MASTER DEGREE IN COMPUTER ENGINEERING - DATA SCIENCE

Design of an AI-based Game for prevention of Dyscalculia

Ivana Orefice

Matricola 266782

Thesis Supervisor

Chiar.ma Prof.ssa Gabriella Olmo

Thesis Co-Supervisor

Chiar.mo Prof. Vito De Feo

Academic Year 2021-2022

Contents

Abstract

Mathematical abilities are extremely important. Among the mathematical disorders, there is Dyscalculia, a Specific Learning Disorder. Dyscalculic children struggle in understanding numerical concepts. It might be possible to train individuals who have difficulties in understanding maths and numerical concepts, during kindergarten or the very first years of elementary school, to try to enhance mathematical skills. In this project, we designed a new tool in this sense with the aim of helping children at risk of a diagnosis of dyscalculia. We propose an AI-based videogame to train kids and try to prevent a future diagnosis of dyscalculia, reducing the possible effects that this disorder could have. This project is based on a piece of research conducted by Manuela Piazza, Vito De Feo, Stefano Panzeri, and Stanislas Dehaene entitled "Learning to focus on number", according to which the dyscalculic individuals, when asked to perform a counting task, struggle to focus only on numerical information, being influenced by the so-called non-numerical characteristics, such as the size of the item to count. In other words, the non-numerical dimensions could interfere and distract when making a judgment that should only be taken depending on the numerical dimension. With age and education, the individual becomes more and more accurate in counting, learning how to filter the numerical information, the relevant one, from all the other, irrelevant, ones. Our project takes inspiration from this research paper, with the aim of designing a game app that should train kids on how to filter the irrelevant non-numerical dimensions during a counting task, helping them in making judgments based only on the number of items to count. The game is assisted by an AI software, that must guide the training and evolve with the improvement of the kid. The game has been designed as much attractive and understandable as possible, given that it will be played by preschooler kids. We hope that the training will help in the prevention and, maybe, in the treatment of Dyscalculia.

Chapter 1

Introduction

Mathematical abilities are, nowadays, extremely important, in many aspects of life, not just in school or workplace, but also in everyday living. It is very important to be able to *identify* if there exists some math ad arithmetic problems as soon as possible, as it is easier to act on those difficulties during childhood, to ensure a **better quality in learning** throughout life in school, and also ensure a better usage of those mathematical abilities in everyday life.

Humans, as well as many other animal species, are born with what is called **Number Sense**, which has been defined by Kesler et al. [1] as a *non-verbal skill that involves spatial relationships between numbers along a mental number line allowing for number comparison and processing.* It is important to understand and define the concept of number sense because it is thought that individuals struggling with mathematical abilities are believed to have a core deficit in number sense. It is by, somehow, testing the Number Sense that scientists could understand if an individual have mathematical disabilities or not. The way scientists could test the Number Sense are several, like:

- tests of numerical distance, i.e. understand how much distance there exists between two different numbers.
- calculation and counting tests, i.e. addition and subtraction.
- number comparison tests.

After those tests, as well as many others, and after having understood if there are mathematical disabilities or not, it is possible to **plan a training program** to intervene on that difficulties and improve mathematical skills.

The proposed project focuses on a specific kind of mathematical disorder, which is **Dyscalculia**. Dyscalculia, also known as Developmental Dyscalculia, is one of the possible forms of SLD (Specific Learning Disorders), alongside, for example, the best known Dyslexia, which affects the ability to acquire school-level arithmetic skills [2], preventing individuals that suffers from such a disease from learning math as other individuals, resulting in a severe impairment in the skills acquisition, and, of course, in a deficit in the entire learning process, from kindergarten to primary school to high school. Dyscalculia can be diagnosed only when primary school begins; this means that, before primary school, kids can never be defined as dyscalculic, but can be considered "at risk of a Dyscalculia diagnosis".

It may be possible to train the brain of the kids, ideally during kindergarten or maybe the very first years of primary school, to prevent a diagnosis of Dyscalculia or to reduce its symptoms. The presented project with this thesis is **a possible intervention tool that is aimed at those children that encounters difficulties in numerical tasks**, with the aim of training their brain and improving the mathematical skills of children as much as possible to allow them to face the world of numbers with a stronger attitude and with much more motivation.

To summarize what is inside this thesis, the first goal has been to deeply study Dyscalculia, and create a game application, made of a client and a server, that is thought to be played by kids that attend the last years of kindergarten (3-5 years old kids), with the aim of training kids and help them improve their mathematical abilities as much as possible.

Dyscalculia will be extensively treated in the next chapter, exploring all the literature to date alongside many possible solutions, but it is worth noticing right now that the decision of implementing a game-like application is not a case: there are various studies that report having success with their "training" games, as it is a way for kids to be entertained and, at the same time, it is proved that they improve their mathematical abilities, or their abilities in general. There are especially two interesting cases that will be mentioned in the dedicated chapter, that are **Rescue Calcularis** [3] and **The Number Race** [4][5]. This last one, particularly, is extremely famous as it has been the first documented game with proved usefulness, and also because it has been available for any platform (Windows, Mac) in many languages.

The project was inspired by a study conducted by Manuela Piazza, Vito De Feo, Stefano Panzeri, and Stanislas Dehaene. The paper, entitled **Learning to focus on number** [6], proposes that, when a dyscalculic individual is asked to answer questions involving numerical information, he or she has difficulty focusing only on it, allowing himself or herself to be influenced by possible non-numerical information, such as size or color. The research, to which an entire chapter of this thesis will be dedicated, therefore proposes a training method that should eliminate the so-called **interference effect**, training the ability of the dyscalculic individual to focus only on the number, filtering out all other possible distracting information.

Finally, one chapter will report out the project itself, how it is built and especially it will focus on the server-side, including all the used technologies and how far the project has gone up until now. The same chapter will also tell about the effort made to make the entire project consistent with the literature and the parameters normally used in literature, and also made it work with the Unity Environment, which is the one used to implement the client-side of the project.

The thesis ends with an explanation about the possible evolution of the game and the project.

Chapter 2

Dyscalculia

As said in the introduction, the game app that has been implemented having in mind the possibility to train children with mathematical or counting difficulties, given that to intervene during childhood should bring the highest possible level of benefit through school and everyday life. But before talking about the game itself, it is good to introduce and discover **the world of number**, how our mind is able to represent number concepts and what are the main difficulties in representing number concepts; later, a discussion about what is dyscalculia is made, talking also about how does it affect children in their lives and if there exists some way to train it.

2.1 Our Mind and The Numbers

Humans, as well as other animals (like fish and non-human primates) are born with an innate ability of representing and compare numerical magnitudes of sets of objects [7], revealing somehow that humans possess the so-called Number Sense. Some studies, like the one by Khanum et al. [7], have also proposed the idea that this ability relies on an Ancient Cognitive System, known as **Approximate Number System** or, shortly, **ANS**. But the difference between humans and many other species is that our species receives also a *formal education*, with the goal of enhancing and elevating the ANS, which is referred to having an approximate idea of numbers and number magnitude in our mind, to a much more accurate and detailed representation. It is believed that training the ANS, both within a laboratory environment or also outside (like in classrooms or schools in general), allow an improvement of basic symbolic arithmetic not only in children but also in adults. These findings belong to the study by Khanum et al., already cited, but this is not the only study on training the ANS that had success. Another study, by Honorè and Noel, [8], cites how the ANS must be considered as the root of numerical and arithmetic skills, how it is shared with other animals species and finally how it is connected with the concept of **Number Sense**: the fact that human beings are born with ANS leads to the idea that there is an innate sense of the numerical concept, alongside number magnitude and number comparison that is born with us, a numerical concept that is simply called Number Sense.

A possible formal definition of Number Sense can be the one given by Kesler et al., [1], which states the following: "Number Sense is defined as a **non-verbal skill** that involves spatial relationships among a **mental number line** allowing for number comparison and processing". In other words, it refers to *a mental representation of the concept of number inside humans' mind*, which leads to the possibility to perform tasks like number comparison and processing. Since Number Sense is believed to be biologically determined, it is possible to affirm that **Number Sense and ANS concepts are strictly correlated** and are fundamental in order to evaluate whether an individual has any kind of numerical struggle: there exists many training tasks to be submitted to kids, in order to understand whether they have some mathematical disability (or not), the severity of that disability and if that disability can be labelled as dyscalculia or not.

Several studies propose several methods and strategies to train the ANS and the Number Sense. A first possibility could be to practise with a game-like, online program which proposed exercises like quantity comparison, mental calculation and sorting tasks ([1]). An interesting study, entitled **Influence of initial mathematical competencies on the effectiveness of a classroom-based intervention**, [9], proposes the so-called **IDR**, which is the acronym for **Integrated Dynamic Representation**, a computer-based tool which aims to test the formal and informal mathematical knowledge. The difference between **formal and informal knowledge** is that, while in the first case it is possible to acquire knowledge *informally, like through the usage of a game or through intuition or even through daily experience*, in the latter case *the knowledge is built through formal education and dedicated exercises and tasks*, such as practising with addition or subtraction. This study is particularly interesting, as it uses IDR to enhance not only mathematics competencies, but also mathematically based word-problem solving abilities. The numerical training is done in three different forms:

- **Iconic Representation**, i.e. the information is presented through the usage of images only
- **Symbolic Representation**, i.e. the information is given through words
- **Mixed Representation**

The tasks are counting, quantity comparison, informal/formal calculation, informal/formal concepts and number facts. Informal/formal calculation means that the mathematical operations are presented twice, firstly informally, later formally. Being the IDR an adaptive tool, people with the highest level of difficulty recorded more consistent improvements, compared to people with less difficulty, who still improved after training. Furthermore, this paper also introduces the concept of **MLD**, which stands for **Mathematical Learning Disorder** as a formal concept, and also classify the severity of the mathematical difficulty as low, medium or high. An important concept that is introduced within this study is that **people with ADHD (Attention Deficit/Hyperactivity Disorder) are the ones who have the most difficulty with numerical**

concepts. In other words, whether an individual suffers from ADHD, then it will be more likely that the same individual has math difficulties or that their difficulties are worsened by the presence of ADHD (ADHD-MLD comorbidity). This is a very interesting finding, that has been cited also by Gonzales-Castro et al., [10], which states not only the heterogeneity of students with MLD, not only that students with MLD continue to experience math difficulties even in the higher grades, such as high school, but also confirms that those difficulties are exacerbated by the presence of ADHD: indeed, students with both MLD and ADHD manifest a set of cognitive problems that must be summed up, resulting in an high risk of failure with maths. Moreover, it states that, considering people with MLD, only 3% to 13% of them experiences serious specific maths disabilities, whereas, in case of MLD-ADHD comorbidity, this percentage rises significantly: between 11% and 26%.

Last concept to introduce is the one of the **Working Memory (WM)**. According to the definition provided by Passolunghi et al, [11], *"Working memory (WM) refers to a mental workspace, which enables a person to hold information in mind while simultaneously performing other complex cognitive tasks"*. In other words, WM could be imagined as **a container in which information resides** as long as it is necessary for the completion of a more complex operation (which can be a mathematical operation, but also of another nature), and it can be considered as a key domain-general predictor of mathematical competence. Moreover, it is bounded both to early and to later numeracy skills, which means that all the numerical calculations, even the simplest ones, require the intervention of the Working Memory. The importance of WM is emphasized by the fact that WM influences the mathematical achievements and that poor WM ability leads to poor mathematical performances. The interesting thing to notice is that, to date, **there isn't a specific training that enhances WM ability only**, but, instead, since the WM is a precursor of early numeracy skills and achievements, by training WM, a so-called **trans-**

fer effect will be produced on early numeracy. In other words, with the WM training, the kid improves not only on WM ability but also on early numeracy, since there will be a transfer effect of knowledge.

The concepts of WM are also cited by another study, [12]: the definition of WM provided by this study is consistent with the one provided by the previous study. A new point of view proposed by this study is that WM is organised as a series of subsystems, alongside a core system, also known as *central executive*: there exists one subsystem to visualize the information (also known as visuospatial sketch pad), two subsystems responsible of storing information and one is responsible for verbal and language-related content. Among the subsystems of WM, the **visuospatial subsystem** is believed to be the best predictor for arithmetic problem solving, followed by the central executive. The study's goal was to perform two different kind of computer-based training tasks (*number-line and magnitude comparison* to train the number sense and a so-called *n-back updating task, a Corsi block task and a letter span task* to train the visuospatial WM) and investigate if the two trainings led to improvements in mathematical skills and if, possibly, one of the two trainings had led to higher improvements than the other training. Even if the gains in the spatial WM training were not as high as the ones in Number Sense, the results suggests that **either training of number sense or WM training may enhance math ability**, which leads to confirm the concept of *transfer effect* previously mentioned.

This study by Gonzales-Castro et al., [10], cited before, paves the way to a specific discussion about dyscalculia: given that dyscalculia is one of the existing mathematical disorders, it is inserted among them, but it is necessary to make a separate discussion given the interest of the scientific community towards this disorder and also the fact that more and more individuals are sufferring from it.

2.2 What is Dyscalculia?

The previous paragraph can be considered as an introduction to try to understand how big the world of number is. Now is the time to turn the spotlight on dyscalculia only. A detailed and long discussion about MLD has been addressed: when referring to *Mathematical Learning Disorder*, it is all about a very wide range of disorders of a mathematical nature, which have been also classified as mild, medium and high,[9], depending on the difficulties that characterize the individual, which means that we can classify the level of difficulty of the single individual to prepare a plan of intervention and training, which best fits the needs and the level of competencies and difficulties of the kid.

Turning the page to the Dyscalculia, as said Dyscalculia, also known as Developmental Dyscalculia, or DD, is one of the possible mathematical disorders. Various definitions of "Dyscalculia" have been proposed and although there is no universal one, it is convenient to report one provided by the **DSM-IV (Diagnostic and Statistical Manual of Mental Disorders)**, which suggests that *a diagnosis of Developmental Dyscalculia (DD) is suggested when "mathematical ability, as measured by individually administered standardized tests, is substantially below that expected given the person's chronological age, measured intelligence, and age-appropriate education"* [2]. Many studies have been conducted on school-based and general population, not only in Western countries, highlighting a possible estimation of people suffering from DD ranging from 3% to 6%. This estimation suggests that dyscalculic individuals may suffer from it because of other specific learning pathologies or SLDs, such as Dyslexia, which creates problems in the case of reading. Another possible definition is the one given by Kucian et al., [3], which states that *Developmental Dyscalculia (DD) is a specific learning disability that affects the acquisition of mathematical skills in children with normal intelligence and age-appropriate* *school education, presumed to be due to impairments in brain functions.* A third and final definition is the one given by Ladislav Kosc: **"Developmental dyscalculia is a structural disorder of mathematical abilities which has its origin in a genetic or congenital disorder of those parts of the brain that are the direct anatomico-physiological substrate of the maturation of mathematical abilities adequate to age, without a simultaneous disorder of general mental functions"**. By analyzing this last definition, it is quite clear that Kosc is making a inherent distinction between:

- **Primary Dyscalculia**: this category includes those mathematical deficits which are the result of a reduced ability when it comes to acquiring mathematical skills.
- **Secondary Dyscalculia** (or pseudo-dycalculia): an individual suffers from pseudo-dyscalculia if its math deficits are influenced by external factors.

This categorization is key to remark the distinction between DD and MLD, and this is also shown by the increasing body of research on DD and MLD growing this last years, identifying in DD *an endogenous learning disorder* whereas considering MLD as *driven by exogenous factors or cognitive deficits not specific to numerical processing, such as working-memory, visual-spatial processing or attention.* The question to be answered now is: **What deficits does a dyscalculic child exhibit?** Answering this question is of fundamental importance as fully understanding the difficulties helps you to think about the way to intervene on those difficulties. To answer this question, it is useful to analyze the behavioral and brain-lever characteristics of primary DD, also making a clear distinction between primary and secondary DD.

2.2.1 Symptoms of Dyscalculia

Many researches have reported their own symptoms of DD. Here, we give a first glimpse of what are the possible symptoms of DD:

- Difficulty in understanding or remembering concepts related to mathematical operations, like multiplication, addition, subtraction, division, fractions, carrying or borrowing, [13].
- Problems in understanding the mapping between numbers when written as a **word** (such as the word "two") and their **corresponding Arabic cipher** (the Arabic cipher '2'), [13].
- Trouble explaining math processes, [13].
- Trouble when asked to complete a mathematical task and trouble when asked to show the steps that involve the completion of a mathematical task, [13].

Another point of view is provided by Navagalli et al., [14], which provides another list of possible sympthoms that DD patients may manifest:

- General difficulties when working with numbers and confusion when using math symbols, not just Arabic ciphers but also plus (+), minus (-) and so on.
- DD patients ofte reverse or transpose numbers (for instance, 29 will become 92).
- Problems with telling and time and computing math mentally.
- Difficulty with estimation and approximation.
- Slowness in given answers to math questions.

All of those symptoms clearly reflect on the mathematical and general performance of the DD patient, influencing his/her academic path and resulting into various types of DD, which will be mentioned in the following subsections.

2.2.2 Behavioral Characteristics

In recent years, the idea according to which DD could originate in a deficit in the neurological system that processes numerical quantities, making the elaboration of complex arithmetic facts was born. Of course, this influences the learning process of math and arithmetic facts, at various stages of the growth, but it also poses a doubt to the scientific community: is it possible that the fundamental and basic skills are not present? It is not just a matter of arithmetic facts: DD patients could show difficulties in processing numerical magnitude information. This may suggest a lack of automaticity in processing numerical information, which actually is not a universal indicator whether the underlying semantic representation is impaired, or whether there is a deficit in the link between the semantic representations and their symbolic referents, but still it is very much used by the scientific community as a method to investigate on a possible presence of DD in a patient.

2.2.3 Non-Numerical Deficits

DD is a specific learning disorder that affects *math skills*, but still it has to do with domain-general cognitive mechanisms such as working memory, visual-spatial processing, or attention, as highlighted in the previous paragraph. This motivates why, when observing a DD patient, it is possible to witness other kinds of difficulties, which must in any case depend on deficits in WM, for example. The role of visuo-spatial processing is key for arithmetic processing: despite that, the literature is not consistent. Indeed, depending on the used criteria, some studies classify primary DD as completely independent from WM impairments, meaning that it is secondary DD patients that may manifest problems related to WM and simila. Finally, an important thing is that some studies report that the root cause of DD must be researched into a **disruption of the mapping between Arabic digits and their numerical magnitudes**, meaning that the DD patients may encount difficulties right in the moment when he/she sees Arabic digits, because in his/her brain the mapping between the ciphers and the concept of the number is completely disrupted.

2.2.4 Neural Characteristics

A detailed analysis of how the brain reacts to number facts in general has been conducted by Kaufmann et al., [15], which collects and arranges various studies on brain imaging precisely in the case of arithmetic facts, focusing its attention on the brains of children. It must be underlined that only 19 researches have been included into this Meta-Analysis: it would be good, in the future, to be able to expand this number. According to this Meta-Analysis, there are three distinct but functionally interconnected neural networks that support different components of number processing and computation:

- the horizontal segment of the intraparietal sulcus, in charge of mental representations of the magnitude of number (also referred to as Number Sense).
- the left angular gyrus (which is situated in the inferior parietal lobe and closely neighbors language-relevant regions), responsible for encoding and retrieval of arithmetic facts.
- the occipital brain regions and its integrity (including the so-called visual word-form-area, known also as VWFA), in charge of processing of written calculation.

