, and 8.)*

[3] M. Boshernitsan and M. S. Downes. *Visual programming languages: A survey.* Citeseer, Berkeley, California USA, 2004. *(Cited on pages 1 and 25.)*

[4] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996. *(Cited on page 17.)*

[5] R. C. Clark, F. Nguyen, and J. Sweller. *Efficiency in learning: Evidence-based guidelines to manage cognitive load.* John Wiley & Sons, 2011. *(Cited on page 17.)*

[6] B. Cleto, J. M. Moura, L. Ferreira, and C. Sylla. Codecubes-playing with cubes and learning to code. In *Interactivity, Game Creation, Design, Learning, and Innovation*, pages 538–543. Springer, GEWERBESTRASSE 11, Cham, Switzerland, 2018. *(Cited on pages 1, 9, and 25.)*

[7] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-computer interaction.* Pearson Education, 2004. *(Cited on page 7.)*

[8] G. Futschek and J. Moschitz. Learning algorithmic thinking with tangible objects eases transition to computer programming. In *International conference on informatics in schools: Situation, evolution, and perspectives*, pages 155–164. Springer, 2011. *(Cited on pages VIII, 6, and 9.)*

[9] A. K. Halabi. Applying an instructional learning efficiency model to determine the most efficient feedback for teaching introductory accounting. *Global Perspectives on Accounting Education*, 3(1):6, 2006. *(Cited on page 17.)*

[10] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006. *(Cited on pages 15 and 16.)*

[11] A. Hinderks, M. Schrepp, and J. Thomaschewski. A benchmark for the short version of the user experience questionnaire. In *WEBIST*, pages 373–377, 2018. *(Cited on page 20.)*

[12] A. Idlbi. Taking kids into programming (contests) with scratch. *Olympiads in Informatics*, 3(1):17–25, 2009. *(Cited on page 5.)*

[13] J. M. S. III, G. Nodalo, J. Valenzuela, and J. A. Deja. Explore, edit, guess: Understanding novice programmers' use of codeblocks for regression experiments. In *Proceedings of the 6th HCI Slovenia Conference 2021*, pages 3–17, Koper, Slovenia, 2021. CEUR Workshop Proceedings. *(Cited on pages 1 and 25.)*

[14] S. R. Klemmer, B. Hartmann, and L. Takayama. How bodies matter: Five themes for interaction design. In *Proceedings of the 6th Conference on Designing Interactive Systems*, DIS '06, page 140–149, New York, NY, USA, 2006. Association for Computing Machinery. *(Cited on pages 2, 6, and 26.)*

[15] V. Koushik, D. Guinness, and S. K. Kane. Storyblocks: A tangible programming game to create accessible audio stories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. *(Cited on pages 1 and 25.)*

[16] D. Krpan, S. Mladenović, and B. Ujević. Tangible programming with augmented reality. In *12th International Technology, Education and Development Conference*, pages 4993–5000, Valencia, Spain, 2018. IATED. *(Cited on pages 1, 7, and 25.)*

[17] J. Liu and Q. Duan. Research of interaction design guided by five senses theory. In C. Stephanidis, editor, *HCI International 2019 - Posters*, pages 49–55, Cham, 2019. Springer International Publishing. *(Cited on page 6.)*

[18] R. E. Mayer. Multimedia learning. In *Psychology of learning and motivation*, volume 41, pages 85–139. Elsevier, Radarweg 29, 1043 NX Amsterdam, The Netherlands, 2002. *(Cited on pages 1 and 25.)*

[19] MIT. Scratch developers, 2006. *(Cited on pages 1 and 25.)*

[20] NASA. Nasa tlx: Task load index, 2006. *(Cited on page 16.)*

[21] G. Nodalo, J. M. Santiago, J. Valenzuela, and J. A. Deja. On building design guidelines for an interactive machine learning sandbox application. In *Proceedings of the 5th International ACM In-Cooperation HCI and UX Conference*, CHIuXiD'19, page 70–77, New York, NY, USA, 2019. Association for Computing Machinery. *(Cited on pages 1 and 25.)*

[22] F. G. Paas and J. J. Van Merriënboer. The efficiency of instructional conditions: An approach to combine mental effort and performance measures. *Human factors*, 35(4):737–743, 1993. *(Cited on page 17.)*

[23] R. Pelánek and T. Effenberger. Design and analysis of microworlds and puzzles for block-based programming. *Computer Science Education*, 32(1):66–104, 2022. *(Cited on page 4.)*

[24] I. Radu and B. MacIntyre. Augmented-reality scratch: a children's authoring environment for augmented-reality experiences. In *Proceedings of the 8th International Conference on Interaction Design and Children*, pages 210–213, 2009. *(Cited on page 7.)*

[25] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009. *(Cited on pages VIII, 4, and 5.)*

[26] M. Schrepp. Ueq-user experience questionnaire, 2019. *(Cited on pages 17 and 20.)*

[27] M. Schrepp, A. Hinderks, and J. Thomaschewski. Design and evaluation of a short version of the user experience questionnaire (ueq-s). *IJIMAI*, 4(6):103–108, 2017. *(Cited on pages 17 and 20.)*

[28] M. Weerasinghe. Instructional guidance in extended reality for learning. In *Adjunct Publication of the 23rd International Conference on Mobile Human-Computer Interaction*, New York, NY, USA, 2021. Association for Computing Machinery. *(Cited on pages 1 and 25.)*

[29] M. Weerasinghe, V. Biener, J. Grubert, A. J. Quigley, A. Toniolo, K. Č. Pucihar, and M. Kljun. Vocabulary: Learning vocabulary in ar supported by keyword visualisations. *arXiv preprint arXiv:2207.00896*, 2022. *(Cited on page 8.)*

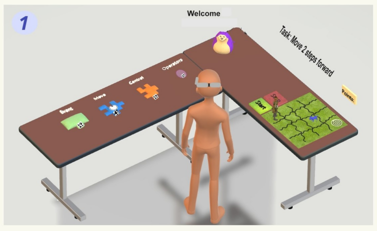
[30] M. Weerasinghe, A. Quigley, J. Ducasse, K. Čopič Pucihar, and M. Kljun. *Educational Augmented Reality Games*, pages 3–32. Springer International Publishing, Cham, 2019. *(Cited on page 10.)*

[31] M. Weerasinghe, A. Quigley, K. Č. Pucihar, A. Toniolo, A. Miguel, and M. Kljun. Arigatō: Effects of adaptive guidance on engagement and performance in augmented reality learning environments. *arXiv preprint arXiv:2207.00798*, 2022. *(Cited on page 6.)*

[32] N. Yannier, K. R. Koedinger, and S. E. Hudson. Learning from mixed-reality games: Is shaking a tablet as effective as physical observation? In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1045–1054, 2015. *(Cited on page 6.)*

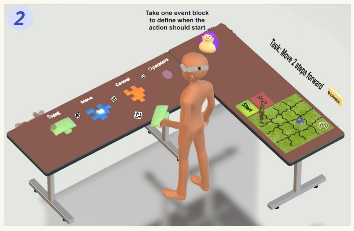
Appendices

A The Storyboard of AR Interface



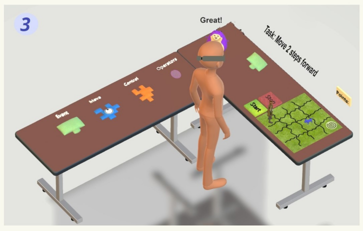
General positioning of the objects

Welcoming the user to pARt Blocks, introduction with the game parts


2
Take one event block to define when the action should start
Task: Move 2 steps forward

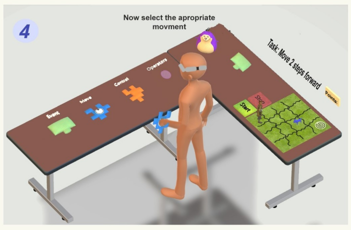
Start programming

The user then sees the task and the avatar points to the next step. At this example, user needs to take one of the event blocks and place on the other desk.