Results of this study are four-fold:

- when aiming to identify brain regions that support the comparison of symbolic and non-symbolic numerical magnitude in control children, it was shown that in children symbolic and non-symbolic processing produces right parietal fMRI signals, while the left parietal lobe appears to respond mainly to the symbolic elaboration of the number.

- when aiming to study age-dependent activation differences between children and competent adults with regards to comparing non-symbolic numbers, it was shown that age modulates the exact localization of parietal activations, thus children activate multiple intraparietal anterior regions, while the intraparietal activations of adults are more posterior and stronger in the right hemisphere.
- when the goal was to identify brain regions that are activated in a coherent but different way from children with and without dyscalculia in the numerical comparison task, the fronto-parietal fMRI responses of children with and without dyscalculia are quite distinguishable because, compared to Dyscalculics, control patients produced stronger activations in the left precuneus, right inferior parietal lobe and right DLPC.
- finally, the ultimate goal was to identify neural correlates that are consistently recruited by typically developing children when solving simple calculations. In this case, it was shown that the children produced consistent and extensive activations in the lower and upper parietal cortices (including bilaterally IPS and right SMG).

2.2.5 Types of Dyscalculia

In the previous sections, the topic "dyscalculia" has been explored in various directions, analyzing what may be the symptoms of dyscalculia, from a behavioral but also strictly numerical point of view, highlighting the aspect of non-numerical deficits and the study of the areas of the brain involved in processing numerical information. However, before moving on to the possible numerical trainings available in the literature, it should be emphasized that, although a classification of the various types of DD has already been provided, that is, the one that distinguishes primary DD from the secondary one, [2], a further classification method is reported, suggested by Navagalli et al., [14], and which is more closely related to the symptoms of DD.

- *Verbal DD*: interpretation of verbal math terms.
- *Operational DD*: performing basic arithmetic operations.
- *Lexical DD*: reading written math terms, symbols (related to symbolic math and Arabic digits).
- *Graphical DD*: symbol manipulation (related to symbolic math and Arabic digits).
- *Ideognostic DD*: mental calculations.
- *Practognostic DD*: pictorial representation, may be by means of images.

Having concluded the discussion about Dyscalculia, the methods of recognition and the types highlighted by the literature, it is time to analyze the possibilities.

2.3 Training Dyscalculic children

A tremendous effort has been made over the past few decades to ensure that DD patients can improve their math skills, and in fact, different ways of training numerical abilities of DD patiens have emerged. Very often, in the literature it is possible to find examples of mathematical training, done by means of number comparison or mental line representation tasks or also of addition (with and without carry) and subtraction (with and without borrow). Here are some of these possible numerical trainings and the purpose for which they were carried out.

The **first example** of numerical training that is worth noticing is the one proposed by Piazza et al., [16]: according to this research, there are no studies that have shown unequivocally that dyscalculic children are compromised in purely non-verbal and non-symbolic numerical tasks: there are, instead, studies that show that dyscalculic children have numerical difficulties when they have to compare quantities expressed through Arabic digits, but not when quantities are expressed as non-symbolic numerosities, which is why the study proposes a numerical comparison task, in which the children saw pairs of black dot arrays and were asked to decide without counting which matrix contained the greatest number of dots.

So the training in this case is a **non-symbolic numerical comparison task**, bringing as a result the fact that number acuity is severely impaired in dyscalculic children in comparison to non-dysclaculic ones. Moreover, this result fits with neuroimaging studies, showing that regions of the parietal cortex, close to the locus of specialised neural systems for numerical quantity are structurally or functionally impaired in dyscalculic subjects, confirming a **severe impairment in DD patients at the brain level**, suggesting also that Dyscalculic children, when facing with arithmetic problems, implement compensatory strategies using auxiliary systems.

Another interesting point of view is offered by Navagalli et al, [14]. This study introduces **the possibility of using computerized tools to submit a mathematical training**, reporting simple tools, such as a simple **Electronic Math Worksheet**, as well as more complex tools such as the **The Number Race** software, which it will be the subject of a detailed description later on. A similar study is the one by Ansari et al, [2], which, again, cites The Number Race as a possible strategy to train dyscalculia. The principal goal of The Number Race is to propose tasks like **placement of numbers on to a number line**: in other words, it is the game's goal to improve the precision of numerical magnitude representations in DD subjects.

Finally one last interesting study, the only one where the subject were adults, was the one by Castaldi et al., [17]. Thanks to this study, it is possible to introduce an important concept that will come back in the next chapter: the **interference between a numerical and a non-numerical dimension**. Indeed, the first aim of the study was to determine the ability of able-bodied adults (non-dyscalculic subjects) to discriminate between the judgement of numerosity versus the judgement of another quantitative dimension (which, in case of this experiment, was the average item size), evaluating the interference effect between the numerical and the non-numerical dimension, thanks to a set of stimuli made of heterogeneous arrays of dots, half black and half white, which were generated varying orthogonally in means size and numerosity. When asked to compare and to guess which one of the two presented array of dots was the most numerous, the adult should be able to discriminate between the numerosity (numerical dimension) and the size of the dots (non-numerical dimension) in order to focus only on numerosity and guess it right. The second objective of the study was to subject adult DD patients to an enormous amount of tests, in order to evaluate their performance on tasks of average size of the object, discrimination of numerosity, trying to find a confirmation to the **hypothesis according to which dyscalculia is associated, or not, with a general deficit of inhibitory control.** The results of this study can be summed as follows: although participants without mathematical problems were able to compare the numerosity without noticeable interference from the non-numerical dimension, they tended to *overestimate* the average size when presented with a high number of dots and tended to underestimate it when presented with a small number. This pattern of results suggests that the interference pattern is not affected by different attention allocation or visuospatial memory load, at least as far as differences in presentation modes are concerned. In contrast to the control subjects, and as could be expected, the dyscalculic group was strongly affected by the congruence of the information of irrelevant (non-numerical) size during the judgment of average size, although during the judgment of numerosity the groups were both influenced in the same way by the number of points in the arrays. This states that the **interference effect** is something that **has to do with both dyscalculic and non-dyscalculic subjects**, even if control subjects are able to not be influenced as much as dyscalculic subjects.

2.4 How to use the knowledge gained until now?

This careful and deep analysis of Dyscalculia, its sympthomps and types and the possibilities on how to train it leads to the core of the project that is object of this thesis. Particularly, it paves the way to the inspiration paper of the game-app itself that this theses wants to tell about, which will be reviewed in the following chapter.

Chapter 3

Learning To Focus On Numbers

To better understand the rationale behind the game-app that will be presented in the following, it is wise to firstly introduce the paper that has inspired the game.

3.1 Premises

This study was concentrated on kids, particularly on their **ability to make numerical judgement**. In the previous chapter, it was highlighted how humans are able to discriminate since the infant age, even if the precision is very low: indeed, newborns are able to discriminate only when the presented sets differ by 300%. This precision in discrimination grows with the kid, until they reach adulthood and become able to discriminate even when a small change happens (i.e. when two sets differ by just 15%-20%). A key role is played by the formal, scholastic education, which speeds up improvement in the ability to discriminate. This observation related to the way in which the human kid improves in performing **discrimination tasks** while growing leads to an important concept, which is the one of the *Optimal Decision Boundary.* A decision boundary can be defined as a straight line, which divides the Cartesian plane into two parts. Each of the two parts of the plane represents a class. The **optimal** one is so called because it **manages to perfectly separate**

the two classes, so it represents the best solution to a classification problem. The spatial characteristics of the Optimal Decision Boundary determine the level of competence of the tested subject: if it is perfectly vertical, then it is the best possible line.

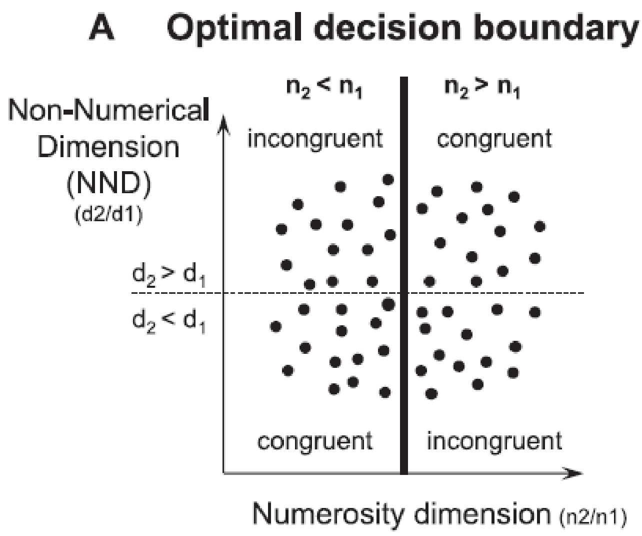

Figure 3.1: Optimal Decision Boundary: each dot represents a single trial. The **vertical line** represents the Optimal Decision Boundary for a comparison task. For each dot, the closer it is to the boundary, the harder is to correctly classify that dot/trial.

Figure 3.1 offers a graphical representation of an Optimal Decision Boundary, which separates the left class, in which the number of dots shown in the second area is lower than the number of dots shown in the first area, from the right class, in which the viceversa happens.

By looking at this Figure, which represents the feature space, it is also possible to highlight the division of the plan into **4 quadrants**: in two of these quadrants the trials are congruent, in the other two, however, the trials are incongruent. The **congruency and incongruency** of the trials refer to the relationship between the numerical variable and the non-numerical ones: when we have congruency between the number of dots and the dots size, it means that the larger is the size of the dots, the greater will be the number of dots. The concept of incongruency is exactly the opposite, meaning that to a larger size of the dots corresponds to a smaller set of dots (and viceversa).

3.2 The Sharpening and Filtering Hypothesis

The **Sharpening hypothesis** can be considered as a possibility, the most straightforward one, to explain the improvement introduced in the last paragraph: it affirms that **"maturation and formal education progressively sharpen the internal representation of numerosity"**. This means that the refinement of the ability to discriminate depends both on factors related to brain growth and development, but also on the formal education, and this affects how the number is represented in the minds of children. The Sharpening hypothesis, however, may not be the only one when it comes to improvements in the ability to discriminate the numerosity. In fact, it is possible that **the improvements are due to an enhancement in the way an individual focuses the attention exclusively on the representation of the numerosity**, filtering out all the distracting information that has nothing to do with the numerical dimension (which is the relevant one) but, indeed, they have a non-numerical nature. This would also be confirmed by the fact that, already during the first years of infancy, children progressively improve in separating the so-called *"numerical variables"*, thus the ones giving information about the numerosity, from the so-called *"non-numerical variables"*, dealing what in the previous paragraphs was called as "interference effect" and producing more and more accurate judgments. This new hypothesis is known under the name of **Filtering hypothesis**. In other words, the Filtering hypothesis proposes the idea according to which the developmental improvements in numerical judgments includes also the possibility to distinguish between the numerical and the non-numerical information, and making judgement should

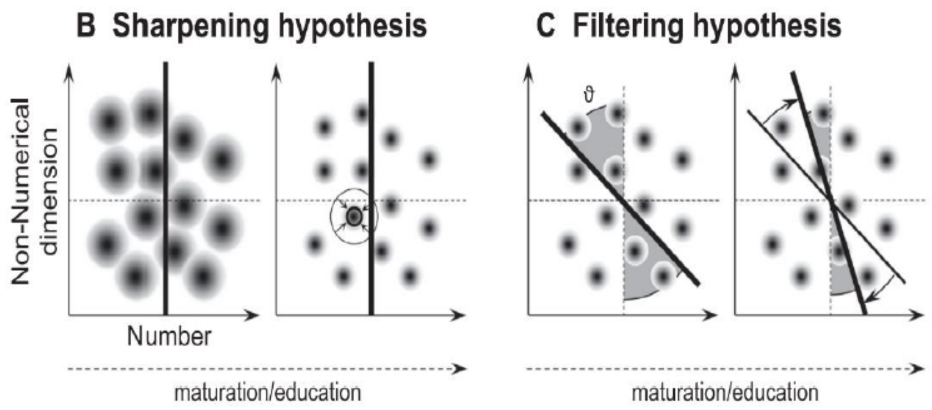

Figure 3.2: Sharpening & Filtering Hypotheses - Evolution with time and education. The width of the dots represents, in Panel B, the **noise of internal representation of number and of another non-numerical dimension**, also called NND. The Sharpening model shows that representations are highly noisy and, becoming more precise (sharper) with age and education. The Filtering model shows the development as a progressive rotation of the decision boundary towards the optimal, vertical line, with a consequent reduction of the angle between the **actual decision slope and the optimal one**.

only depend on the numerical information.

To summarize, while the Sharpening hypothesis has to do with the **precision of the representation** that improves with age and education, the Filtering one affects the **effectiveness of the decision system at discarding task-irrelevant representation**, focusing the attention only on the relevant dimension, the numerical one.

An important thing to point out is that those two hypothesis are not mutually exclusive: in other words **both of them are valid and will presumably occur during the learning process**, but it is useful to report an analysis to understand what could be the effects if only one the two cited hypothesis is tuned.

- If sharpening is the only mechanism, then there should be an **overall reduction in error rates**, meaning that the reduction happens for both **congruent and incongruent** trials.
- If only filtering is at work, on the other hand, **learning should differen-**

tially affect the congruent and incongruent trials: progress should be mostly observed on incongruent trials, but it should be absent on congruent trials. It is also possible that the kid performs worse on congruent trials.

As already said, the most probable possibility is that both Sharpening and Filtering hypothesis occur together during the development of the kid mathematical abilities: this will result in improvements in both congruent and incongruent condition, but mostly on the incongruent ones. However, to find out which of the two hypotheses is really the determining one, especially as regards dyscalculia, an experiment has been conducted.

3.3 The Experiment

Before describing the experiment, it is useful to introduce the concept of **trial**, which can be defined as a single run of the experiment, including all the parameters that characterize the entire run. The experiment conducted by the team of researchers consists of a **numerosity comparison (or number discrimination) task**: each trial is made up of a pair of arrays composed of black dots, shown inside two white disks, one on the right of the screen and one on the left. The participant was requested to identify the area within which there was the greatest number of dots before the timer elapsed: if the response arrived after the timer elapsed, it was considered as an incorrect answer. The participant was asked to answer as quickly as possible, without trying to count the number of dots but trying to guess the most numerous set. The difficulty of this type of experiment lies in the fact that, although it seems a simple task, not only the number of dots within the areas was altered, but also their size was changed from trial to trial, which leads to the possibility to have congruent trials (when the number of dots and the size of the dots are both large or small) or incongruent ones (when the number of dots is large and the size is

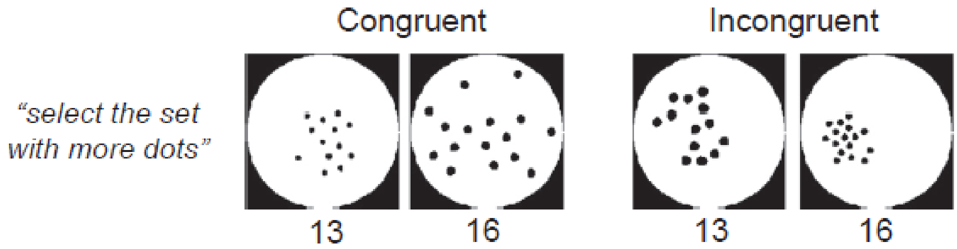

Figure 3.3: Example of Congruent and Incongruent Trials

small or vice versa).

The many participants were chosen in such a way that the social, cultural, educational and age background was as different as possible: in fact, 44 Italian kindergarteners (aged between 3.6 and 6.2 years), 29 Italian school-aged kids (aged between 8 and 12 years), 20 educated Italian adults (aged between 22 and 33 years), 25 Italian children with dyscalculia (aged between 8 and 12 years) and 38 Munducurù kids and adults (aged between 3 and 63 years) participated to the experiment.

The variables that guide the experiment and define the stimuli space are chosen in order to be consistent with the previous literature:

- **Item Surface Area** or, shortly, ISA: indicates the area occupied by a single dot.
- **Total Surface Area** or, shortly, TSA: indicates the item surface areas multiplied by the number of items.
- **Field Area** or, shortly, FA, also known as "convex hull": indicates the portion of the space where dots actually fall into.
- **Sparsity**: indicates the field area divided by the number of items.

The goal of the experiment was to analyze the **interference effect between the numerical and non-numerical dimension** (not a specific non-numerical dimension among ISA, TSA, FA and Sparsity); this is why the team has merged the 4 variables previously mentioned into a single one, called NND, which stands for **Non-Numerical Dimension**, which has been defined as the "first Principal Component (estimated by means of Principal Component Analysis) of the set of the 4 previously defined non-numerical variables". The NND was computed so well to be able to explain the 98.9% of the variance of FA, ISA, TSA and Sparsity, and this makes it a good summary measure. Moreover, the usage of PCA technique guarantees that the NND built in this way will be orthogonal to all the other variables. To build the NND, the weight used for the PCA were 0.557 for Sparsity, 0.487 for ISA, 0.473 for TSA and 0.467 for FA: this indicates that all of them are equally loaded onto the NND.

3.4 Analysis of the Results

The first investigation was done on **accuracy**. By separating congruent trials from incongruent ones, and also separating the analysis, it emerged that the response to congruent trials was overall more accurate than incongruent ones, except for the group of dyscalculic children, **confirming what was already foreseen by the filtering hypothesis**.

Data analysis was carried out using two different approaches, known as **Logistic Regression** and **Mutual Information** (also known as Shannon Information). The Logistic Regression analysis has been conducted both on a group level and on a single-subject level, to investigate the contribution of the two parameters (i.e. the numerical one and the NND), their weights in the decision making process and what is the logratio of the numerosity and non-numerical dimensions of the two sets. The results show that all groups suffered from the interference effect from irrelevant non-numerical dimensions, in a independent way with respect to age and education (this was confirmed by analyzing the results by Munducurcù people). Moreover, it is wise to notice that the interference modulation guides in a dominant way the developmental change in performance with age, education or in dyscalculic subjects, and this, in the end, **matches the filtering hypothesis**.

The Shannon information approach provides another confirmation of the results obtained with the previous approach. While the Logistic Regression approach explains the relationship between the numerical choice and the numerical and non-numerical variable, the Shannon approach (or mutual information), being a general and assumption-free approach, investigated every possible statistical relationship, linear and non-linear, estimating again the effect of number and the NND subjects' choice. The results of this analysis confirm the previous findings: **all groups are influenced, at least partially, by the task-irrelevant non-numerical dimensions when making a numerical decision, and age and education contributes in reduction of the interference effect between the number and the NND**, becoming the dominant factor in the developmental change in performance.

To summarize, it is good to comment Figure 3.2. Panel B, dedicated to the Sharpening Hypothesis, shows that, with maturation and education, the envisioning of the features of the trial is more clear. In other words, the way humans perceive the numerical difference happens to be clearer, and this is shown by the fact that, if on the left side the dots are larger and their edges are more shaded, on the right side the edges of the dots become more defined and the dots progressively smaller. Therefore, the **numerical acuity**, or the ability to discriminate quantities, improves. Panel C is dedicated to Filtering Hypothesis. Numerical acuity, in this case, has nothing to do with this hypothesis: rather, it is observed that the **Decision Boundary changes**. Indeed, with age and education, the Decision Boundary moves, becoming more vertical, meaning that, with age and education, the ability to classify increases.