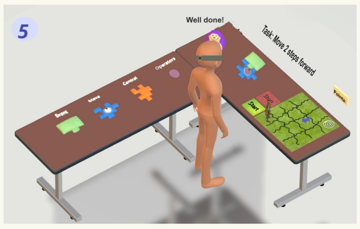
First step done

User puts one of the blocks and the avatar congrats him.



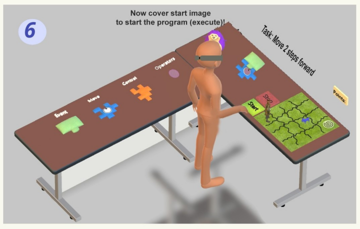
Next programming step

Avatar gives hint what should be done next in order to complete the task.



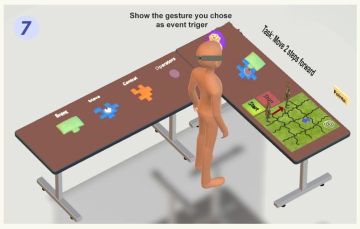
Finishing with the programming part

User has formed the block based code (connected the blocks) and the avatar points to that.


6
Now cover start image to start the program (execute)!
Task: Move 2 steps forward

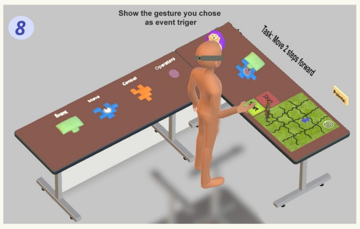
Time to execute

Avatar instructs that it is time to start the game and see the result (execute the code).


7
Show the gesture you chose as event triger
Task: Move 2 steps forward

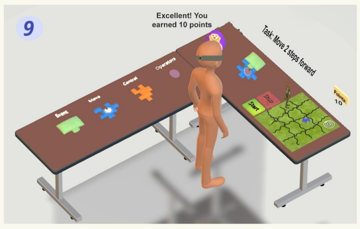
Triggering event

The user is instructed by the avatar how to trigger the programmed event


8
Show the gesture you chose as event triger
Task: Move 2 steps forward

Watching the result

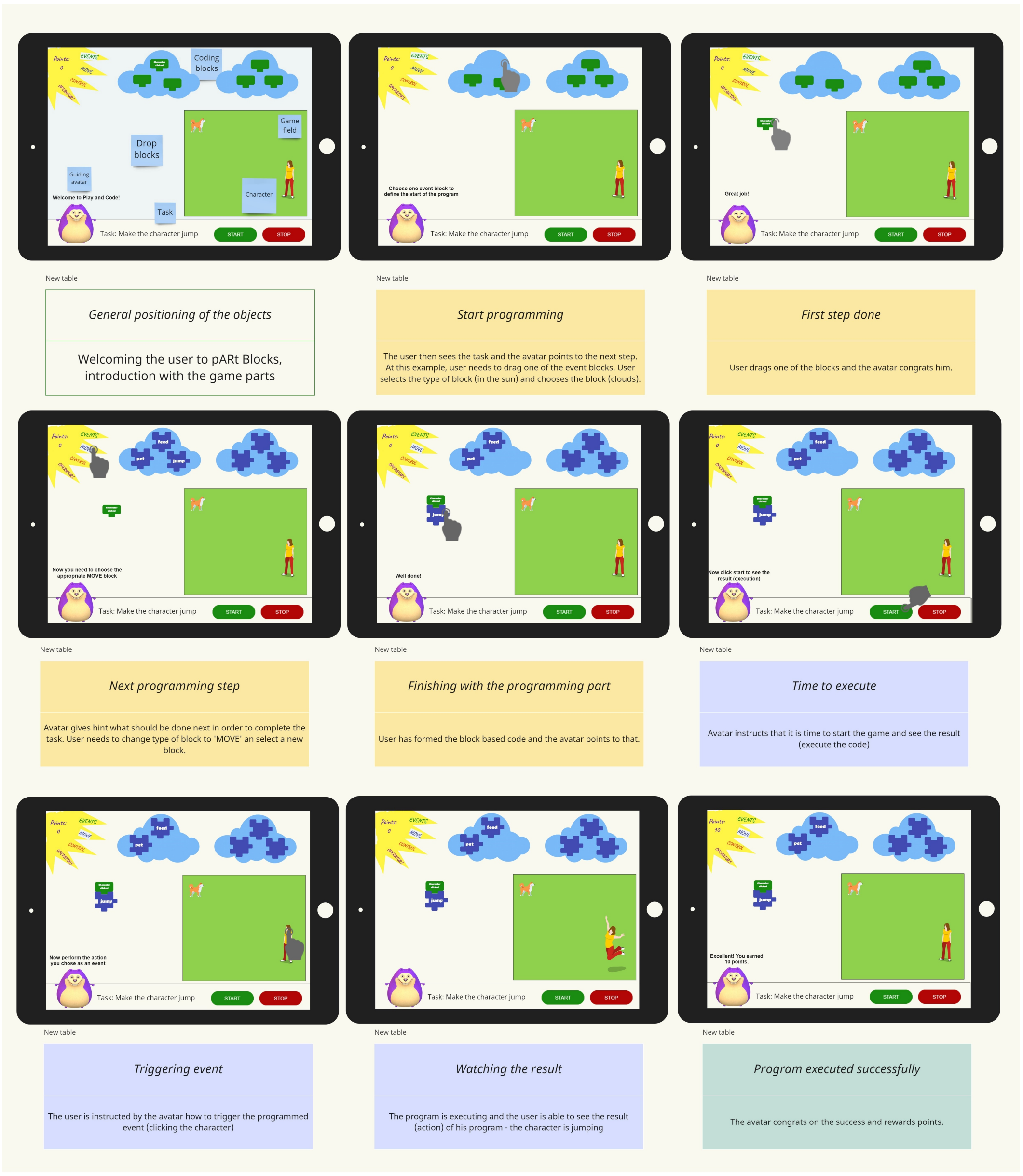
The program is executing and the user is able to see the result (action) of his program - the character is jumping



Program executed successfully

The avatar congrats on the success and rewards points.

B The Storyboard of non-AR (Desktop or Tablet) Interface



Welcome to Play and Code!

Task: Make the character jump

New table

General positioning of the objects

Welcoming the user to pARt Blocks, introduction with the game parts

Choose one event block to define the start of the program

Task: Make the character jump

New table

Start programming

The user then sees the task and the avatar points to the next step. At this example, user needs to drag one of the event blocks. User selects the type of block (in the sun) and chooses the block (clouds).

Great job!