Dyscalculic subjects have difficulties in concentrating on the numerical dimension, being influenced, much more than other able-bodied subjects, with non-numerical dimensions. This statement is confirmed by the results of the previous analyzes (Logistic Regression and Shannon information), which show that the numerosity comparison task is a multi-class classification problem, distinguishing the effects of age from the effects of education when making a numerical decision, making the education as a key factor whose primary influence is to be focused on significantly reducing the effect of non-numerical features on numerical decision-making.

3.5 To Conclude: how to implement a game app?

The acquired theoretical knowledge and the results provided by this research study are the basis of the development and implementation of the game-app. The main mission of the project is to propose a **number discrimination task**, which is conceptually similar to the one proposed by the experiment, but it is presented in a more funny way, with a dedicated interface, in order to educate the children in focusing their attention only on the numerical dimension of a trial, and this must be done at an early stage of their life and school path, making it the most enjoyable experience possible.

The trials proposed to the kid will be both congruent and incongruent ones, and this depends on the child's mathematical skills: if mathematical skills are very poor, congruent trials will be the first to be proposed, because they are easier to guess. As the child improves, incongruent trials may be proposed, which are more difficult to guess.

The game-app will be integrated with an AI agent, which will have a twofold purpose: on one hand, it must be able to compute the first ever trial for a specific kid, in an automatic way and based on a space of possible combinations and on the age of the kid. This way, the beginning of the treatment with the app will be automatized and without human intervention. The other aim of the AI agent will be to collect the performances and the results by the kid who is playing the app, meaning that it will collect the correct and incorrect answers provided by a kid, alongside the variables that compose the specific trial, producing new trials that will be more accurate and that fit the results obtained by the child, somehow evolving with the kid's mathematical skills. In the end, the training is realized by reviewing the correct and incorrect answers, analyzing the trials that correspond to an incorrect answer and finally deciding what trial should be given to the child in subsequent attempts, and this is decided base on the performances on the previous tasks, and it will continue until the final goal of being able to discriminate between numerical and non-numerical dimension is reached.

In conclusion, this is the great ambition of the game: helping children in avoiding mathematical problems along their scholastic and life path, averting, perhaps, a possible diagnosis of Dyscalculia.

Chapter 4

Training by means of Apps

The implementation choice to create an app for this project must be researched not only in the fact that proposing a game to children (especially very young children) is certainly the mode that can best capture their attention, but it is also supported by several studies that show the effectiveness of these game-apps, demonstrating how to benefit from apps designed in the form of a game. A review of researches and games supporting this point of view is then reported below.

4.1 Serious Game

An interesting work conducted in the UK, [18], with the purpose of raising early achievements in mathematics, summarizes quite well the reasons why it is convenient to adopt an approach with the apps. Because those apps' principal aim is not to entertain but to train and teach something to the player, it is possible to define them as **"Serious Games" or as "Knowledge Games"**[19]. The usage of Serious Games is growing and, as a confirmation of this statement, it is enough to see the Apple Store: over 1000 applications, targeted to kindergarteners, are available in order to teach them something, especially because of the motivational effect that the usage of an app has on a kid, independently by the age. It is good to notice, anyhow, that playing too much could be harmful for a kid, and this is why the introduction of a mathematical training with the usage of ICT is still a controversial topic. But if the training is controlled and supervised, the benefits are astonishing. This is confirmed by a number of studies, including the study which is object of this paragraph: it states that *"Educational math apps delivered on touch-screen tablets offer an opportunity for individualized math practice targeted to children's needs"* and a possibility could be to integrate the training with the app with the normal teaching already provided at school, creating a sort of *blended learning approach*, to enhance and improve mathematical skills. To be effective, the app must provide an **active, engaged, meaningful and socially interactive learning, with specific goals**; moreover, it is good to notice that, not only, as said, learning with the help of touch-screen devices is very motivating, especially for young children, but also access to this type of devices is growing, both in the school and at home. Math apps' common features must include:

- explicit instruction.
- repetitive and cumulative training in mathematical concepts.
- immediate feedback.
- challenge and early reward.
- individualized, self-paced learning.

Therefore, the questions underlying this research are many, all oriented towards the development of mathematical skills: what are the *benefits of a "blended learning" approach*, that is an approach composed of both standard lessons and training with apps? What are the *benefits of a math app used instead of standard lessons*? The comparison is done thanks to a control group that has only received standard math lessons. Two existing apps were used for this study, which target **factual knowledge and basic conceptual understanding**, for example, simple numerical operations (addition and subtraction).

The results demonstrate the effectiveness of apps proposed on a touch-screen device, both alone and in combination with standard math lessons, for children aged 4 to 5 attending the first year of school. Thus, as suggested by the paper itself, *"the apps are a form of quality math instruction and suggests an increased focus on early math at the class level, can take the form of efficient instructional practices without the need for extra time implied in learning math"*.

Another important implication of this research is that it is the first study to adopt a **rigorous and scientific methodology** in investigating mathematical skills and how they can be improved with touch-screen technologies. This makes the results obtained even more important.

4.2 Rescue Calcularis

The first example that will be explained in the following is about a game-app, named **Rescue Calcularis**[3]. The game-app has been used for the training phase, having a twofold goal:

- improve the **spatial representation** of numbers among the mental number line.
- improve the **association between the representation of numbers and space**, alongside the comprehension of the ordinality of numbers, estimation and arithmetical skills.

Taking a small step back, the research focuses its attention on DD, theorising that one of all the possible deficits observed in children with DD is certainly related to the **representation of numbers and quantities**. Our mind represents and organizes numbers exploiting a mental representation, known as **mental number line**, which is can be imagined as a line along which the numerical progression is built. The mental number line belongs to that set of basic skills, such as Number Sense, which the human being develops and possesses already a few hours after birth. With admission to school and education, the representation of the mental number line changes: **if at first the representation is logarithmic, it will later become linear**. This shift from logarithmic to linear representation, however, is not observed in dyscalculic children: 10-year-old dyscalculic kids are observed to perform at the level of 5-year-old normally achieving children. For this reason, it is necessary to design and implement targeted training for dyscalculic children to enhance their mental numerical representation; this, presumably, will bring benefits in learning mathematical skills in general.

Rescue Calcularis can be considered as a tool to train and enhance the mental number line representation: it is thought to be played for 15 minutes a day, 5 days a week for 5 weeks. There's also a story behind the game, thought to make the game more attractive and the game experience more funny and enjoyable: the player had the mission to rescue a planet, called "Calcularis", indeed, and to do so, the player should be able to overcome various levels with increasing difficulties. The levels will display some tasks like comparison tasks or addition/subtraction tasks.

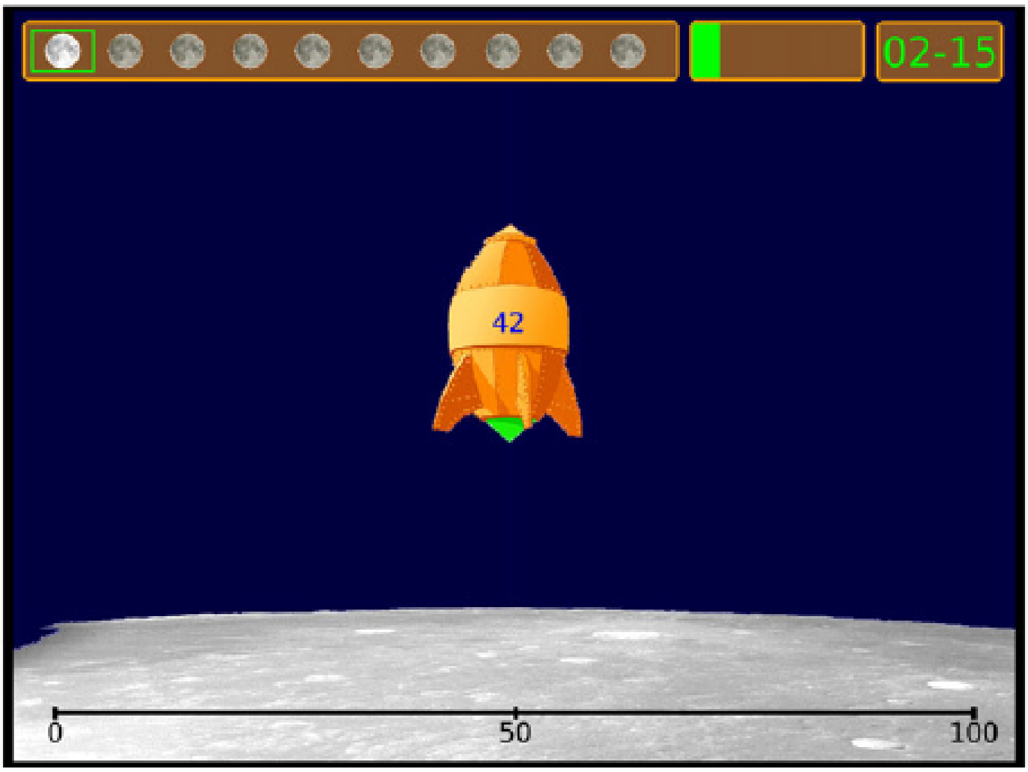

Figure 4.1: Example of Number Placement on a Numerical Line

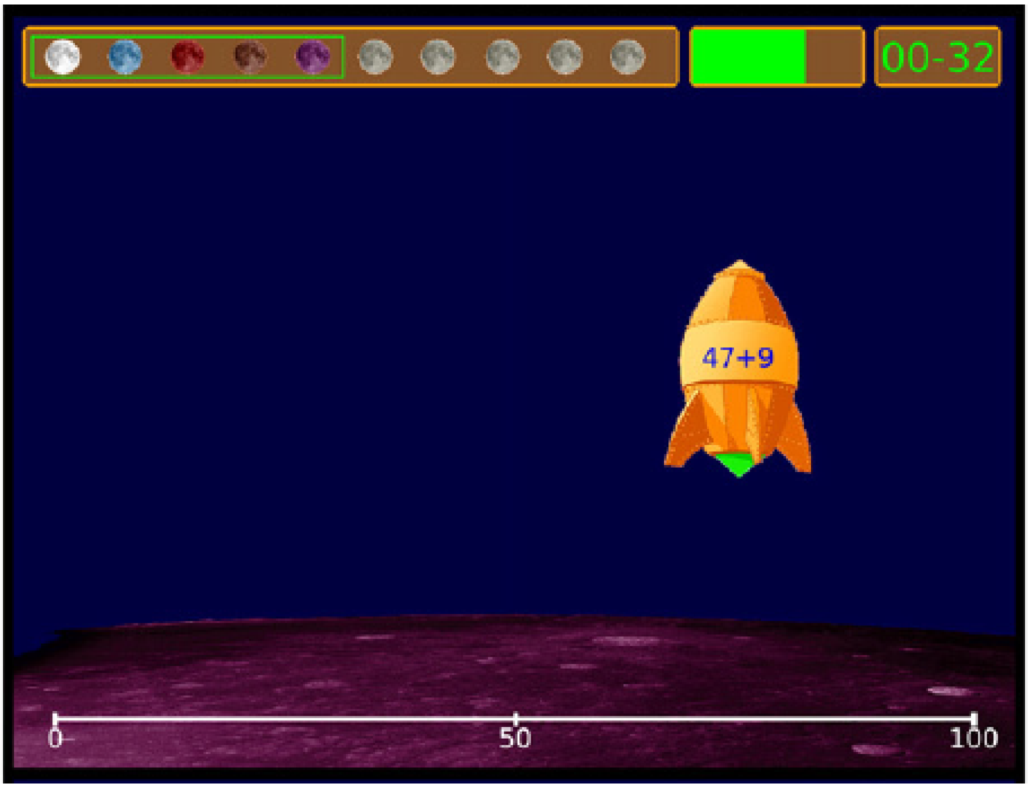

Figure 4.2: Example of Addition Task with Placement on a Numerical Line

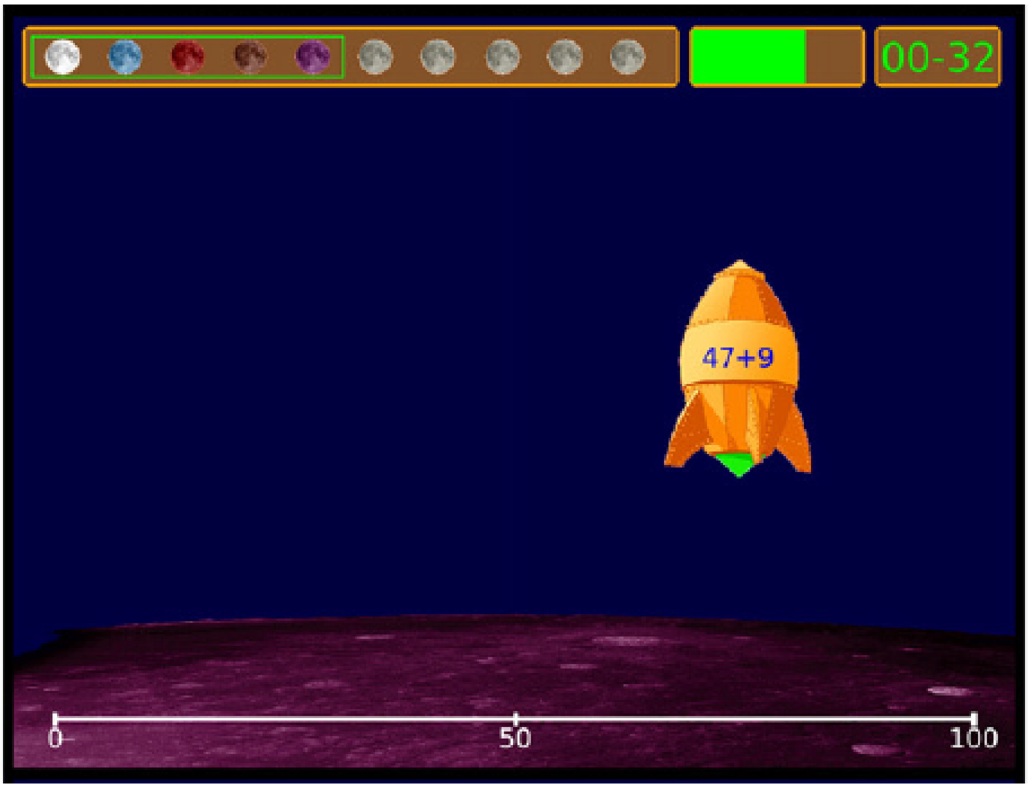

Figure 4.3: Example of Subtraction Task with Placement on a Numerical Line

The post-training results show that there have been consistent improvements in children's math skills, improvements seen in both dyscalculic and able-bodied children. However, as might be expected, the **dyscalculic children improved more than the others**. These results are also supported by analysis of the **fMRI images**: if, before training, a maximum activation in the superior frontal gyrus was observed in dyscalculic children, the same repeated post-training analysis led to a notable decrease in activation mainly in the areas of the frontal lobe, including the middle and upper frontal regions, but also in the left postcentral gyrus, the left intraparietal sulcus and the left insula in both groups, as shown by Figures 4.4 and 4.5. Finally, a note about the game: several feedbacks were collected from the children, who reported a pleasant and fun playing experience, as well as the fact that they found the game very easy to use.

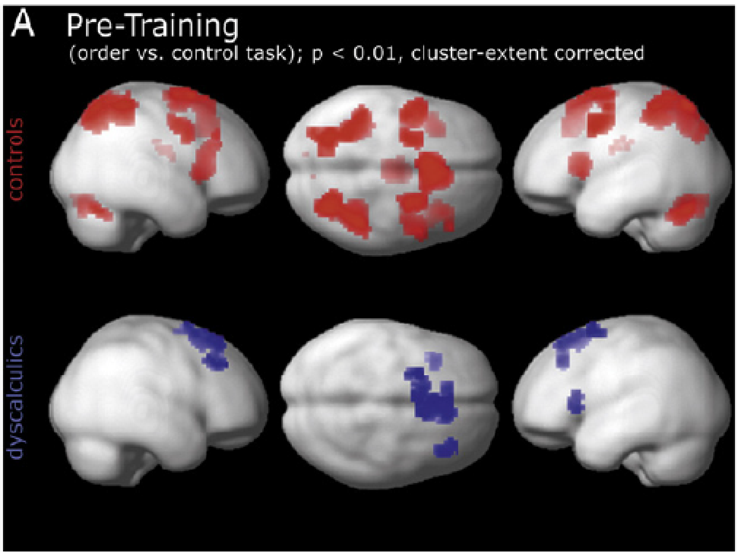

Figure 4.4: fMRI Image - Pre-Training

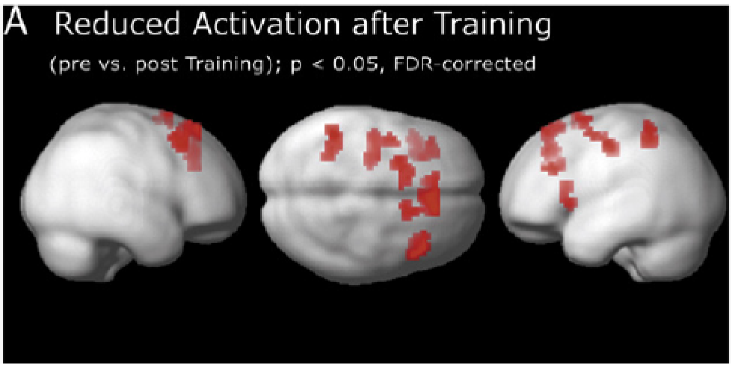

Figure 4.5: fMRI Image - Post-Training

4.3 The Number Race

The second example reported here relates to a very famous game app among the scientific community that deals with mathematical disorders: it is the game app "The Number Race", famous, as mentioned, because it was the first example of a game app based on a so-called *adaptive algorithm*, which had already found success in the past to train another SLD, namely dyslexia, but which had not yet been tested in the case of mathematical disorders. The design of the game, [4], has been thought to act on four aspects of dyscalculia, which we can think of as four goals of the app:

- To **enhance the Number Sense**: to pursue this goal, a *number comparison task* has been implemented as the primary task of the game, to work on the quantity representation in the player's mind, but also to emphasize the association between representation of number and space, which are supposed to be strictly related.
- To **cement the link between representations of number**: the goal is to create a solid connection between non-verbal quantity representation and the symbolic representations of numbers, such Arabic numerals or number words. To do so, a *repeated association task* was implemented, as shown in Figure 4.7, panel *a*.
- To **conceptualize and automatize arithmetic**: another goal was to reduce the delay in doing addition or subtraction operation, and this is why the game proposes *addition/subtraction tasks* in order to improve fluency. An example of addition task is reported in Figure 4.7, panel *b*.
- To **maximize motivation**: in order to avoid losing interest in the game due to poor results, an adaptive algorithm was implemented in order to maintain the level of difficulty which ensures performance at 75% correct.