Task: Make the character jump

New table

First step done

User drags one of the blocks and the avatar congrats him.

Now you need to choose the appropriate MOVE block

Task: Make the character jump

New table

Next programming step

Avatar gives hint what should be done next in order to complete the task. User needs to change type of block to 'MOVE' an select a new block.

Well done!

Task: Make the character jump

New table

Finishing with the programming part

User has formed the block based code and the avatar points to that.

Now click start to see the result (execution)

Task: Make the character jump

New table

Time to execute

Avatar instructs that it is time to start the game and see the result (execute the code)

Now perform the action you chose as an event

Task: Make the character jump

New table

Triggering event

The user is instructed by the avatar how to trigger the programmed event (clicking the character)

Task: Make the character jump

New table

Watching the result

The program is executing and the user is able to see the result (action) of his program - the character is jumping

Excellent! You earned 10 points.

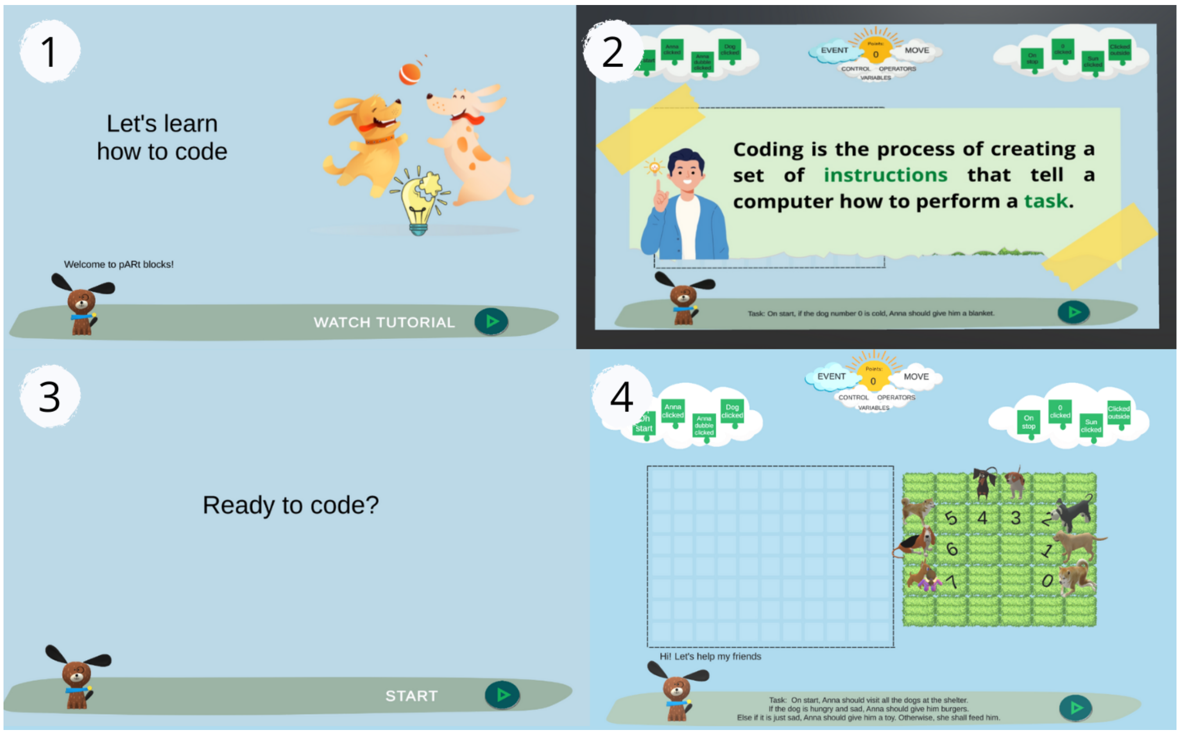
Task: Make the character jump

New table

Program executed successfully

The avatar congrats on the success and rewards points.

C Scenes from the Desktop Prototype

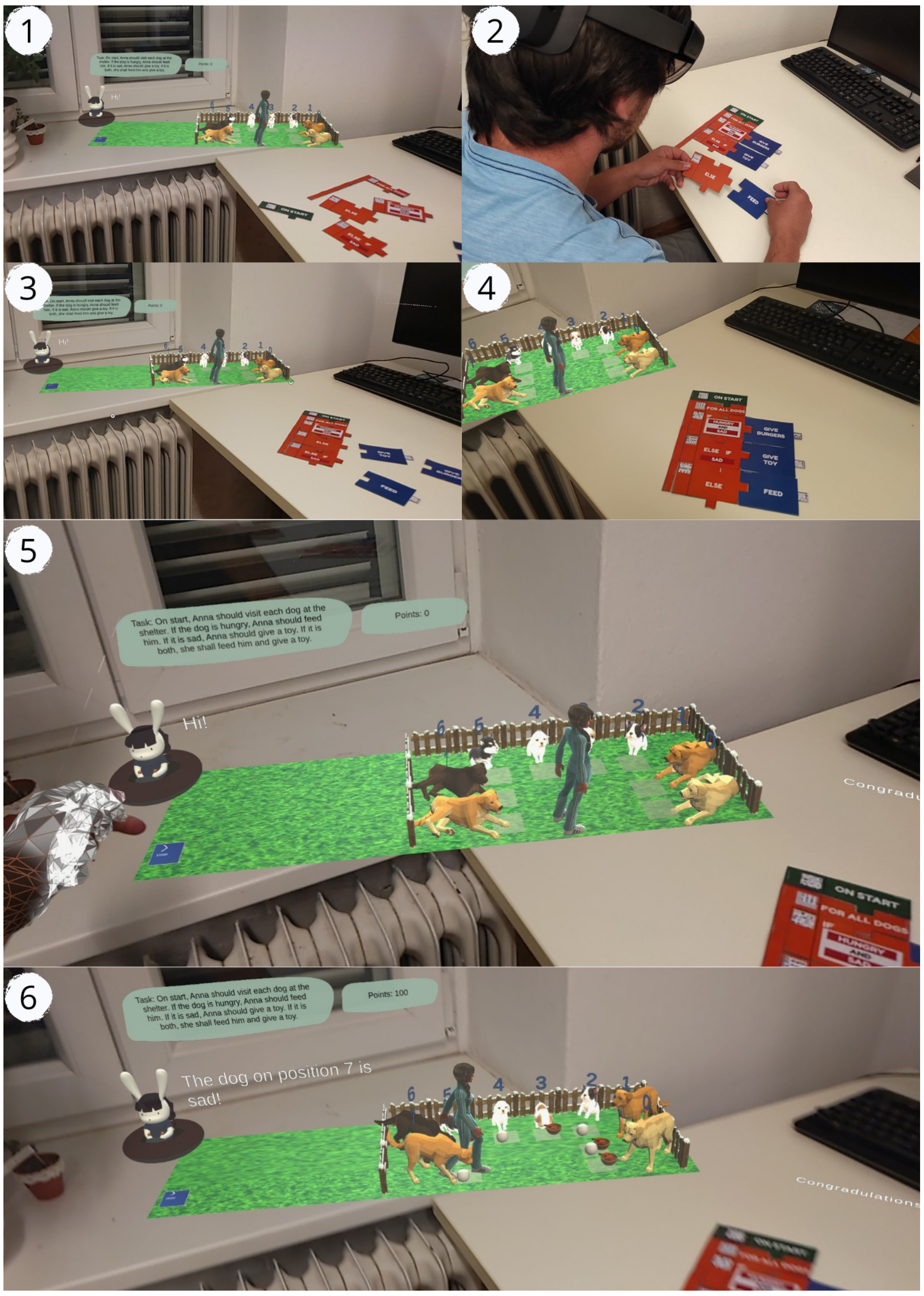


Coding is the process of creating a set of instructions that tell a computer how to perform a task.