The **adaptive algorithm** is certainly the most interesting part of the study. As mentioned, this is the first adaptive algorithm experiment used to train dyscalculia. In particular, for the implementation of this algorithm, three "dimensions" have been chosen to vary, based on the results obtained by the player himself. These dimensions are the following:

- the **distance**, intended as the distance between one number and another: the closer two numbers are, the smaller the distance between them, the more difficult it will be to solve a number comparison problem.
- the **speed** in resolving a task

- the **conceptual complexity**: it is a composite dimension, designed to teach children numerical symbols and elementary arithmetic.

The algorithm is designed so that there is a variation of one, two or all three of these dimensions, which, in fact, constitute a sort of **three-dimensional "learning space"**, imagined by the authors of the paper as a cube, along which it is possible to move in order to generate problems with a level of difficulty that suits to the player's performance level.

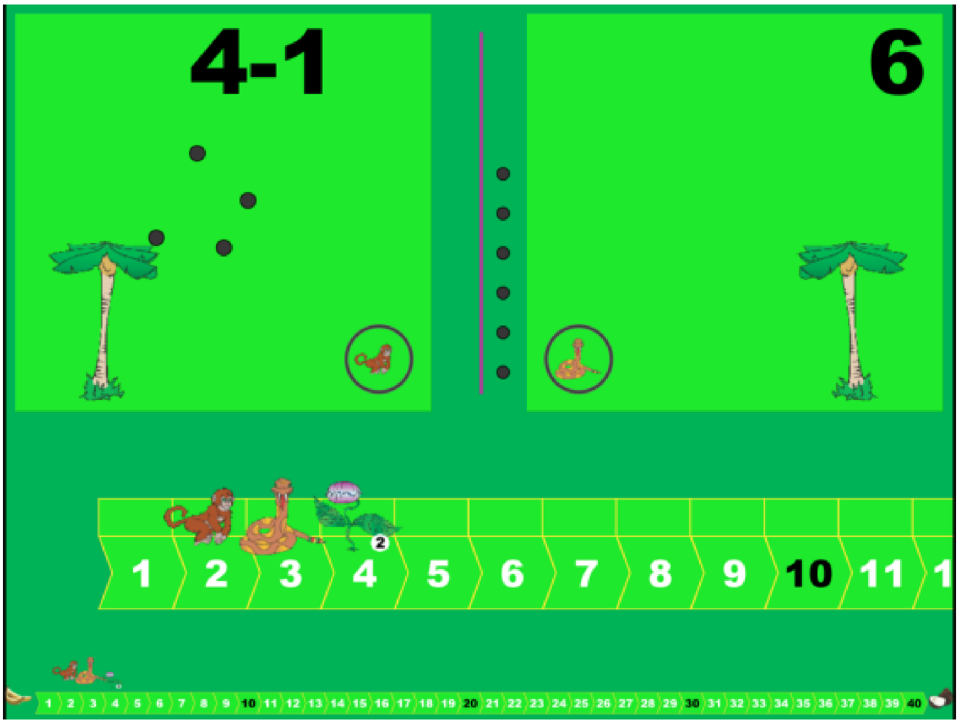

Figure 4.6: Example of game "The Number Race"

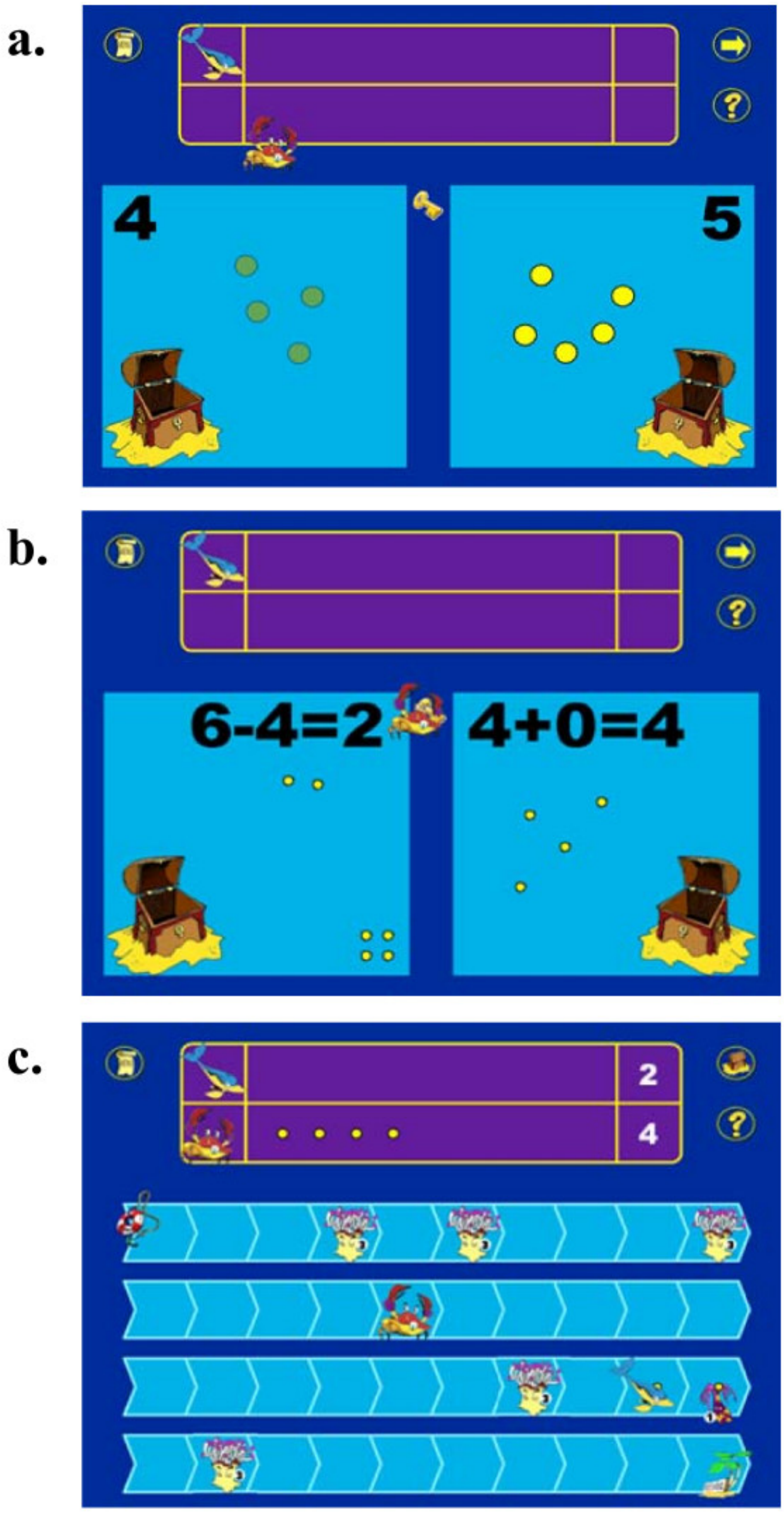

Figure 4.7: Example of game "The Number Race". a. Association Task between the quantity and its Arabic cipher. b. Addition Task. c. Mental Number Line Representation Task

The game was both simulated and tested. The simulations were then compared with the results obtained from the tests, and were quite faithful. The test results were reported from the second study related to "The Number Race", [5], which is slightly more practical in nature. Within the latter, all the tasks to which the tested children were subjected (dot enumeration, counting, base-10 comprehension, addition, subtraction), described in detail, and the improvements obtained task by task are reported. Summarizing, the **tested children progressed**, obtaining important performances on number sense tasks, both in terms of symbolic and non-symbolic comparison. Progress has also been recorded in the enumeration task and in the accuracy of the addition and subtraction operations, and this is consistent with the core deficit theory in children with DD. The only point on which there are no improvements is the base-10 comprehension, on which, however, there were no great expectations.

In conclusion, after carrying out the tests, it is possible to note that, out of the 4 objectives proposed by the theoretical paper, the fourth was certainly implemented correctly, given that the adaptive algorithm tested proved to work correctly. The first two objectives were also achieved, albeit not completely or perfectly. The third objective is the one that has been reached halfway: the arithmetic conceptualization is to be considered achieved, the full automation of mathematical operations instead is not.

The importance of this study and of the game that derives from it, however, lies in the fact that it is a **first example of an adaptive algorithm** used to train dyscalculia in an effective way, which is why The Number Race has become and still is famous. It certainly represents a **turning point in training DD**, although some aspects can be enhanced and improved: the first aspect is due to the fact that, at the time it was published, there was no consistent and rigorously scientific way to describe DD. Another aspect, linked to the validity of the tests, is to be found in the very limited number of participants in the study and in the absence of a control group.

4.4 Differences with respect to our game-app

Since the publication of "The Number Race" and the supporting studies in 2006, countless other games have been developed and released, all supported by related studies. For the sake of brevity, the discussion stops here, also because these two reported examples are sufficient to highlight some of the aspects that the game app explained in this thesis wants to improve.

One aspect to be highlighted is that none of the existing games in the literature provides for the implementation of an **Artificial Intelligence Assistance**: the implementation of this feature could automate the process of selecting the initial parameters in order to start the training, and subsequently the process of selecting the parameters as the player continues to play. The AI Assistance could be considered as an **evolution of the adaptive algorithm** proposed by "The Number Race". Another aspect to be considered is related to the fact that the apps analyzed in this chapter are targeted at school-age children who have been diagnosed with dyscalculia. This highlights a lack of apps which, on the other hand, could have the aim of **preventing** the onset of this pathology. A third aspect, already seen during the analysis of "The Number Race", is linked to the **testing phase**: the sample of children tested is too small to be able to provide scientifically state-of-the-art conclusions. In fact, in order for the results to be consistent, a testing phase that includes a sufficiently large sample, with at least 100 subjects, is required. Last but not least, none of the mentioned studies take into account **the importance of the interference effect between numerical and non-numerical variables**, a fundamental aspect, especially in subjects suffering from dyscalculia, which, as explained in the paper "Learning to Focus on Number", [6], struggle to focus on the numerical dimension, being greatly influenced by the non-numerical one.

To conclude and summarize, here are all the highlighted point of views, that the implemented game app wants to consider when doing the design.

- Implementation of the AI Assistance to automatize.
- Taking into account the possibility of implementing an app for preventing Dyscalculia.
- Test the app on a sufficient sample.
- Taking into account the importance of the Interference Effect.

Chapter 5

The Project

Before describing in detail the features of the developed game app, the subject of this thesis, it is useful to remember the objectives that characterize the development of the game app.

- To train the **child decision process** with the final goal of, hopefully, decrease the probability of developing in Dyscalculic disorder.
- To **train dyscalculic kids**, to improve their numerical abilities.
- To offer advanced training technology, based on **Artificial Intelligence**, to provide training that evolves with the evolution of the child's skills.
- To offer a **gaming experience** that is as pleasant and attractive as possible, to prevent the child from getting bored and not completing the training necessary for it to be effective.

It is also useful to remember that the game app targets pre-schooler children, between 3 and 5 years old, who are not diagnosed with dyscalculia, but already have difficulties in counting or in numerical tasks, but **also dyscalculic children**. Finally, it is useful to report the situation of the project before the start of the work, in order to better understand all the work done and what will be the next steps to be faced and developed in the future.

5.1 Starting point: where did we start?

The development of the entire project originated some time ago, by the colleague Fletcher Hurn, who, under the supervision of Prof. De Feo, began the implementation. In order to easily manage the two aspects that characterize the game, it was decided to divide the **graphic part**, therefore related to the implementation of the level and the graphic elements within it, from the part of **processing and collection of data**, which should contain everything related to the computation of the trials (both the initial one and the subsequent ones), Artificial Intelligence and data analysis.

For this reason, the entire game is divided into **Client Side**, focused on the graphic part, and **Server Side**, the subject of this thesis. The technologies used by the Client Side have been Unity (version 2020.1.9f1) for the development of the game environment, alongside Visual Studio, used as editor to write scripts (the code is written in C#), Blender and Mixamo for modeling and animating characters. Moreover, Github and Google Drive were used for the storage of the entire project.

The technologies used server side were many. In order to write the code, in Python language, **Spyder** was the chosen editor. Many libraries are used, especially Numpy and Pandas, but also Math library, used to exploit all the necessary mathematical functions. The Socket library was also used in order to implement an effective **communication protocol** with the client: to communicate with the client and to send and receive data, the server establishes a connection using the IP address 127.0.0.1 as host and 65432 as port, in agreement with the client. The functions available in the Socket library allow the server to build a bond using the specified host and port, through the **bind** method, and then to "listen" to the client, thanks to the **listen** method, waiting for the client to connect. When the client initiates a connection (using the same Socket library to create a connection on the same port that the server is listening on), the server accepts that connection with the **accept** method and creates a new client handler on a new thread. In other words, each connection is handled through a single thread: this choice has been taken to avoid any user getting locked out of the connection when multiple users are connected to the server. In this way, the **TCP connection is successfully established** and the client and server can start exchanging data. **The data is formatted by means of JSON objects**. The JSON format is used because it easily serializes both C# and Python objects. This is also why the classes in C# and Python need to be defined using the same variables, in order to allow an easier communication between client and server.

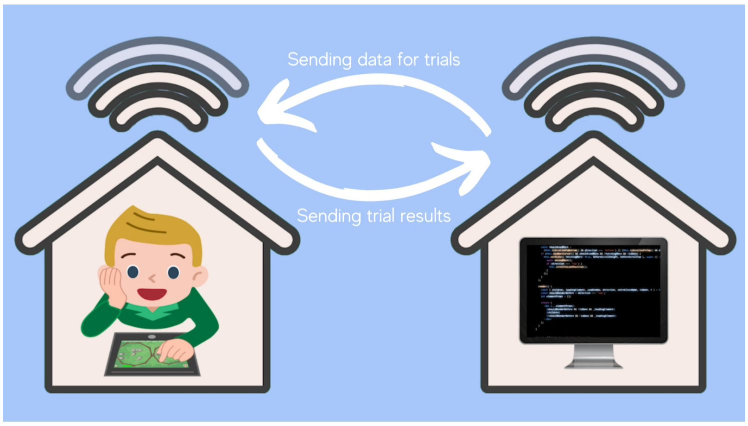

Figure 5.1: Representation of the communication Protocol between the Client and the Server, exchanging data.

Finally, to make the game more captivating, **a story** has been included to justify the graphical choices adopted. The game is set on a farm: the farmer James has so much work to do that he cannot count all the chickens he owns and who live on the farm, so he asks the player to help him out. James is also the character thanks to whom the rules of the game are explained, as shown in Figure 5.2: as soon as the level begins, the player will see two fences that are populated with chickens in a defined time based on the parameter provided on the server side. The player will have to guess which of the two fences contains more chickens, however he or she will have a very limited time available (this parameter, the time to guess, is also decided on the server side), which is why he or she will not be able, in fact, to count the chickens in the fences, but will have to guess. Time factor is very important in the game: the amount of time that the child needs to answer correctly is one of the variables needed by the AI in order to compute more and more suitable trials. Finally, James is the character who gives feedback to the player, communicating whether the answer provided is correct or not.

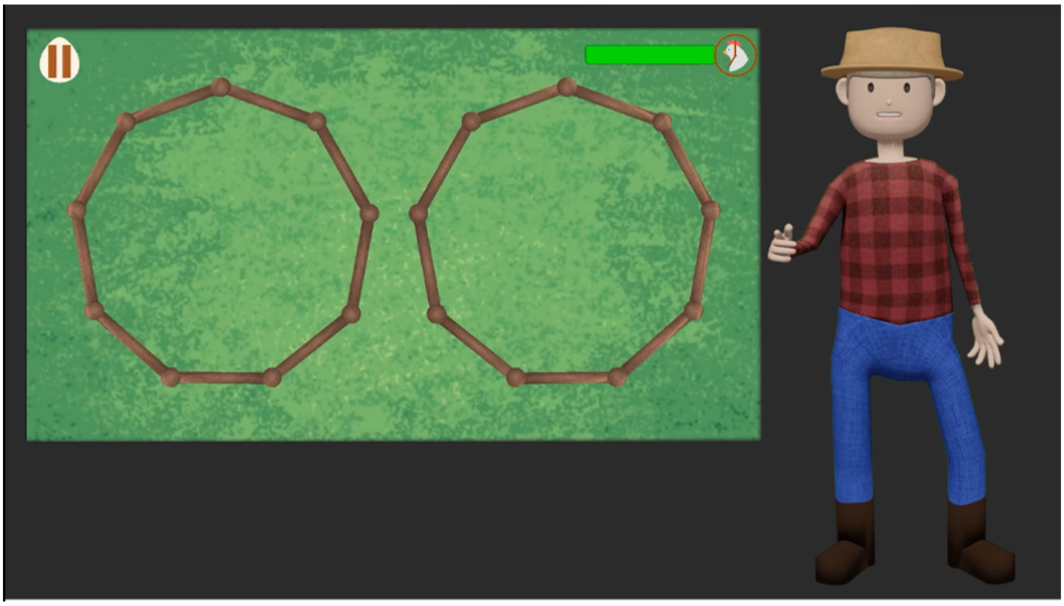

Figure 5.2: James explaining the Game Rules

Of course, for each trial, it is not only the number of chickens that varies, but also their size, the average space between the chickens and the size of the fence, indicated by its radius, to test the interference effect between the numerical variable, i.e. the number of chickens in the fence, and non-numerical ones, i.e. all the others listed. These are all parameters provided by the server, but which will be the subject of a subsequent paragraph. Analyzing the way in which the game is built (two fences containing the chickens), the connection with the experiment that inspired the game, [6], is quite immediate. In the experiment, in fact, the tested subject had to guess which of the two areas contains the greatest number of dots. The game is conceptually the same, and this can be caught graphically taking into account Figure 3.3: of course, being a game and not an experiment, changes have been made to make the experience more enjoyable. This justifies the presence of a story and an additional character, James, the change of the dots into chickens and the presence of the fences to contain them, instead of simple, white areas.

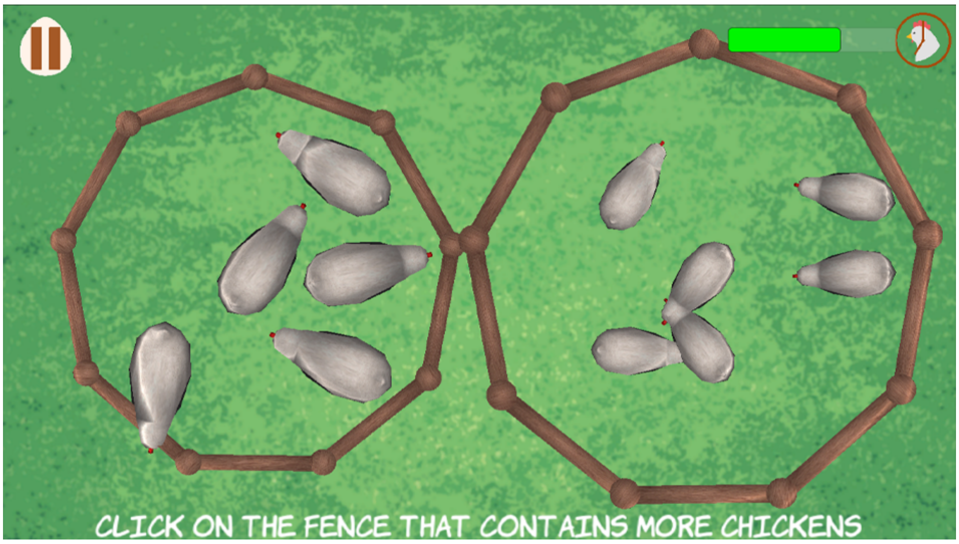

Figure 5.3: Fences with chickens: it is very easy to make a parallelism with Figure 3.3, replacing the dots with the chickens and the edge of the circle with the fence.

From Figure 5.3, it is possible to see an example of the game. Thanks to this picture, it is convenient to introduce the parameters that drive the game, at least client side. Each parameter that will be introduced refers to **one single area**; to play the game, we need to specify several parameters per each area of the game (left or right fence).

- **Number of Chickens**: it is the sole numerical variable, and sets how many chickens must be displayed in a single area.
- **Size of Chickens**: this non-numerical parameter determines how big

each chicken in a single area must be, so it measures the size of one chicken but applies to all chickens in the same area, so that all chickens in the same area will be equally big.

- **Circle Radius**: this is a non-numerical parameter, that gives information about the fence that contains the chickens. It was decided to set the radius instead of the Area because, from the Unity point of view, it was easier to manage.
- **Average Space Between**: this non-numerical parameter indicates the average space between every chicken and the other, inside the same area.

This list of parameters refers to the parameters that must be "doubled", therefore each parameter must be specified twice, one for each area.

Moreover, there are two other fields that are related to the time: **Chicken Show Time** controls the time available to see the chickens in the fences, and **Max Trial Time** indicates the overall duration of that specific trial (so it includes the Chicken Show Time plus some other time, in which the player sees empty fences and must decide which of the two fences contain more chickens).

An analysis of what has been implemented and developed server side will begin. Concerning the client part, it should be emphasized that this has been completely revolutionized compared to the previous work, greatly improving the graphic elements and therefore contributing greatly to the gaming experience. Although the main topic of this thesis is not the graphic part, it should be noted that the work between client and server went on in parallel, to progress together on both sides.

5.2 The Dummy Game and the Client Handler

As already specified above, the server allows to manage both real and simulated games. To do this, it was necessary to implement two classes: the **ClientHandler class** manages the real game while the **DummyClientHandler class** manages the simulated game. Having a simulated version of the game allows to test some features without running the real game. This is necessary as the Unity environment has a very high computational cost, which requires a suitable machine and a lot of time.

5.2.1 ClientHandler class

When the thread that manages the connection between client and server is created, an object of this class is instantiated. The class receives several parameters:

- **Client**: it is a new socket object, which can be used to send and receive data on the connection. It is returned back (alongside the address parameter) by the **accept** method, which is available in the Socket library.
- **DB**: this is an object of class **DBConnector**. DBConnector is a customized class, that includes all the methods required to manage the MySQL Database, named 'Dyscalculia', which contains two tables:
  - *player*: this table is used to store information relating to the players. In particular, this table will contain the player id, a unique identifier for each player (therefore, an integer number), a username and the IP address associated with that player.

	player_id	username	ip_address
▶	1	NULL	2130706433
*	NULL	NULL	NULL

Figure 5.4: Player Table.

It is possible to see in Figure 5.4 that there's one only player, for now. Ideally, in the future, this table will be populated by many players.

– *trial_result_new*: this table is fundamental to collect the results that are sent back from the client to server. It includes all the parameters that are passed to the client, the ones indicated in previous paragraph, for each area, that make the client work properly, alongside the *ID of the specific trial result*, used as primary key, the *player ID*, used to understand which of the players in the *player* table the entire row of results refers to. There's also a field named *correct*, which can assume binary values, 0 and 1, and indicates if the player has guessed the trial correctly or not. Specifically, **a correct value equals to 0 means correct answer, 1 means wrong answer**. Last field is named *created* and it is basically a timestamp of the moment in which the new entry is inserted in the table.

Column	Type
trial_result_id	bigint(20) unsigned
player_id	bigint(20) unsigned
correct	bit(1)
decision_time	bigint(20)
area_1_circle_radius	decimal(5,2)
area_1_size_of_chicken	decimal(5,2)
area_1_average_space_between	decimal(5,2)
area_1_number_of_chickens	int(11)
area_2_circle_radius	decimal(5,2)
area_2_size_of_chicken	decimal(5,2)
area_2_average_space_between	decimal(5,2)
area_2_number_of_chickens	int(11)
chicken_show_time	decimal(5,2)
created	datetime

Figure 5.5: Trial Result New Table

Having introduced the topic of database and the tables that compose the database of the project, it is useful to specify what are the **queries that allow the interaction with the DB**, which are defined in the DBConnector class, mentioned above. Moreover, since the code is written in Python, it was impossible to directly write and run SQL queries: this is why every query is wrapped inside a specific Python function, in a way that, once the function is called, the query is executed. Before talking specifically about the queries, it must be highlighted that, to make the functions and the submit of the queries effective, the connection with the DB must be established, and this is made through the function __init__:

```
def __init__(self):
    self.cnx = mysql.connector.connect(user='root', password='IvanaOrefice180496.',
                                       host='127.0.0.1',
                                       database='dyscalculia')
```

Figure 5.6: Initialization of the variable *cnx*, which establishes the connection with the DB and makes effective the communication with it, alongside the possibility to run the queries.

This way, the variable named *cnx* represent the connection, and thanks to it, it is possible to commit queries to the DB.

A total of 5 queries have been implemented by now:

- **Add player**: it adds a new player. This query allows the insertion of a new player inside the DB:

  ```
  INSERT INTO player (username) VALUES ('{}')
  ```

  ```
  INSERT INTO player (ip_address) VALUES
  (INET_ATON('{}'))
  ```

  There are two queries doing the same thing because we can decide to add a new entry in the *player* table either with the username of the player or the IP address of the device used by the player. By now, the IP address query is the used one. The other query has been added for completeness and because, in the future, it could be useful to have it.

- **Get Player**: it retrieves a specific player, given its IP address:

```
SELECT player_id FROM player WHERE ip_address =
INET_ATON('{}')
```

– **Add Result**: it allow the insertion of a new entry in the *trial_result_new* table, with all of its fields:

```
INSERT INTO trial_result_new (player_id, correct,
decision_time, area_1_circle_radius,
area_1_size_of_chicken, area_1_average_space_between,
area_1_number_of_chickens, area_2_circle_radius,
area_2_size_of_chicken, area_2_average_space_between,
area_2_number_of_chickens, chicken_show_time, created)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
```

area_1 refers to the left area, whereas *area_2* refers to the right area. It is useful, in this case, to report the entire code of the function that wraps up this query, in order to understand where the parameters are taken from.

```
def add_result(self, player_id, result):
    cursor = self.cnx.cursor()
    now = datetime.now()
    add_result = ("INSERT INTO trial_result_new (player_id, correct, decision_time, "
                  "area_1_circle_radius, area_1_size_of_chicken, area_1_average_space_between, "
                  "area_1_number_of_chickens, area_2_circle_radius, area_2_size_of_chicken, "
                  "area_2_average_space_between, area_2_number_of_chickens, chicken_show_time, created) "
                  "VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)")

    result_trial_data = result.trial_data

    area_1_data = result_trial_data.area1Data
    area_2_data = result_trial_data.area2Data

    data_result = (player_id, result.correct, result.decision_time, area_1_data.circleRadius,
                   area_1_data.sizeOfChicken, area_1_data.averageSpaceBetween,
                   area_1_data.numberOfChickens, area_2_data.circleRadius,
                   area_2_data.sizeOfChicken, area_2_data.averageSpaceBetween,
                   area_2_data.numberOfChickens,
                   result_trial_data.chickenShowTime, now)

    cursor.execute(add_result, data_result)

    self.cnx.commit()

    cursor.close()
```

Figure 5.7: Add Result Query Function

Figure 5.7 shows that the values that will be inserted are taken from the parameter given to the function, that is called *result*, and it must contain all the necessary fields to fill an entire entry of the table. This Figure shows also what are Python functions, included in the library named **mysql.connector**, that allow the actual run of the query: the function *commit* is the one that submits the query to the DB, namely the **execute and commit** functions.

- **Get Result**: it retrieves an entry of the table *trial_result_new* given the played ID

```
SELECT trial_result_id, correct, decision_time,
area_1_circle_radius, area_1_size_of_chicken,
area_1_average_space_between, area_1_number_of_chickens,
area_2_circle_radius, area_2_size_of_chicken,
area_2_average_space_between, area_2_number_of_chickens,
chicken_show_time FROM trial_result_new
WHERE player_id = {}".format(player_id)
```

It must be noticed that, once the data are retrieved, an object of class **Trial Result** is instantiated. This class is used to analyze and manipulate data received back from the Client, particularly to manipulate and see the response given by the player, alongside the correctness of the response, taking into account also the visualization of data.

- **player_id**
- **Trials Matrix**. The discussion related to the Trials Matrix will be addressed in the next paragraph. Just to give a hint, it is a matrix, made of one or more arrays, in which there must be all the needed parameters for the game to work out.

The class **ClientHandler** includes various function. One of the most important is the **run** function, whose code is reported in Figure 5.8.

```
def run(self):
    print("1 RUN FUNCTION")
    while True:
        data = self.connection.recv(2048)
        reply = self.get_reply(data.decode('utf-8'))
        if not data:
            break
        self.connection.send(str.encode(reply))
        print('Sent: ' + reply)
    print('Connection Closed')
    self.connection.close()
```

Figure 5.8: Run function of the ClientHandler class

This function is important because, being the object instantiated a Thread, it must be continuously "listening" to the client and to every modification that it does. This, also, justifies the presence of a "while" cycle, meaning that data reception is always active, until the connection is closed.

Another important method is the **get_reply** one. In it, there's a switch case. This switch is guided by the variable *data*, which is set Client side. Depending on the behaviour set client side, the function may react. The most common case is receiving the keyword **TRIALS**, which triggers the execution of another function, named **handle_trials_message**, thanks to which data are formatted in JSON format, by creating an object of class **Trial**. This class is fundamental to format the data, as said, in a JSON format, in a way that data is comprehensible for both Client and Server side. Once the data are formatted properly, i.e. they've "passed through" the Trial class, the exchange of data between Client and Server can be fluent.

The discussion about the ClientHandler will be stopped here. In the next paragraph, related to the implementation of the AI assistance, the Client Handler will come back.

5.2.2 DummyClientHandler

This class represents a **simulation of a kid playing the game**. In order to choose a simulated version of the game, it is possible to set a variable, called *simulation_on*, to 1, in *main.py*. To instantiate an object of class DummyClientHandler, it is firstly necessary to create a dummy trials matrix, which will be filled with parameters generated thanks to a specific function, called *dummy_matrix_generator*. The trial matrix, in the case of the simulated game, is created considering **a numerical variable** (the number of chickens) and the **two non-numerical variables** (i.e. the Field Area and the Item Surface Area (the discussion will be addressed in the next paragraph)). One of the two non-numerical variables remains fixed, while the other non-numerical variable together with the numerical variable vary, going from a minimum value to a maximum value, with a certain increment. This choice was adopted in order to **explore the space of possible game trials** and obtain a simulation as complete as possible. The function that allows the creation of a **simulated trial matrix** accepts two parameters:

- *nnd_selector*: this variable allows the selection of the non-numerical variable to vary. Particularly, if this variable is equal to 1, then it will be the Field Area to vary. Otherwise, it will be 2 to vary the value of the Item Surface Area.
- *nnd_number*: it controls the number of trials that will be computed by the Dummy, which will be equal to the this number elevated to 4.

It returns the trials matrix once it is computed.

```python
def dummy_matrix_generator(nnd_selector, nnd_number):
    if nnd_selector == 1:
        # nnd_selector = 1 means we want to vary the value of field_area as
        # nnd, alongside the numerical value, number_of_chickens. The other,
        # item_surface_area, is fixed
        item_surface_area = 10
        temp_matrix = partial_matrix_generator(0.2, 0.3, 0.2, 0.3, nnd_number
                                               , 1, 2, 1, 2)

        # We scan the obtained partial matrix, and for each row, we append the
        # various fields in the right position in the matrix
        for row in temp_matrix:
            trials_list = []
            trials_list.append(row[0])
            trials_list.append(row[1])
            trials_list.append(row[2])
            trials_list.append(row[3])
            trials_list.append(item_surface_area)
            trials_list.append(item_surface_area)

            trials_matrix.append(trials_list)

    elif nnd_selector == 2:
        # nnd_selector = 1 means we want to vary the value of
        # item_surface_area as nnd, alongside the numerical value,
        # number_of_chickens. The other, field_area, is fixed.
        field_area = 300
        temp_matrix = partial_matrix_generator(0.1, 0.1, 0.1, 0.1, nnd_number
                                               , 1, 1, 1, 1)

        # We scan the obtained partial matrix, and for each row, we append the
        # various fields in the right position in the matrix
        for row in temp_matrix:
            trials_list = []
            trials_list.append(row[0])
            trials_list.append(row[1])
            trials_list.append(field_area)
            trials_list.append(field_area)
            trials_list.append(row[2])
            trials_list.append(row[3])

            trials_matrix.append(trials_list)

    # print(trials_matrix)
    return trials_matrix
```

Figure 5.9: Code of 'dummy_matrix_generator' function

From Figure 5.9, it is possible to see that there's another function, which can be though of as a support function for the *dummy_matrix_generator*, which is called *partial_matrix_generator*. This function has the purpose of taking the initial value of the first and the second non-numerical value (there are two non-numeric variables, as it is necessary to have one for each area to be filled, both the right and the left one), as well as the first and the second numerical variable (one for the right area, one for the left area), alongside their step, also known as increment. Finally, it takes the *nnd_number* parameter, which is needed as a guide the cycle, i.e. understand when to stop the generation of new points.

Once the trials matrix is available, it is possible to instantiate an object of the DummyClientHandler class. This class has two functions:

- *Run*: it is the function that is called by the object of class DummyClientHandler, once this is created. The returned value is an array called **response_vector**, which contains information related to the correctness or incorrectness of the answers given. In the real game, the player can give a wrong or right answer; this answer is recorded as 0 in the case of correct one and as 1 in the case of incorrect one. The simulated game allows you to record the same information (although it comes from a simulation). The storage of the correctness/incorrectness of the simulated answers is done thanks to this array. Also, it appends the result of each trials to the correspondent row in the trials matrix, so that, in the end, the trials matrix will include the parameters plus the answer given.

- *ChildSimulator*: this is the **core function** of the Dummy game. It analyzes each trial in the trials matrix. Each row contains the parameters of a trial; the function wants to emulate the behavior of the child during a given trial, applying Filtering and Sharpening Hypothesis: in particular it returns 1 if the child answers correctly, 0 if not. This procedure is iterated row by row, returning, in the end, an array containing zeros and/or ones.

Focusing on the *ChildSimulator* function and the application of the Filtering and Sharpening hypothesis, it is useful to recall Figure 3.2. The idea was to rebuild panels B and C in the implementation of the simulation. About the application of **Filtering Hypothesis**, we represent the decision boundary that has an inclination α. For each trial, a point is plotted on the feature space, where the features are the numerical and non-numerical variables used in that trial: the point is plotted in the feature space alongside the decision boundary. We model the children behaviour as they were using a mental classifier with a given decision boundary. Exploiting this classifier, it is possible to assign each point (trial) to the class of correct answers (green points) or incorrect answers (red). If α is zero, the decision boundary is perfectly vertical, so there is no interference between numeric and non-numeric variables. As α increases, the interference of non-numeric variables on numeric ones increases.

The numerical acuity is modeled with a Gaussian probability density function (pdf) expressing the probability of individuate the trial in a certain region of the feature space (represented as a shadow surrounding the point in Figure 3.2). The Gaussian pdf is centered on the original point (trial) and has standard deviation σ which decreases as the numerical acuity increases. If σ is 0, the acuity is maximum. As σ increases, numerical acuity decreases.

The answer given to each trial will be appended to a vector, called **correct_vector**, which at the end of the *ChildSimulator* function, will be returned to the calling function.

A series of plot examples are shown in the Figures 5.10. 5.11, 5.12 and 5.13, as the α and σ values vary: α chosen values are 0°, 15°, 30° and 45°, whereas σ values are 0, 0.1, 0.2, 0.3. Our function simulates the child behaviour assigning each trial (dot) to the class of correct (green) or incorrect (red). This simulation function, by varying α and σ, takes into account both the mechanisms associated with the Filtering and Sharpening hypotheses.