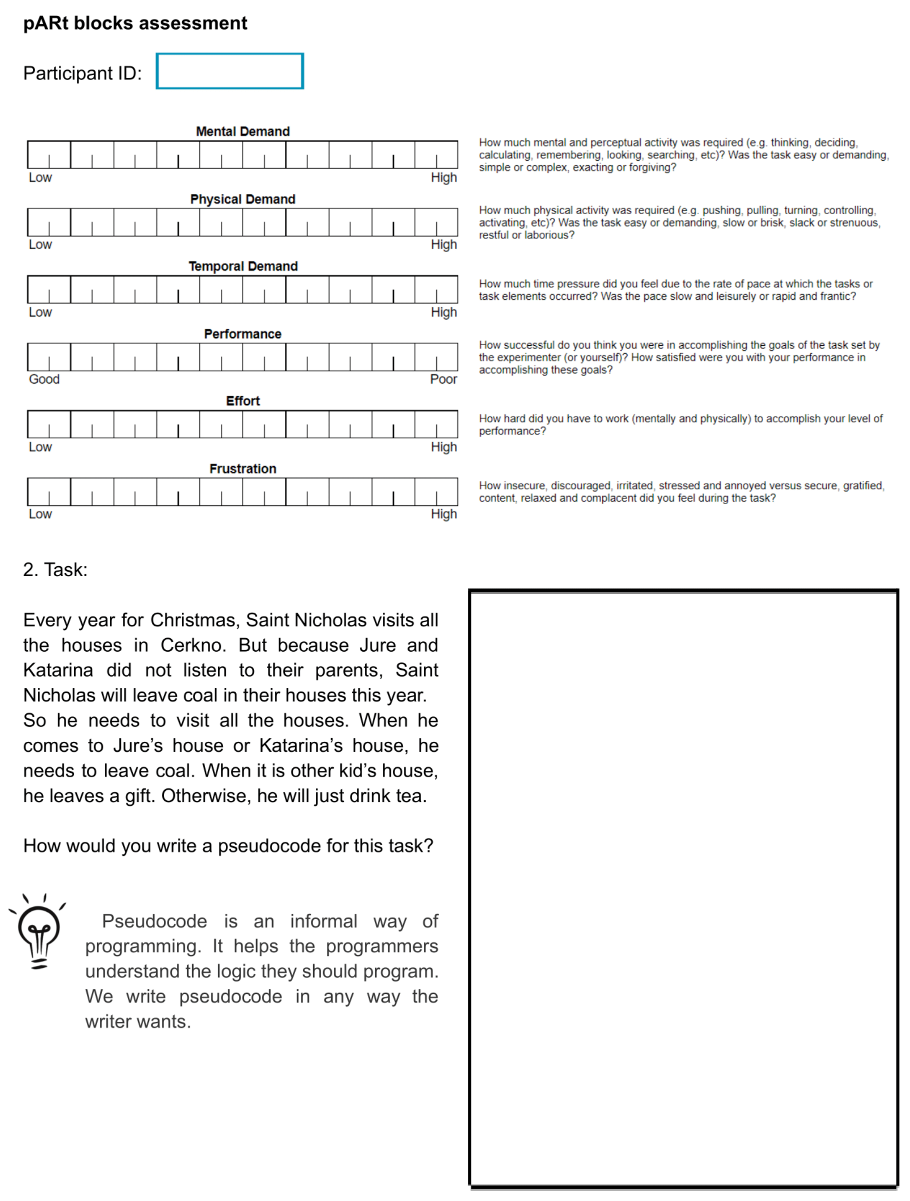
Task: On start, if the dog number 0 is cold, Anna should give him a blanket.

Task: On start, Anna should visit all the dogs at the shelter.
If the dog is hungry and sad, Anna should give him burgers.
Else if it is just sad, Anna should give him a toy. Otherwise, she shall feed him.

D Step-by-step AR Prototype


1
2
3
4
5
6
Task: On start, Anna should visit each dog at the shelter. If the dog is hungry, Anna should feed him. If it is sad, Anna should give a toy. If it is both, she shall feed him and give a toy.
Points: 0
Hi!
Points: 100
The dog on position 7 is sad!
Congradulations
ON START
FOR ALL DOGS
IF
HUNGRY
AND
SAD
ELSE IF
SAD
ELSE
GIVE BURGERS
GIVE TOY
FEED

E NASA-TLX Questionnaire



pARt blocks assessment

Participant ID:

Mental Demand

Low — High

How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Physical Demand

Low — High

How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

Temporal Demand

Low — High

How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

Performance

Good — Poor

How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

Effort

Low — High

How hard did you have to work (mentally and physically) to accomplish your level of performance?

Frustration

Low — High

How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

2. Task:

Every year for Christmas, Saint Nicholas visits all the houses in Cerkno. But because Jure and Katarina did not listen to their parents, Saint Nicholas will leave coal in their houses this year. So he needs to visit all the houses. When he comes to Jure's house or Katarina's house, he needs to leave coal. When it is other kid's house, he leaves a gift. Otherwise, he will just drink tea.

How would you write a pseudocode for this task?

Pseudocode is an informal way of programming. It helps the programmers understand the logic they should program. We write pseudocode in any way the writer wants.

F UEQ and SUS Questionnaires

UEQ and SUS

*Required

1. Participant ID *

____

UEQ questionnaire

Decide as spontaneously as possible which of the following conflicting terms better describes your learning experience. There is no "right" or "wrong" answer. Only your personal opinion counts!