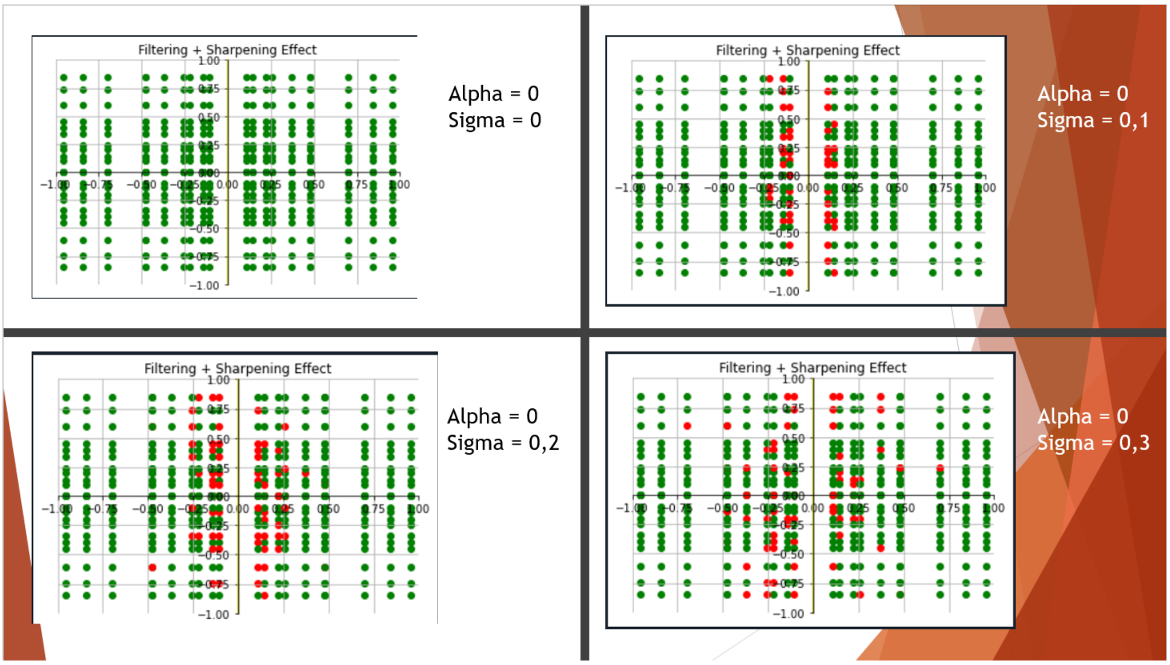

Figure 5.10:

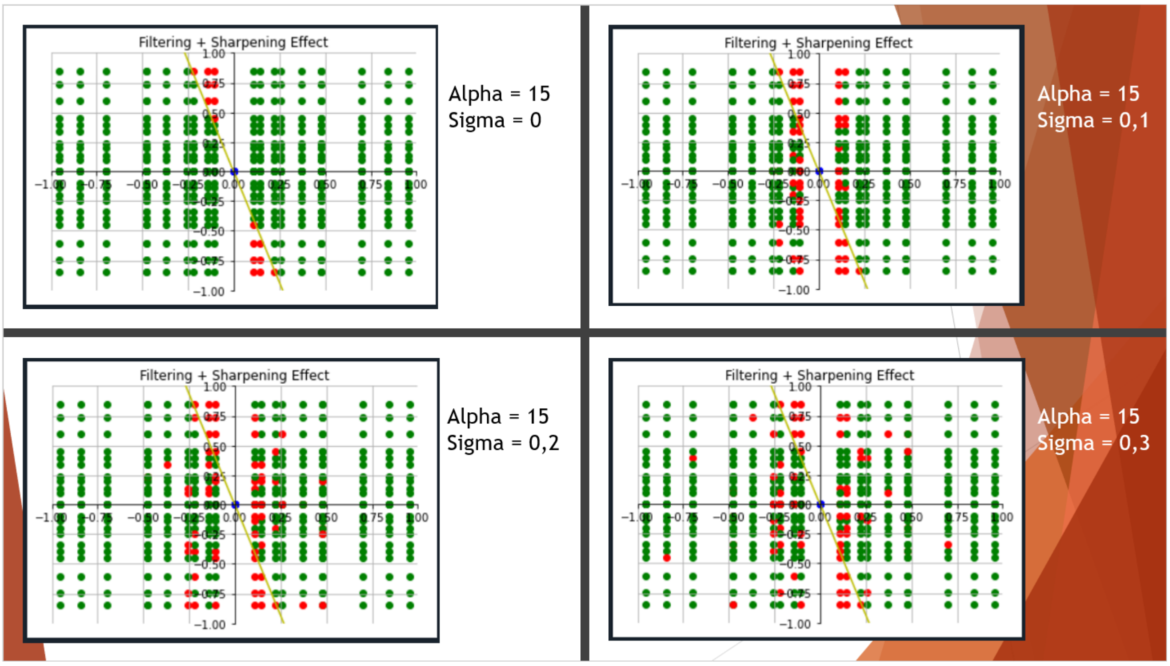

Figure 5.11:

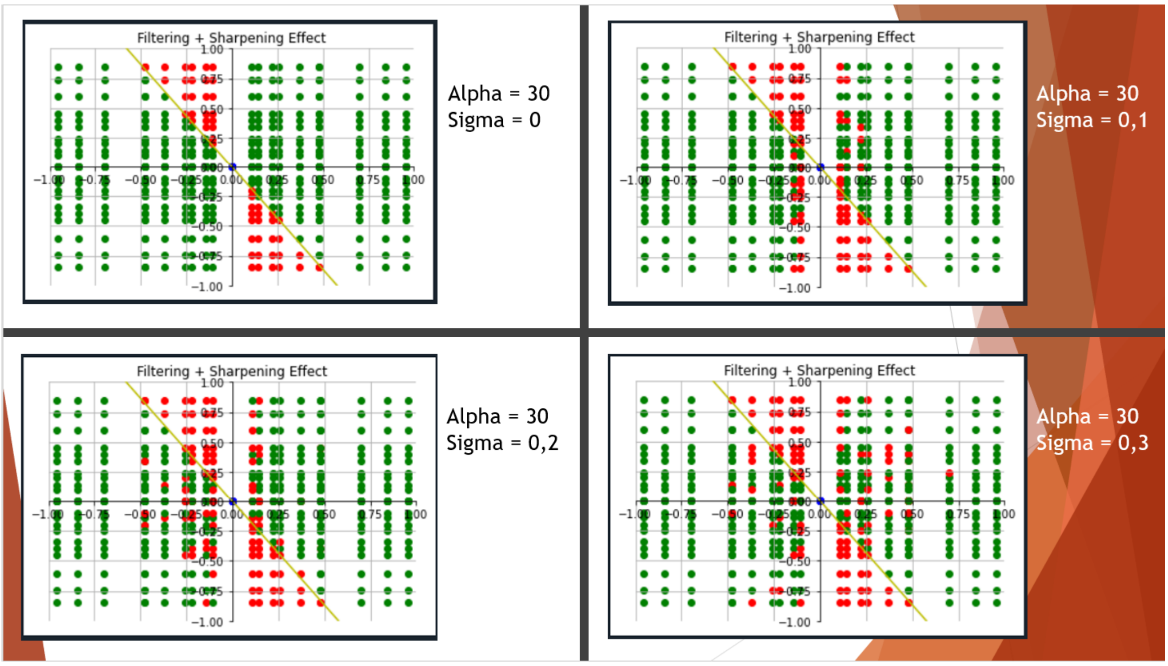

Figure 5.12:

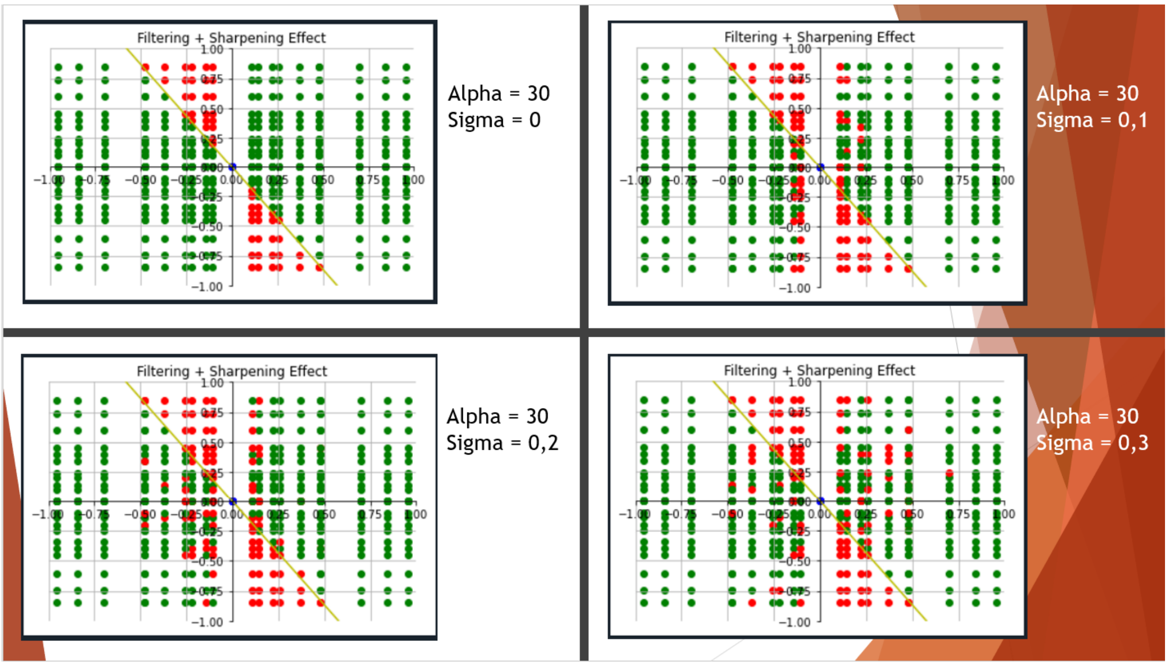

Figure 5.13:

The **PlotTrials function has been implemented**. The peculiarity of PlotTrials is that this function is independent of the type of game, whether it is simulated or real, and it is called to plot the trials in their space, to show how they vary. It also takes into account the correctness of the trials, because it receives as parameter the **response_vector**, which contains the information related to the correctness of each trial. This information is exploited to assess whether the answer was correct (green point) or not (red point). The library used to accomplish the plot of dots and the display of the graph has been *matplotlib.pyplot*.

5.3 Parameters used in the Literature and Unity

In literature, when doing experiments with numerical tasks, there are some non-numerical variables that are always used. Those non-numerical variables are actually the ones that are used also in the paper "Learning to Focus on Number"; although they have been extensively explained in Paragraph 3.3, for completeness they are also reported here:

- **Item Surface Area**
- **Total Surface Area**
- **Field Area**
- **Sparsity**

The game does not use those variables, but instead it is based on some other variables or parameters, which, from now on, will be defined **game variables or parameters**. The reason why there's this difference between the variables used in literature and the game variables is because the game variables better suit to the Unity environment and, in general, to the graphic part. The game variables are:

- **Number of Chickens**: it indicates how many chickens must be displayed, for each area.
- **Size Of Chickens**: it is a variable related to the dimension of the chickens displayed in each area.

- **Circle Radius**: it measures the radius of the fence that contains the chickens.
- **Average Space Between**: this variable indicates the average space between each chicken inside each area. It is also used to **assign a position to the chickens**.

The goal was to find a correspondence between the variables used in the literature and also in the experiment that has inspired this app, and the game variables. To discover this correspondence, it is useful to report what is the NND and its formula: the NND (Non-Numerical Dimension) is the variable in which all the non-numerical variables that characterize the experiment are condensed. In paper, "Learning to Focus on Number", for the particular dataset considered in that paper, each non-numerical variable has a weight on the NND; by considering each weight of each variable, it is possible to find the following formula:

$$NND = 0.577 * SPARS + 0.487 * ISA + 0.473 * TSA + 0.467 * FA$$

The dataset analyzed in the experiment has reported some other formulas, thanks to which we can express some of the non-numerical variables exploiting some other non-numerical variables and the numerical one.

$$TSA = ISA * Number - Of - Dots$$
$$SPARS = \frac{FA}{Number-Of-Dots}$$

Thanks to this formulas, it is possible to **rewrite the NND formula**. It is possible to replace 2 out of 4 non-numerical variables, thanks to the introduction of he numerical variable. This also allows a **space dimensionality reduction** from 4 non-numerical variable to 2 non-numerical plus 1 numerical variable.

$NND = (0.577 + 0.467 * Number) + \frac{FA}{Number-Of-Dots} + (0.487 + 0.473 * Number - Of - Dots) * ISA$

By analyzing the meaning of each of the four non-numerical variables, which have guided the experiment, and the meaning of the game variables, new considerations on how to do the bind have emerged:

- The **Number of Dots** is directly connected with the **Number of Chickens**, meaning that each dot will correspond to a Chicken in the Game.
- The **Field Area** and the **Circle Radius** are connected, because the area of a circle can be calculated as $\pi * Radius^2$
- The **Item Surface Area** is bonded with the **Size Of Chickens** in a 1:1 relationship.

Once those relationship have emerged, it was necessary to find some conversion formulas, to be used to convert the data expressed in the "literature variables" format into "game variables" format. It is important to notice that, before trying to find conversion formulas, a set including all the possible combinations that worked with the Unity game environment was already available. To find the conversion formulas, the procedure followed ca be synthesized in the following steps:

- The variables found to be *conceptually* connected were taken into account (for example, Field Area and the Circle Radius).
- The ratio between the Left and the Right quantity was calculated for both (i.e. ratio between Left FA and Right FA and same for Circle Radius was calculated), obtaining a maximum and minimum value.

- Comparing the ratios obtained. This allowed to define some constraints on the values to assign to the game variables.

By analyzing the ratios, it emerged that it was necessary to **change the minimum values of Size of Chickens and the Circle Radius**: the minimum value of Size of Chickens was lowered from 4 to 2.5, whereas the Circle Radius minimum value was lowered from 0.8 to 0.45. Nothing was changed concerning the Number of Chickens and the Number of Dots.

This led to the creation of a new space of possible solutions for the game, containing **all used combinations** of Number of Chickens, Size of Chickens and Circle Radius, whose ratios are compatible with the ratios emerging from the study of the experiment dataset.

Finally, by considering the maximum and minimum values of the "connected" variables, it was possible to obtain the transformation formulas that take, as input, the experiment variable and give, as output, the game variable. The transformations are shown in the Figures down below:

$$CR\ (circle\_radius) = \frac{\sqrt{AREA_{Unity}}}{\pi} = \gamma \frac{\sqrt{AREA_{Unity}}}{\pi} = \alpha\sqrt{FA}$$

$$CR^2 = k * FA \rightarrow k = \alpha^2 = \frac{CR^2}{FA} = \frac{(VALUE\_MAX\_CR)^2}{VALUE\_MAX\_FA} = \frac{1.2^2}{39522} = 3.6 * 10^{-5}$$

$$\alpha = \sqrt{k} = \sqrt{3.6 * 10^{-5}} = 6 * 10^{-3}$$

Figure 5.14: Circle Radius calculation, starting from the Field Area

$$SC^{\,2}(size\_of\_chicken) = AC\ (area\_chicken) = \gamma * ISA$$

$$SC = k * \sqrt{ISA} \rightarrow k = \frac{SC}{\sqrt{ISA}} = \frac{VALUE\_MAX\_SC}{\sqrt{VALUE\_MAX\_ISA}} = \frac{10}{\sqrt{308{,}2}} = 0.57$$

Figure 5.15: Size of Chicken calculation, starting from the Item Surface Area

Last thing to consider is the Average Space Between: it is a measure of how much space exists between the chickens, therefore, to obtain an analytically correct and precise calculation, it should be linked both to how big the fence is (i.e. the Circle Radius) but also to how big the chickens are (i.e. the Size of Chicken). However, given that, for each combination, a maximum and minimum value of Average Space Between were found, and that the value of Average Space Between could be in this range, this allowed a margin to simplify the calculations. For each Size of Chickens value, from the minimum to the maximum the minimum possible value of the corresponding Average Space Between was taken into consideration, obtaining the following table:

SIZE of CHICKENS	AVG SPACE BETWEEN
2,5	0,6
3	0,6
3,5	0,61
4	0,61
4,5	0,67
5	0,73
5,5	0,9
6	0,99
6,5	1,13
7	1,2
7,5	1,21
8	1,27
8,5	1,6
9	1,61
9,5	1,61
10	1,61

Figure 5.16: Size of Chickens and the correspondent value of Average Space Between

From this table, considering the values in the first and last row, the equation of the line passing through two points was used, in order to find a relationship that connected the Size of Chickens to the Average Space Between. The relationship that emerged is, therefore, the following:

$$Avg\_Space\_Between = (Size\_Of\_Chickens * 0.167) + 0.183$$

The other found formulas are:

$$Circle\_Radius = 6 * 10^{-}3 * \sqrt{(Field\_Area)}$$

$$Size\_Of\_Chicken = 0.57 * \sqrt{(Item\_Surface\_Area)}$$

All those transformation formulas were used, in the Python code, to implement a function, called *TransformMatrix*, that takes, as input, a matrix with values expressed in the "experiment" format and gives, as output, a matrix made of values in the "game" format.

```
def TransformMatrix(trials_matrix_original):

    for row in trials_matrix_original:
        area_1_circle_radius = ((0.006) * math.sqrt(row[2]))
        area_2_circle_radius = ((0.006) * math.sqrt(row[3]))

        area_1_size_of_chicken = 0.57 * math.sqrt(row[4])
        area_2_size_of_chicken = 0.57 * math.sqrt(row[5])

        area_1_average_space_between = (0.167 * area_1_size_of_chicken) + 0.183
        area_2_average_space_between = (0.167 * area_2_size_of_chicken) + 0.183

        trials_row = []

        trials_row.append(area_1_circle_radius)
        trials_row.append(area_2_circle_radius)
        trials_row.append(area_1_size_of_chicken)
        trials_row.append(area_2_size_of_chicken)
        trials_row.append(area_1_average_space_between)
        trials_row.append(area_2_average_space_between)
        trials_row.append(row[0])   # Number of Chicken 1
        trials_row.append(row[1])   # Number of Chicken 2
        trials_row.append(row[6])
        trials_row.append(row[7])

        trials_matrix.append(trials_row)

    return trials_matrix
```

Figure 5.17: TransformMatrix function code

This entire analysis led to the definition of a three-dimensional space of possible combinations: the Number of Chickens, the Field Area and the Size of Chickens were reported on the axes, and it is reported in Figure 5.17

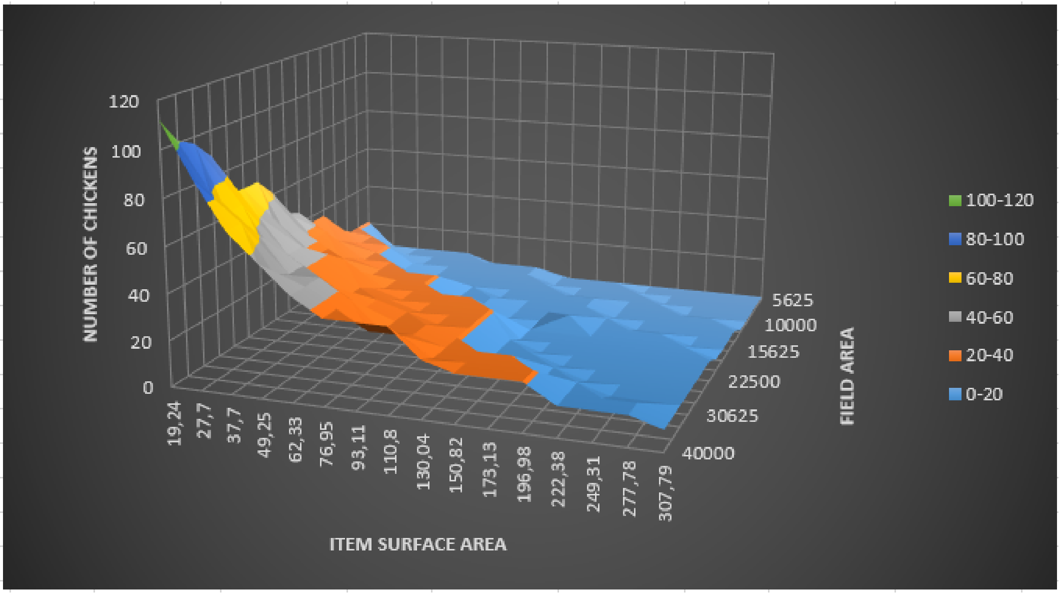

Figure 5.18: 3D space of possible combinations

The space below the multi-colored curve represents combinations that lead to acceptable values, both for the game and for the data provided by the original experiment. For further information on the subject, please refer to Appendix A.

5.4 Exploring the Feature Space

To verify that the space of combination obtained allows the game to work correctly without crashing, some tests were carried out with the aim of testing the software, so that one could move within the space of combinations indicated by the Figure 5.18. It represents an important result for what concerns the **future development of AI assistance**. It is good to recall what are the **purposes of the AI**:

- Initially, the AI must select the parameters that characterize the **first trial**. When a kid logs for the first time to the game, the AI must select some parameters, but the AI must know the parameters considered to be acceptable by the game in order to be able to select them.

- After the first ever game, the kid keeps playing and, hopefully, he or she will improve his or her mathematical skills. The **AI must be able to evolve with the kid's mathematical skills improvement**, so it must select more and more trials parameters that are still acceptable (i.e. they are selected considering the space of possible combinations) but that fit the improvement of the kid, helping the kid to continue the improvement.

This thesis focuses on the step preceding the development of AI, that is to explore the space of possible combinations and evaluate their validity. The first function to analyze is called *ValidTrial*: it takes, as input parameters, the point coordinates, made of Number, Field Area and Item Surface Area, and returns, as output value, a Boolean value set 0 in the case of an invalid point, 1 if the point and its coordinates are acceptable. Recall that **the three coordinates represent a valid point if the point lies below the curve showed in Figure 5.18**. A valid point is a combination of parameters that are acceptable by the game. The function code is showed in the figure down below.

```python
def ValidTrial(number, fa, isa):

    dataset = UploadDataFromExcelDataset()
    isValid = 0

    for i in range(len(dataset)):
        if(number <= dataset[i][0]):
            # Number ok. Check for FA validity
            print("Number ok")
            if(fa <= dataset[i][1]):
                # FA ok. Check for ISA validity
                print("FA ok")
                if(isa <= dataset[i][2]):
                    # ISA ok. The point is valid, break the loop
                    print("ISA ok --> Point valid")
                    print(dataset[i][0])
                    print(dataset[i][1])
                    print(dataset[i][2])
                    isValid = 1
                    break
                else:
                    # if ISA is not valid, continue to the next iteration
                    print("ISA not ok")
                    continue
            else:
                # if FA is not valid, continue to the next iteration
                print("FA not ok")
                continue
        else:
            # if number is not valid, continue to the next iteration
            print("number not ok")
            continue

    # Reaching the end of the loop means that no match is found, so isValid is 0,
    # as it was set at the beginning
    return isValid
```

Figure 5.19: ValidTrial code

The Figure shows that the three coordinates are checked to be below the correspondent **value of the combination set**. The combination set is organized with three columns, respectively the Number of Chickens, the Field Area and Item Surface Area, and includes **256 entries that represents the possible combinations**: the function must check whether the Number-Field Area-Item Surface Area combinations stay below the Number-Field Area-Item Surface Area that is available in the combination set. **If all of three are verified to be less than the ones of the i-th entry in the combination set, then the point is set to be valid;** otherwise, the i value goes on to check the following entry. **If there is no correspondence between the combination of the three parameters and the entries available in the combination set, the point is judged to be invalid**.

Once the *ValidTrial* function was ready, it was necessary to test it: the first point to be tested was the **Medium Point**: it was thought to be a point where the three coordinates should be selected, more or less, as the medium of their possible, acceptable values, and, moreover, it had to be a valid point, so that when the coordinates of the chosen point were given as parameters to the *ValidTrial* function, the function had to return 1. Having chosen the medium point coordinates, those coordinates were selected as the **starting point** for the selection of the trial, thanks to the *GenerateNewTrial* function.

This function takes several input parameters:

- Number
- Field Area
- Item Surface Area
- Minimum Increment for the Number, set to 1
- Minimum Increment for the Field Area, set to $(Max\_Value+Min\_Value)/100$

- Minimum Increment for the Item Surface Area, set to $(Max\_Value + Min\_Value)/100$
- an additional variable i, used to drive the while loop in the function.

The function, whose code is shown in Figure 5.20, must randomically generate a value equal to -1, 0 or 1, one for each of the three dimensions that have been passed as parameter: if the value is equal to 1, then the new value of the dimension must be equal to the previous value added to the minimum increment defined for that dimension; if, instead the random number is equal to -1, then the new value of the dimension must be equal to the previous value subtracted to the minimum increment defined for that dimension.Finally, if the random number is equal to 0, the dimension value remains as it is, without adding or subtracting anything.

After this procedure, three new coordinates have been obtained, that are different from the initial medium point, so it is needed to check the validity of the new point found. If the new point is valid (i.e. its coordinates are acceptable), then that point is **selected as the initial point to start the training**. Alternatively, a new point must be found, starting from the original coordinates.

```python
def GenerateNewTrial(i, num, min_inc_num, fa, min_inc_fa, isa, min_inc_isa):

    while (i < 257):
        i = i + 1
        random_dim_1 = random.randint(0, 2) - 1 # for number
        random_dim_2 = random.randint(0, 2) - 1 # for FA
        random_dim_3 = random.randint(0, 2) - 1 # for ISA

        # Tener traccia delle coordinate di prima
        if (random_dim_1 == -1):
            new_num = num - min_inc_num
        elif (random_dim_1 == 1):
            new_num = num + min_inc_num

        if (random_dim_2 == -1):
            new_fa = fa - min_inc_fa
        elif (random_dim_2 == 1):
            new_fa = fa + min_inc_fa

        if (random_dim_3 == -1):
            new_isa = isa - min_inc_isa
        elif (random_dim_3 == 1):
            new_isa = isa + min_inc_isa

        pointIsValid = ValidTrial(num, fa, isa)
        if(pointIsValid == 0 or num < 0):
            ax.scatter3D(fa, isa, num, marker='<', label="Invalid", c="red")
            return GenerateNewTrial(i, num, min_inc_num, fa, min_inc_fa, isa, min_inc_isa)
        else:
            print(num)
            print(fa)
            print(isa)
            print("point is valid")
            ax.scatter3D(new_fa, new_isa, new_num, marker='o', label="Valid", c="green")
            return GenerateNewTrial(i, new_num, min_inc_num, new_fa, min_inc_fa, new_isa, min_inc_isa)
```

Figure 5.20: GenerateNewTrial function code

In the code, it is possible to see a while cycle, that has been introduced to test the space of possible combinations. A plot of the exploration of this space has been produced, where green points are acceptable and red ones are unacceptable:

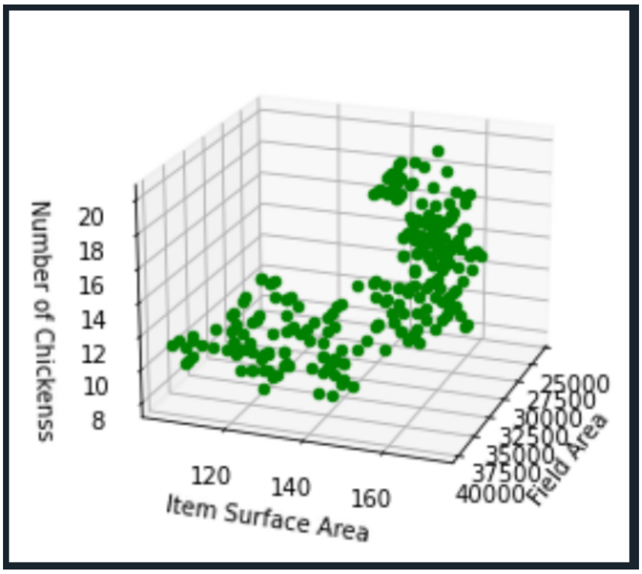

Figure 5.21: Possible outcome of the GenerateNewTrial execution (First Example).

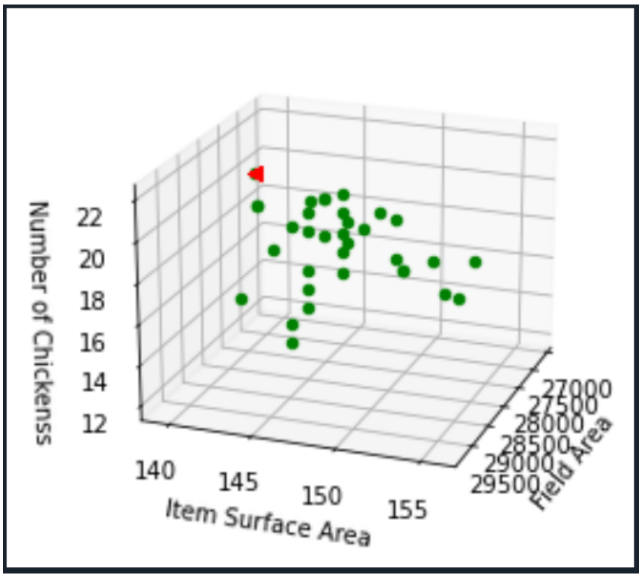

Figure 5.22: Possible outcome of the GenerateNewTrial execution (Second Example). Take into account that the two plots are so different because the generation of new point is guided by the randomic generation of values between -1 and 1. Those plots represents also an **exploration of the space**, confirming what has been found previously (see Figure 5.18).

Chapter 6

Future Developments and Conclusions

There's still a lot of work to do to mark this project as completed, even if important results have been obtained during the work that has been told in this thesis. To conclude, it is good to report what are those results obtained, that represent the starting point for future developments.

The first, important result obtained is related to the **completion of the implementation of the DummyClientHandler**. The simulated version is very important: it allows to tests may functionalities without actually running the real game, and so without using many resources and much time, that, instead, it would be needed for running the real game in the Unity environment. Moreover, with this stable version of the DummyClienHandler class, there's the implementation of the *ChildSimulator* function, which represents an accurate simulation of the kid behaviour when playing the game and also the simulation of the **Filtering and Sharpening hypothesis** application.

Second: **the definition of a space of possible combinations for the Game**, which are based on literature but that can also satisfy the practical needs of the game, from an implementation point of view, had a huge impact on the project, not to mention that it can give the input to the birth of the Artificial Intelligence assistance. From now on, anyone working on the project will be able to choose clearly and safely **which trials (intended as combination of parameters) can be used for the game**, so which are acceptable and which are unacceptable.

Finally, the third important result obtained is a direct consequence of the second, thanks to which it was possible to explore the space of possible combinations, evaluating if the combination is acceptable or not.

It is quite easy to identify what are the **future developments** of the project, from the server-side point of view.

- The functions related to the Generation of a New Trial can be used in the future for designing the AI. Particularly, by now, the exploration of the space is done randomically. The AI must select the trials in an intelligent way, based for example on the age of the kid, exploiting the validation function to see if a combination is acceptable or not.
- After, the AI must be able to compute trials that evolve with the evolution of the numerical skills of the kid. This mechanism is totally missing, by now, and it must be implemented from scratch.
- The game must be tested, with a fairly large sample of at least 100 subjects. This phase is important to obtain a confirmation of the fact that the training actually works and leads to concrete benefits.

My hopes lie in my successors: I hope that the project is soon completed, because I am sure that its use in the real world, with real children, will bring great benefits, both at school and in their everyday life. Working on this project was incredible: discovering the world of Neuroscience, being able to work in contact with dynamic people, with great stimuli, has contributed a lot to my growth. I leave the project in good hands!

Appendix A

Parameters used in the Literature and Unity

This appendix represents an in-depth analysis of the Paragraph 5.4.

One of the challenges of this project has been to make it consistent with the variables that are normally used in Literature, which are Field Area, Item Surface Area, Total Surface Area and Sparsity. This part of the work was carried out in strong collaboration with the developer of the client part, the colleague Gaia Brugo. This was necessary because every small change had to correspond to a client-side test to verify that the game continued to work correctly. Furthermore, the dataset born from this work had to be perfectly in line with the possible combinations accepted by the game, on the client side.

Before introducing the starting point of this portion of work, it is useful to report the game variables, present on the client side and, originally, also on the server side.

- **Number of Chickens**: it indicates how many chickens must be displayed, for each area.
- **Size Of Chickens**: it is a variable related to the dimension of the chickens

displayed in each area.

- **Circle Radius**: it measures the radius of the fence that contains the chickens.
- **Average Space Between**: this variable indicates the average space between each chicken inside each area. Moreover, it is fundamental when **assigning a position to the chickens inside a fence**: as it is shown by Figure 5.3, the game is made by two fences and chickens go inside those two fences. Considering only the portion of space within the fence, so considering the radius of the fence itself, the average space between has been used to calculate a sort of matrix of positions, also called **grid of positions**: this grid is, therefore, used when it is necessary to choose the position to be assigned to each chicken that is expected to be shown. It is not possible to choose a position that is not present in this grid.

The grid of positions is calculated client-side, as it is part of the game design implementation, which is why all of these listed parameters are still present on the client side. For this reason, it was decided that server-side had to implement all the chages.

A careful review of the paper "Learning to Focus on Number" has been conducted, especially on how the stimuli space is made. Although the review of the parameters that guide the stimuli space in the experiment has already been carried out in Paragraph 3.3, it is useful simply to report the names of these 4 parameters are reported: **Item Surface Area, Total Surface Area, Field Area and Sparsity** (to review the meaning of each of these parameters, please refer to Paragraph 3.3). Those variables represents the non-numerical features of each trial of the experiment, and they're accompanied by the numerical variable, the **Number of Dots** displayed in each area;

therefore **5 features define our feature space, 1 is numerical and 4 are non-numerical**.

With the aim of condensing the non-numerical variables into one, the **Non-Numerical Dimension** was calculated, briefly called **NND**, which is defined as the first principal component (estimated by means of Principal Component Analysis) of the set of the 4 previously cited non-numerical variables. **Each of the non-numerical variables has a weight on the NND**, which have been already cited in Paragraph 3.3. This specific weight of each non-numerical variable on the NND, combined with all the other weights, leads to the following relationship:

$$NND = 0.577 * SPARS + 0.487 * ISA + 0.473 * TSA + 0.467 * FA$$

This formula shows that all of the 4 non-numerical variables are equally loaded on the NND. By carefully studying the original dataset of the experiment, which contains all the parameters, in terms of FA, ISA, TSA and Sparsity, **new relationships have emerged**, that bind the non-numerical variables to the numerical one, which is completely absent in the previous formula.

The new relationships were the following:

$TSA = ISA * Number - Of - Dots$

$Spars = \frac{FA}{Number}$

Having found these new relationships, the idea was to **rewrite the NND formula by replacing two of the four non-numerical variables and introducing, in their place, the numerical variable** which cannot be replaced in any way within the game, but indeed it is crucial to guide the number of chickens to be shown in each area/fence. Moreover, this choice has the advantage to **reduce the feature space from four non-numerical variables to 2 non-numerical and 1 numerical variables**, and the new obtained formula is the following:

$$NND = (0.577+0.467*Number)*\frac{FA}{Number}+(0.487+0.473*Number)*ISA$$

where $\frac{FA}{Number}$ is the substitute for the Sparsity and $ISA * Number$ replaces the TSA.

This has led to a new theoretical awareness: it has opened up to the possibility to bind some game variables to some other experiment variables, but, to understand the binding, it was first necessary to carefully study the dataset analyzed in the paper. It contains the following fields:

- **Age** of the subject: it goes from 3 to 33. For now, the age, in the game, is not taken into account.
- **Left and Right Number**: those fields indicate how many dots are displayed in the left and right area, respectively.
- **Left and Right Sparsity**: they refer to the Sparsity values in the left and right areas, respectively.
- **Left and Right Item Surface Area**: those indicate the values of the ISA both for left and right areas.
- **Left and Right Total Surface Area**: as for the ISA, those represents the values of Total Surface Area for left and right side.
- **Left and Right Field Area**: they state how big the white disc area is, in terms of measured area.

By analyzing the meaning of each of the four non-numerical variables, which have guided the experiment, and the meaning of the game variables, new considerations on how to do the bind have emerged:

- The **Number of Dots** is directly connected with the **Number of Chickens**, meaning that each dot will correspond to a Chicken in the Game.
- The **Field Area**, in the experiment, is related to the size of the area in which the dots fall into, one for each area. In the game, the dimension of the area/fence is guided by the **Circle Radius** which is not a measure of the area of the fence in which the chickens fall into, but can be easily connected with it, since the fence is, in Unity, designed as a circle, and the area of the circle can be calculated as $\pi * Radius^2$. This is why Field Area and Circle Radius are connected.
- The **Item Surface Area** is a measure of the size of the dots. In the game, there's a variable, namely the **Size of Chicken**, which, as the name already suggests, indicates how big the chickens in the areas are. Of course, there's one Size Of Chickens for each area, as any other game parameter. Again, the ISA and the Size of Chicken have been connected one another in a 1:1 relationship.

After having found a correspondence between the variables of the game and those of the experiments described in the paper, it was also necessary to find a "practical" correspondence, that is to **understand the right way to transpose the variables of the experiment numerically** in such a way that, once these had become game variables, the values obtained were acceptable for the game.

To achieve this goal, once again, the original dataset was studied. This time, however, the numerical values assigned to each of the parameters that characterized the experiment were taken into account. The dataset contains all the possible combinations of Number, FA, ISA, TSA and Sparsity that characterized the trials, one for each area.

It is useful to note that, before this theoretical study, there exists some values that were assigned to the game variables, so this means that possible combinations of Number, Circle Radius, Size of Chicken and Average Space Between were already available, with which the game it works properly. The challenge, at this point, focuses on **understanding whether the combinations found and tested on Unity find a theoretical match with the data available in the dataset analyzed in the paper**.

It was decided to proceed by considering the **ratios** between the quantities on the left and those on the right: to give an example, considering the Field Area and the Circle Radius, which are the variables that have been found to be *conceptually* connected, the ratio between the left and right Field Area was calculated, taking into account the values in the dataset, then doing the same thing for the left and right Circle Radius, this time taking into consideration the values found by the team, through different tests on the game. In the end, these ratios were compared, to verify if the **attempts made directly on Unity are consistent with what is present on the dataset**. The ratios were calculated for both the maximum and minimum values, as to have both a *lower bound and an upper bound for the game variables*. Therefore, the following ratios were found:

Left Num	Right Num	ratio Num
32,00	20,00	1,6
16,00	17,00	0,94

Figure A.1: Maximum and Minimum ratio for the Number of Dots

Left FA	Right FA	ratio FA	ratio radius
31818	4892	6,5	2,55
35941	35949	1	1

Figure A.2: Maximum and Minimum ratio for Field Area

Left ISA	Right ISA	ratio ISA
246,46875	76,3409091	3,23
106,28125	106,833333	0,99

Figure A.3: Maximum and Minimum ratio for Item Surface Area

The ratios for the Number of Chickens, the Circle Radius and the Size of Chickens were then calculated. Evaluating the ratios, two nw considerations have emerged. The range relating to the **Size of Chickens** has been increased in order to comply with the limits imposed by the calculation of the ratio, **bringing the minimum value of the size of chickens from 4 to 2.5 and maintaining, instead, the maximum value fixed at 10** for compatibility reasons with the game itself (it has been proven that it is impossible to represent chickens larger than 10). The same was done for the Circle Radius: to respect the ratios, the **minimum value** of the Circle Radius was **lowered from 0.8 to 0.45**, while maintaining the constraint according to which the **sum of the right and left circle radius must be, at most, equal to two**. Regarding the Number, there were no particularities that emerged from this analysis: the values to be assigned to the Number of Chickens is a variable that is fairly free from any constraint (compatibly with the values assumed by the non-numerical variables).