2. Obstructive (impeding) / supportive (helpful) *

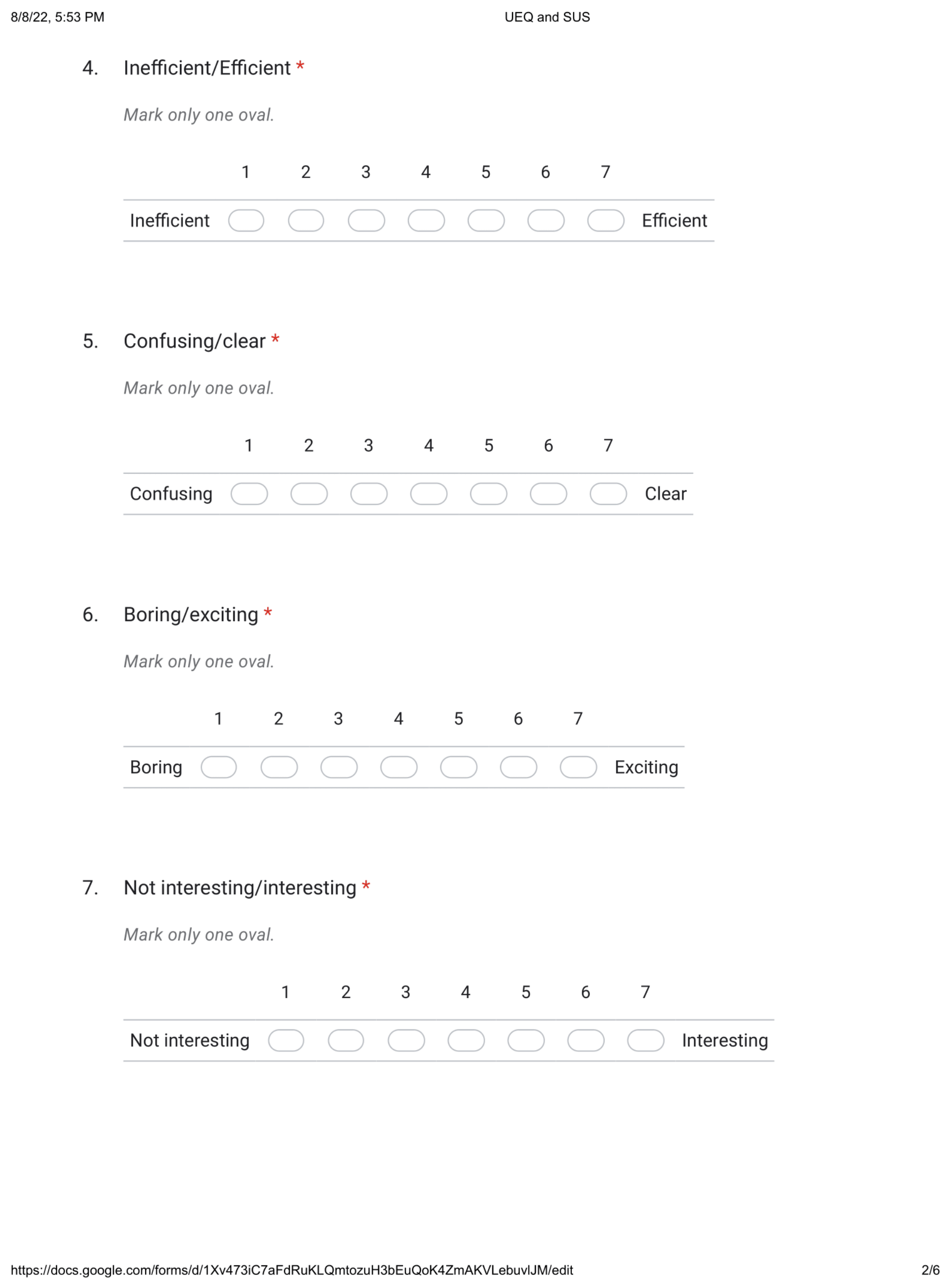
Mark only one oval.

	1	2	3	4	5	6	7	
Obstructive	◯	◯	◯	◯	◯	◯	◯	Supportive

3. Complicated/easy *

Mark only one oval.

	1	2	3	4	5	6	7	
Complicated	◯	◯	◯	◯	◯	◯	◯	Easy



4. Inefficient/Efficient *

Mark only one oval.

	1	2	3	4	5	6	7	
Inefficient								Efficient

5. Confusing/clear *

Mark only one oval.

	1	2	3	4	5	6	7	
Confusing								Clear

6. Boring/exciting *

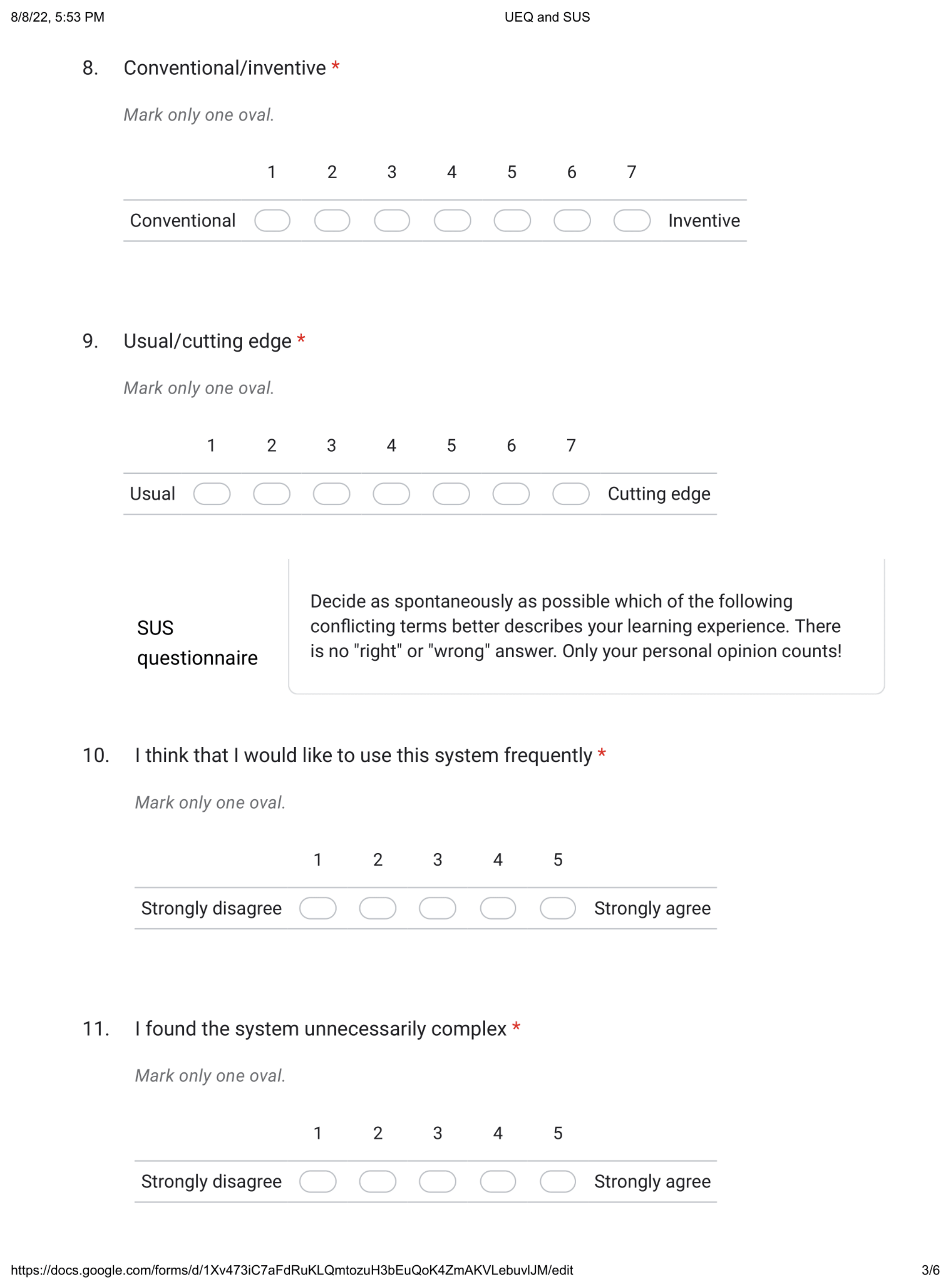
Mark only one oval.

	1	2	3	4	5	6	7	
Boring								Exciting

7. Not interesting/interesting *

Mark only one oval.

	1	2	3	4	5	6	7	
Not interesting								Interesting



8. Conventional/inventive *

Mark only one oval.

	1	2	3	4	5	6	7	
Conventional	◯	◯	◯	◯	◯	◯	◯	Inventive

9. Usual/cutting edge *

Mark only one oval.

	1	2	3	4	5	6	7	
Usual	◯	◯	◯	◯	◯	◯	◯	Cutting edge

SUS questionnaire

Decide as spontaneously as possible which of the following conflicting terms better describes your learning experience. There is no "right" or "wrong" answer. Only your personal opinion counts!

10. I think that I would like to use this system frequently *

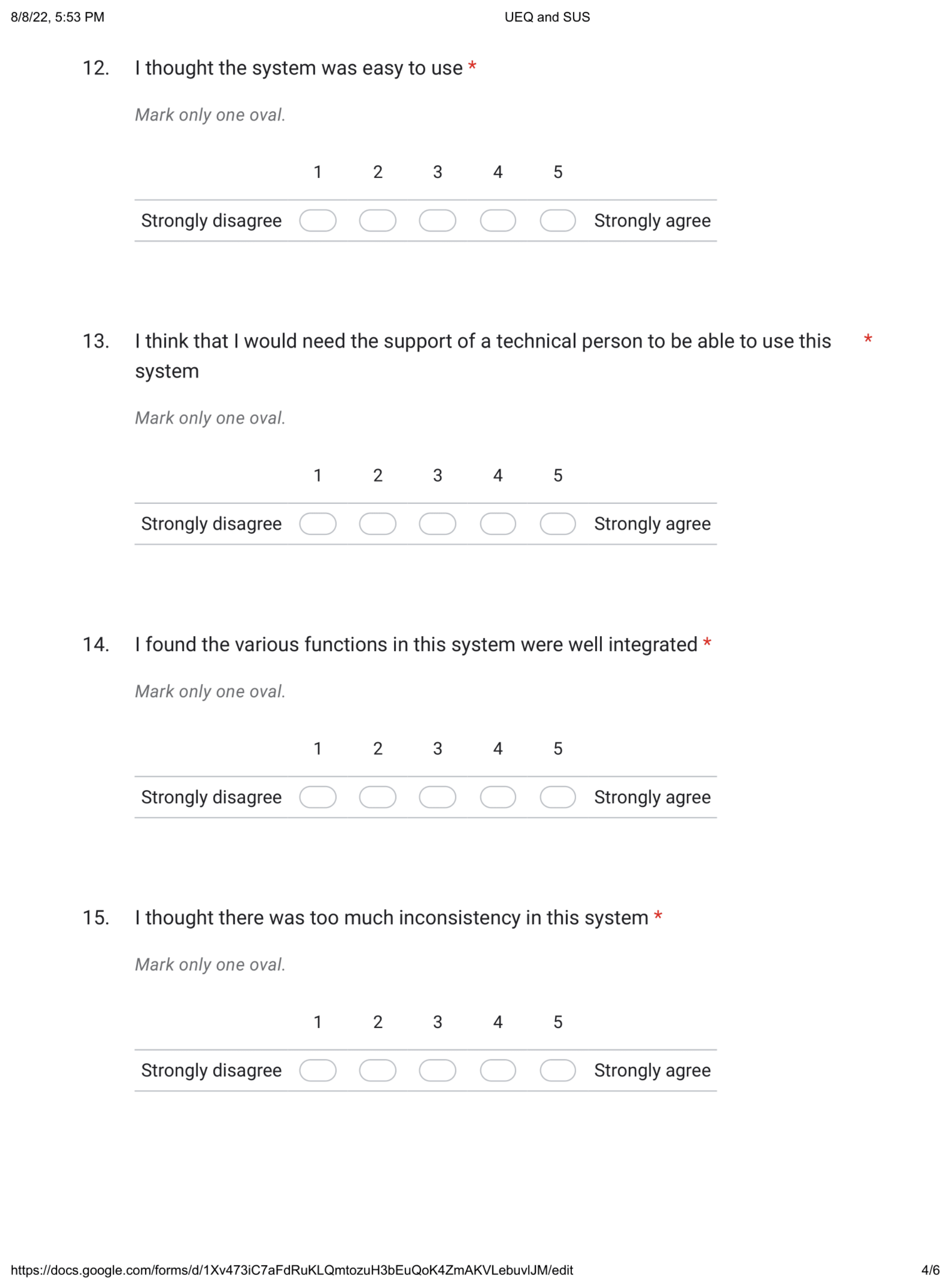
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

11. I found the system unnecessarily complex *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree



12. I thought the system was easy to use *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

13. I think that I would need the support of a technical person to be able to use this system *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

14. I found the various functions in this system were well integrated *

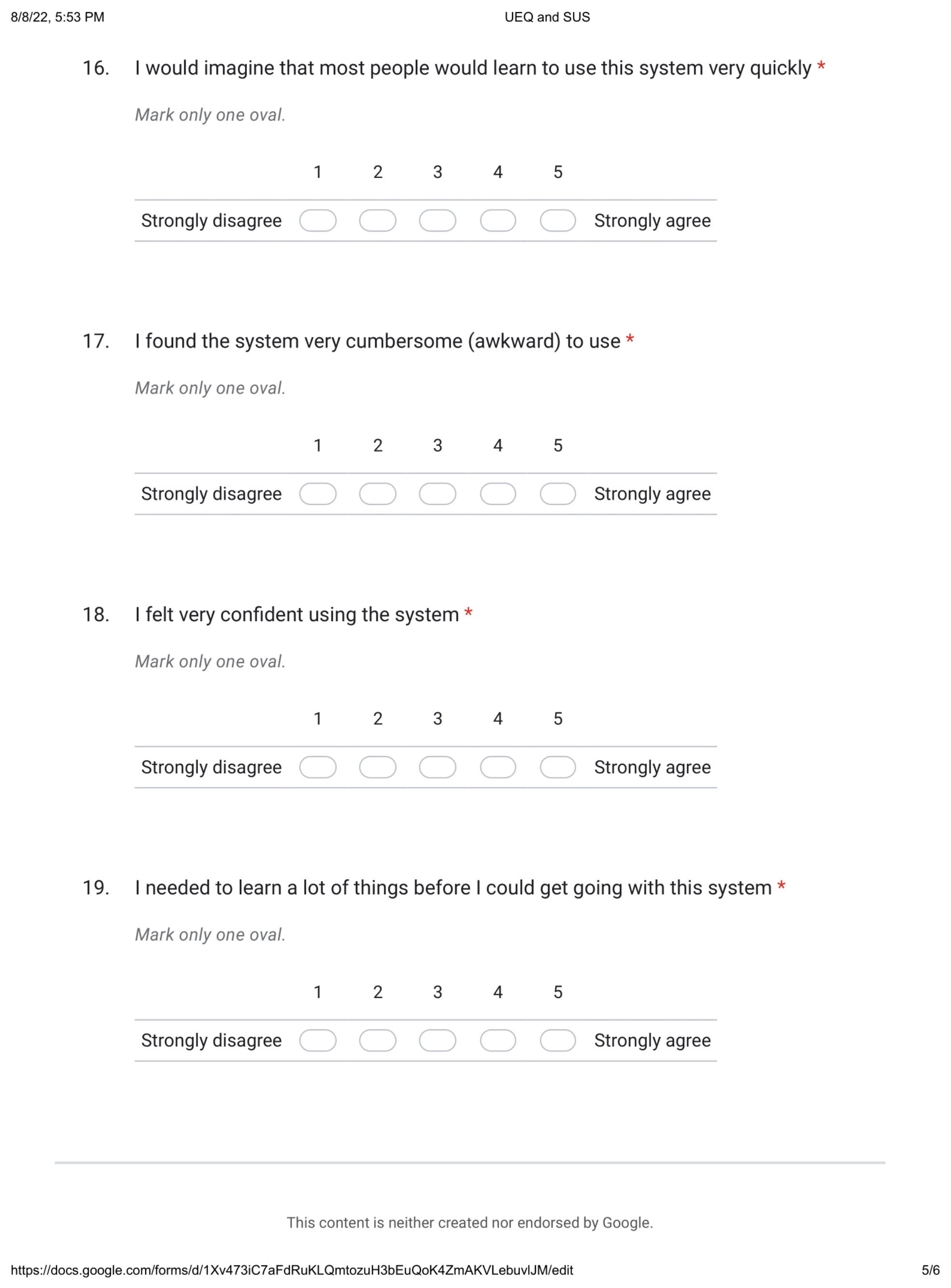
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

15. I thought there was too much inconsistency in this system *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree



16. I would imagine that most people would learn to use this system very quickly *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

17. I found the system very cumbersome (awkward) to use *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

18. I felt very confident using the system *

Mark only one oval.

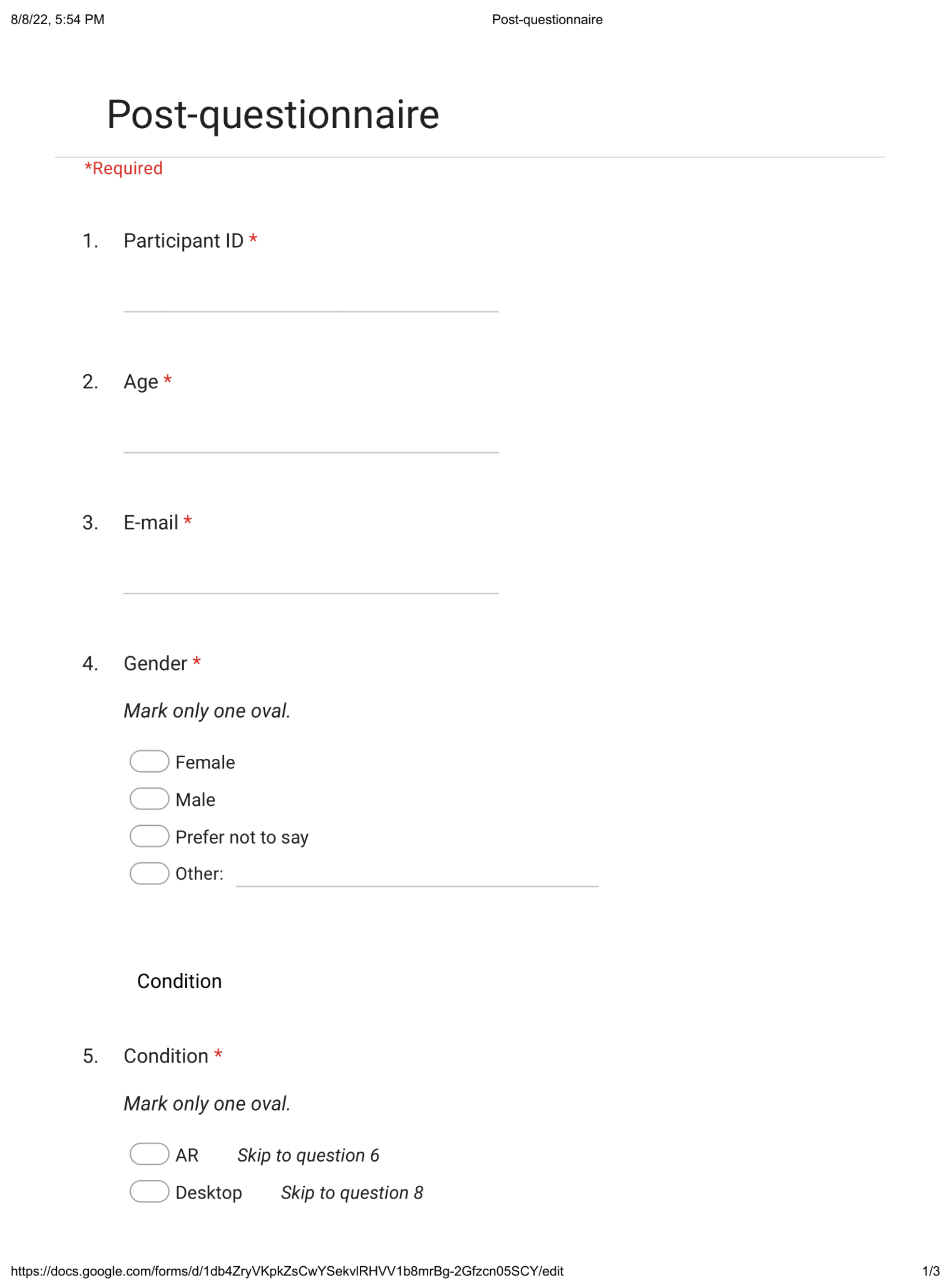
	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

19. I needed to learn a lot of things before I could get going with this system *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

G User Information Questionnaire



8/8/22, 5:54 PM Post-questionnaire

Post-questionnaire

*Required

1. Participant ID *

   ____

2. Age *

   ____

3. E-mail *

   ____

4. Gender *

   Mark only one oval.

   ○ Female
   ○ Male
   ○ Prefer not to say
   ○ Other: ______________________

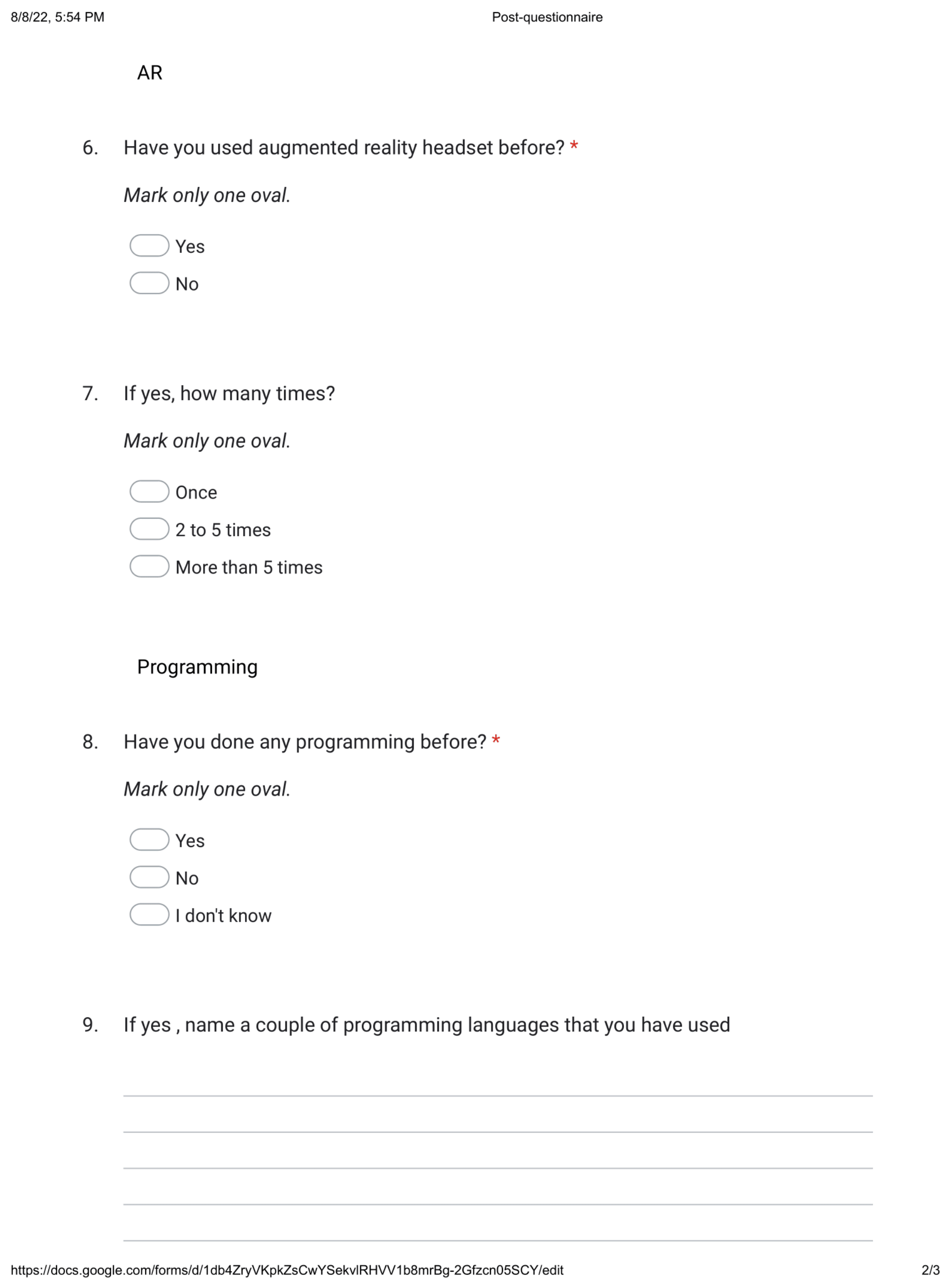
Condition

5. Condition *

   Mark only one oval.

   ○ AR *Skip to question 6*
   ○ Desktop *Skip to question 8*

https://docs.google.com/forms/d/1db4ZryVKpkZsCwYSekvlRHVV1b8mrBg-2Gfzcn05SCY/edit 1/3



8/8/22, 5:54 PM Post-questionnaire

AR

6. Have you used augmented reality headset before? *

Mark only one oval.

- ◯ Yes
- ◯ No

7. If yes, how many times?

Mark only one oval.

- ◯ Once
- ◯ 2 to 5 times
- ◯ More than 5 times

Programming

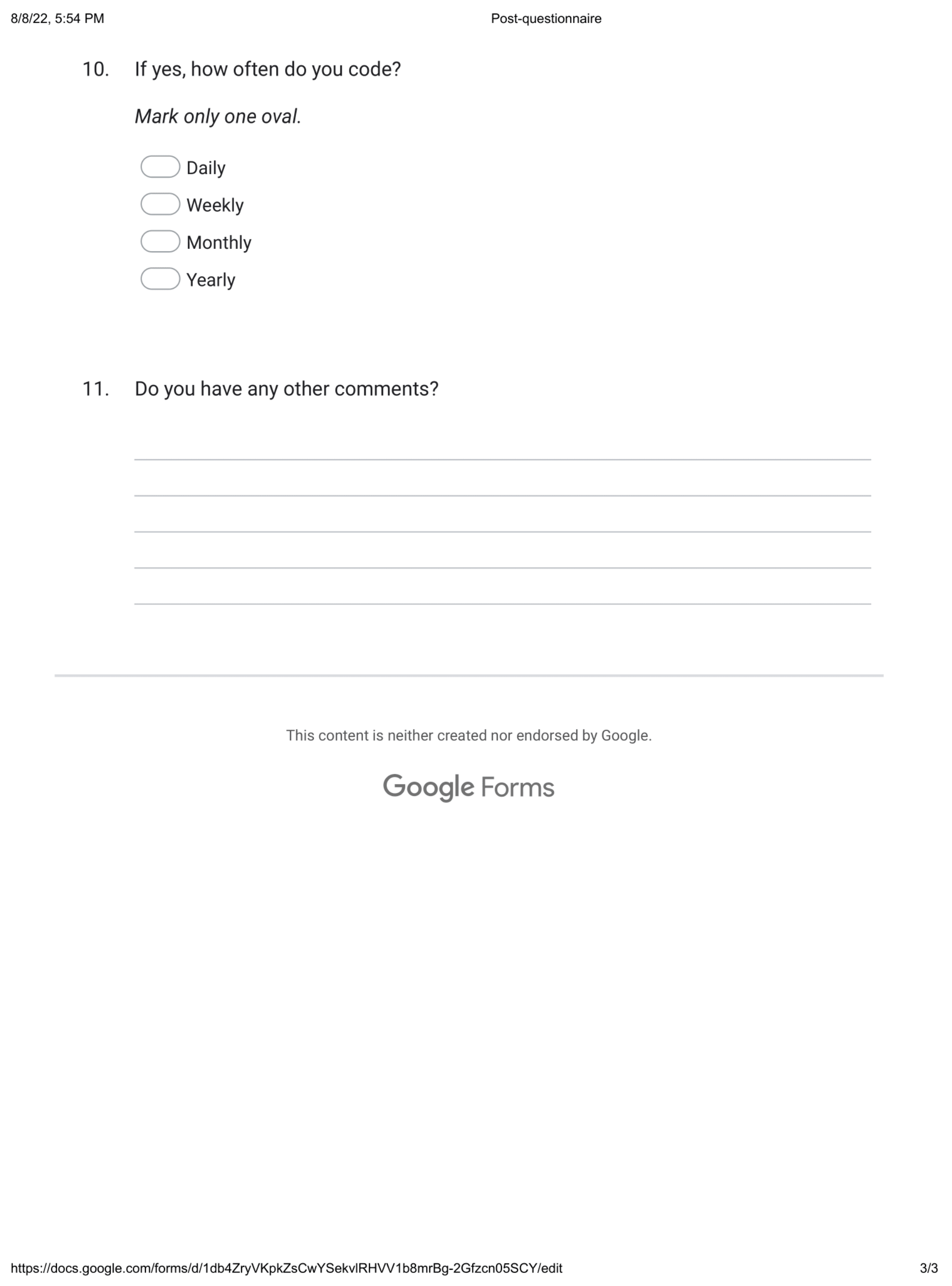
8. Have you done any programming before? *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ I don't know

9. If yes , name a couple of programming languages that you have used

https://docs.google.com/forms/d/1db4ZryVKpkZsCwYSekvlRHVV1b8mrBg-2Gfzcn05SCY/edit 2/3



10. If yes, how often do you code?

Mark only one oval.

- ◯ Daily
- ◯ Weekly
- ◯ Monthly
- ◯ Yearly

11. Do you have any other comments?