For the sake of completeness, it should be noted that this analysis led to the **creation of two datasets for the game**: one of them contains all possible combinations, including what can be defined as the *most challenging* ones, which is why it can be interpreted as a dataset indicated only for **adult training**; another dataset contains a smaller number of combinations, which are *easier to guess*, and for this reason suitable to **train adults**, of course, **but also children.**

Having, therefore, found a dataset for all the possible combinations for the game variables, of course **maximum and minimum values have emerged**. By exploiting the maximum and minimum values of two "connected" variables (for example, Circle Radius and Field Area, or Item Surface Area and Size of Chickens), it was possible to obtain the transformation formulas that allow to accept the experiment variables and turn them into game variables. The transformations are shown in the Figure A.4 and A.5.

$$CR\ (circle\_radius) = \frac{\sqrt{AREA_{Unity}}}{\pi} = \gamma \frac{\sqrt{AREA_{Unity}}}{\pi} = \alpha\sqrt{FA}$$

$$CR^2 = k * FA \rightarrow k = \alpha^2 = \frac{CR^2}{FA} = \frac{(VALUE\_MAX\_CR)^2}{VALUE\_MAX\_FA} = \frac{1.2^2}{39522} = 3.6 * 10^{-5}$$

$$\alpha = \sqrt{k} = \sqrt{3.6 * 10^{-5}} = 6 * 10^{-3}$$

Figure A.4: Circle Radius calculation, starting from the Field Area

$$SC\ ^2(size\_of\_chicken) = AC\ (area\_chicken) = \gamma * ISA$$

$$SC = k * \sqrt{ISA} \rightarrow k = \frac{SC}{\sqrt{ISA}} = \frac{VALUE\_MAX\_SC}{\sqrt{VALUE\_MAX\_ISA}} = \frac{10}{\sqrt{308{,}2}} = 0.57$$

Figure A.5: Size of Chicken calculation, starting from the Item Surface Area

The discussion relating to the Average Space Between is separated from the other non-numerical variables. At the beginning of this study, it was thought to be related to Sparsity, however its use in the game, i.e. for calculating the grid of positions, showed that this was not true, because the Average Space Between is used to calculate the grid of positions: if the Average Space between increases, then the positions that can be occupied will be reduced because, in fact, the space among the various chickens must increase.

Despite this, having previously found a database of possible combinations for the Circle Radius, the Size of Chicken and the Number of Chickens, the minimum and maximum values of the Average Space Between were found for each combination, and those values have been tested during game runs and allow a correct functioning of the game. An example is reported in Figure A.6.

circle_radius	size_of_chickens	average_space_between		number_of_chickens
0,45	2,5	0,6	0,6	21
0,45	2,5	0,67	0,61	13
0,45	2,5	0,95	0,68	9
0,45	2,5	1,34	0,96	5
0,45	2,5	3,5	1,35	1

Figure A.6: Example of Possible Combination when Circle Radius = 0.45 and Size of Chicken = 2.5. This example shows that, by varying the value of the Average Space Between, the allowed Number of Chicken varies from a minimum of 1 to a maximum of 21. This clarifies also the **importance of the Average Space Between** chosen value, but it also explains that, to obtain a specific Number of Chickens, there's not just a specific value of Average Space Between, but it is a range that gives us the same result in terms of Number of Chickens.

The Average Space Between is a measure of how much space exists between the chickens, therefore, to obtain an analytically correct and precise calculation, it should be linked both to how big the fence is (i.e. Circle Radius) but also to how big the chickens are. (i.e. Size of Chicken). However, given that, for each combination, a maximum and minimum value of Average Space Between were found, and that therefore the value of Average Space Between could be in this range, this allowed a margin to simplify the calculations. In particular, for each Size of Chickens value, from the minimum to the maximum, therefore from 2.5 to 10, the minimum possible value of the corresponding Average Space Between was taken into consideration, obtaining the following table:

SIZE of CHICKENS	AVG SPACE BETWEEN
2,5	0,6
3	0,6
3,5	0,61
4	0,61
4,5	0,67
5	0,73
5,5	0,9
6	0,99
6,5	1,13
7	1,2
7,5	1,21
8	1,27
8,5	1,6
9	1,61
9,5	1,61
10	1,61

Figure A.7: Size of Chickens and the correspondent value of Average Space Between

From this table, considering the values in the first and last row, the equation of the line passing through two points was used, in order to **find a relationship that connected the Size of Chickens to the Average Space Between**. The relationship that emerged is, therefore, the following:

$$Avg\_Space\_Between = (Size\_Of\_Chickens * 0,167) + 0,183$$

At this point, the study on the dataset ends: all the relationships that allow to implement the transformation from the variables and data of the experiment to those of the game have been found. The last relationship found is linked to the Average Space Between, and for completeness the other two found are also reported:

$$Circle\_Radius = 6 * 10^-3 * \sqrt{(}Field\_Area)$$
$$Size\_Of\_Chicken = 0.57 * \sqrt{(}Item\_Surface\_Area)$$

As said at the beginning of this Paragraph, those transformations found have been implemented server-side, leading to the definition of a function named *TransformMatrix*, which accepts a parameter, named *trials_matrix_original*, and applies all the transformations to the proper fields, returning a trials matrix that fits the game, in terms of game parameters. The code is reported down below.

```
def TransformMatrix(trials_matrix_original):

    for row in trials_matrix_original:
        area_1_circle_radius = ((0.006) * math.sqrt(row[2]))
        area_2_circle_radius = ((0.006) * math.sqrt(row[3]))

        area_1_size_of_chicken = 0.57 * math.sqrt(row[4])
        area_2_size_of_chicken = 0.57 * math.sqrt(row[5])

        area_1_average_space_between = (0.167 * area_1_size_of_chicken) + 0.183
        area_2_average_space_between = (0.167 * area_2_size_of_chicken) + 0.183

        trials_row = []

        trials_row.append(area_1_circle_radius)
        trials_row.append(area_2_circle_radius)
        trials_row.append(area_1_size_of_chicken)
        trials_row.append(area_2_size_of_chicken)
        trials_row.append(area_1_average_space_between)
        trials_row.append(area_2_average_space_between)
        trials_row.append(row[0])   # Number of Chicken 1
        trials_row.append(row[1])   # Number of Chicken 2
        trials_row.append(row[6])
        trials_row.append(row[7])

        trials_matrix.append(trials_row)

    return trials_matrix
```

Figure A.8: TransformMatrix function code

This entire analysis led to the definition of a dataset of values of parameters, from which a **three-dimensional space of possible combinations** was obtained: the Number of Chickens, the Field Area and the Size of Chickens were reported on the axes, and it is reported in Figure A.9.

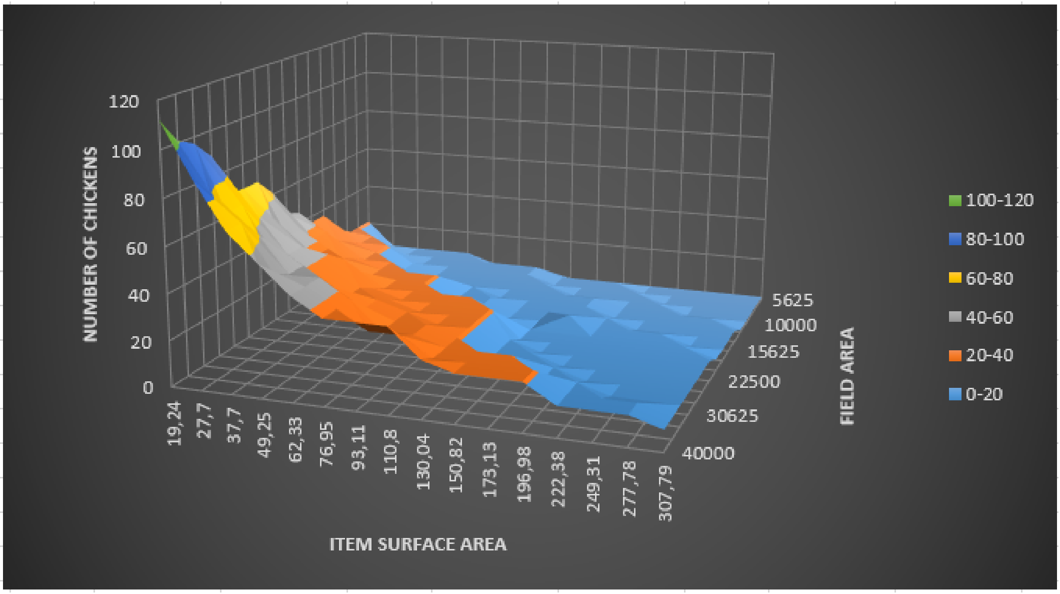

Figure A.9: 3D space of possible combinations

The space below the multi-colored curve represents combinations that lead to acceptable values, both for the game and for the data provided by the original experiment.

Appendix B

Acknoledgments

This entire thesis project has been a ride, and what a ride! I had fun, panicked many times, doubted about myself and my programming skills and I've learned so much, especially on how to be a Computer Engineering and still be caring about creating solutions that can save people's life (or, at least, make their life better!). For this reason, I would like to thank Prof. Vito De Feo. He took me on this ride, gave me the tools to get to know the world of Neuroscience and inspired me to keep working, and working harder and better, because our work matters and can make the difference. Thanks, professor for the most mindblowing experience. I'll remember this last 9 months of working to this incredible project for the rest of my life. I'd like to thank as well Prof.ssa Gabriella Olmo, for having encouraged and calmed me about deadlines and how to write the thesis in an effective way. Thanks, for having been so involved during the writing process and for all the advices you gave me. Finally, I'd like to thank my colleague and friend Gaia. My dear friend, we did it! I am so glad to have worked with you for 8 months and I couldn't make it all alone. Thanks for the constant sharing of ideas and worries and joys. I grew up a lot thanks to you and your work, and I'll always be grateful for this.

It's been three long, very long years since I've been in Politecnico, and the

world has completely changed: probably, it was the longest three years of my life, and, today, this long chapter of my life ends. Seems sad, somehow, but it wouldn't be me without a little bit of meloncholy. Today, thinking about not just this last three years, but starting back in 2015 after high school, I just think: what a long journey you did, girl! I'm here because I dared to dream. I dreamt so many dreams to be here today: the little province girl, now graduating in a important university, crowning her dream to be an Engineer. But I've always been this dreamy, since I was a kid. Despite dreaming is considered to be a beautiful thing, my dreams were so painful sometimes. I have cursed my dreams many times throughout those last 7 years, blaming them for my unhappiness, for my anxiety, for my fear to not being good enough, for my fear of not doing it. I loaded them with an immeasurable weight, and they peacefully carried this cross with me. The truth is, it hasn't been easy. I though to leave university two or three times a day, especially at the beginning. But I always knew that my destiny was to become what I am now, an Engineer. And, finally, here I am, after 7 long years, where everything changed. But before moving on to the next chapter of my life, I need to thank people that, somehow, contributed to my life and to my decisions, bringing me to the place in which I am today.

First of all, I must thank the city of Turin. My relocation happened in a period of my life in which I felt I didn't belong to no place, like I wasn't feeling home nowhere, which is a feeling that I never felt before. I felt lost and broken and I needed to discover myself belonging to something again, something that could help defining my identity, which, in March 2019, was completely absent, or better it was completely twisted. In Turin I found not jut a temporary house, but I place that I can call home, in which I grew up and became the person I want to be as an adult woman, a place I which I discovered and rediscovered myself, again, in a more mature and "adulty" way. I don't know how my life will be from now on, but a piece of my heart will forever stay here, in Turin.

I would like to thank all my women-cousins that are Engineers, particularly my cousin Michaela: she inspired me in choosing of moving to Turin and apply to Politecnico, giving me the courage to do such a choice, and being interested in all the steps of my university career. I am very grateful for that.

Thanks to my godfather, Giorgio, for the endless phone calls, for the invaluable advice and for thinking about me when I don't think about it. You are the best in the world, thank you.

Thanks to my beloved GiFra family. Each and everyone one has changed me, and I'm so lucky to call you brothers and sisters. you're my Family-in-the-north. I never thought I'd find another family in such a new and different place, yet you have been (and will be) a key element in helping me find my way home. My dearest president, Ludovica. You're the craziest, how can I live without texting about how mad we are?? I wish that our relationship will remain as it is forever. My sweet Chiara. You're the funniest girl in the world, and your empathy inspires me to be a better woman. My little sister, Lidia. you showed me that it is possible to be friends into a family, and also you showed me how easy it is. I'm so lucky to call you sister and friend. And finally, thanks to Enzo. I never had a big brother, and God knows how I desperately needed one. And then He sent me you. I'm so lucky to have you as a brother and as a guide. Thanks for being this to me.

Thanks to Brunella, my dearest housemate. You literally gave me a home, throwing open the doors of what will always be our home, giving me a place within our little family, and I can never be grateful enough for that.

Thanks to Alessandra, my actual and dearest housemate. How can I express in words how thankful for you am I? You are one of the most beautiful people I know, patient, empathetic, kind. You acted as my big sister by always having the right word for me at the exact moment I needed it. You always made me feel like a person of great value, you always believed in me, during every exam and interview. I am grateful to have you in my life.

Thanks to all of my girls: Anna, Antonella, Camilla, Dalila, Domenica, Martina and Roberta. I love you to pieces and spending time with you all is always quality time. To have you in my life means to smile and laugh everytime we are together. Thanks for every chit chat, ever single moment when I came back in town, thanks for making me feel like nothing has changed even if I moved to another city. Thanks because, despite the distance, you still make me feel home and safe. I am forever grateful for this.

Particularly, thanks to Dalila. You're more than just a friend, you are my sister. You're part of my family and I'm so glad I got to spend with you so many memories, since we were babies. You're my certainty and you have a great power: transforming every place in which we are together into a home for us. Thank you, cuore.

Particularly, thanks to Martina. You were there during the hardest times of my existence, recalling to my memory the valuable person that I am, always seeing in me something good that deserved to be saved, always believing in me. I would not be standing here today if it wasn't for you, you know that, and I'll always be grateful for this.

Thanks to my grandmothers. Oh, how I love you! You're my $mummy^2$ and the number one reason for which I take an airplane everytime to come back with a smile on my face. To see the pride on your eyes today, looking at my success, this is the light of my life. Grandma Zina: I wish I could one day have your ability to never lose heart. Grandma Carmelina: I wish one day I could be your empathy, and the right words to say everytime, for everyone. I couldn't be luckier.

Finally, the most sincere thanks.

To my little (but great) sister, Irene. You're the light of my life, my greatest hope. I dedicate to you every single, beautiful thing I've accomplished, hoping I can see you happy and satisfied with your life and your choices as much as I am today. You're my sincerest smiles.

To Ignazio, my dad. You're the most brilliant and smart person I know, and also the person who pushes me to overcome all my limits. It is thanks to your metaphors on offsides and corner kicks that you help me solve the biggest dilemmas of my life. Thanks for making it that simple.

To Laura, my mum. You're not just my mum, you're a sister and a friend, my partner in this chaotic life, always ready to hear my feelings and help me through them, always keeping me at the center.

Mum, Dad, I'm so grateful to God for having been born in your family. The three of you you are my rock, my shield and my fortress, the reason why this whole absurd and incredible dream exists and is true. You have believed in me without ever wavering for a second, you have given me my time and my space and you have understood me, without having claims on me. This is not taken for granted, not even in a family. I love you, thanks for everything.

I dedicated my previous thesis work to my grandfather Nino. He taught me pride and the strength to fight for your goals. When I came back after an exam, he immediately wanted to know how it went and, although sometimes he was not so happy with the grade, without even trying to hide it, he never stopped believing that I was number 1. He did not get a chance to see me crowned, but I haven't stopped feeling his presence, not even for a moment, and I know that he is here with me now and that he is bursting with pride for his little Oreficina.

I dedicate this thesis work to my grandfather Ignazio. He taught me that no one gives you anything, that everything you work for is yours and that even if at a certain moment no one recognizes your merit, time is able to settle the scores and make the knots come to a head, just have a little patience! He also taught me that the most important thing in one's life is family, and the tenderness and dedication with which he has dedicated himself to me throughout my life, I'll keep in my heart with great affection and gratitude forever.

I wish I could conclude these thanks by saying "and finally, thanks to me, because I made it". The truth is that nobody saves himself alone: of course, I have worked hard, but what saved me was having people who believed in me, even and above all in the moments when I did not believe in myself. Without you, without your encouragement, without your reminding me that I am a worthy person, I would have given up on the second day of university (whoever was there during the first year of my Bachelor knows it). So, cheers to you!, to your successes and thank you for giving me my dream: to be an engineer, but most of all, to be a woman in the exact way I had dreamed.

Appendix C

Ringraziamenti

Questo progetto di tesi è stato un viaggio, e che viaggio! Mi sono divertita, sono andata in panico tante volte, ho dubitato di me stessa e delle mie capacità come programmatrice e ho imparato così tanto, sopratutto ho imparato come essere un'Ingegnera Informatica e avere a cuore la creazione di soluzioni che possono salvare la vita delle persone (o, quantomeno, rendere la vita più semplice!): per questo motivo, vorrei ringrazire il Prof. Vito De Feo, che mi ha accompagnato in questo viaggio, dandomi gli strumenti per conoscere il mondo meraviglioso della Neuroscience mi ha dato l'ispirazione per continuare a lavorare, sempre di più e sempre meglio, perchè il nostro lavoro è importante e può fare la differenza. Grazie, professore per questa esperienza magnifica. Ricorderò questi 9 mesi di lavoro a questo progetto incredibile per il resto della mia vita. Vorrei ringraziare anche la Prof.ssa Gabriella Olmo, per avermi incoraggiato e calmato per quanto riguarda la stesura della tesi in modo efficace. Grazie, per essere stata così tanto presente durante il processo di scrittura e per tutti i consigli. Infine, ringrazio tantissimo la mia collega e amica Gaia Brugo. Ce l'abbiamo fatta, non mi sembra vero! Sono così felice di aver lavorato con te per 8 mesi, non ce l'avrei fatta da sola. Grazie, per il tempo e per la condivisione costante di idee, ma anche di gioie e preoccupazioni e ansie. Sono cresciuta moltissimo confrontandomi con te, e per questo te ne

sono molto grata.

Sono stati tre lunghi, lunghissimi anni al Politecnico, e il mondo è cambiato totalmente nel frattempo: probabilmente, sono stati i tre anni più lunghi della mia vita, e, oggi, questo lungo capitolo della mia vita finisce. Sembra triste, in qualche modo, ma non sarei io senza una punta di malinconia. Oggi, ripensando non solo agli ultimi tre anni, ma a partire dal 2015 dopo il liceo, penso solo: che viaggio lungo che hai fatto, ragazza! E sono qua perchè ho avuto **il coraggio di sognare**. Ho sognato tantissimi sogni per arrivare qua: la ragazzetta di provincia, che si laurea al Politecnico di Torino, coronando il suo sogno di essere un'Ingegnera. Oggi sono qua perché ho avuto il coraggio di sognare. Ho fatto tanti, tanti sogni per arrivare qua. Sognare, anche un po' troppo ad occhi aperti, in fondo, mi ha sempre caratterizzato. Eppure, in questi anni, i miei sogni li ho maledetti molte volte, li ho incolpati della mia infelicità, della mia ansia, delle mie paure di non essere all'altezza, di non farcela. Li ho caricati di un peso incommensurabile, e loro, pacificamente, hanno portato con me questa croce. E la verità è che non è stato facile. Ho pensato di mollare almeno due o tre volte al giorno, soprattutto all'inizio di tutto. Però, io l'ho sempre saputo che io dovevo diventare ingegnera. Ed eccomi, sono arrivata, dopo 7 lunghi, intensi e spiazzanti anni, in cui è successo di tutto. E prima di passare al prossimo capitolo della mia vita, ho bisogno di ringraziare le persone che, in qualche modo, hanno contribuito alla mia vita e alle mie decisioni, portandomi a questo punto.

Per prima cosa, devo ringraziare la città di Torino. Il mio trasferimento è arrivato in un punto della mia vita in cui non sentivo più appartenermi a nessun luogo, come se non mi sentissi a casa da nessuna parte, sentimento, tra l'altro, che non avevo avuto mai il piacere di provare. Mi sentivo smarrita e a pezzi, e avevo bisogno di tornare a riscoprirmi appartenente a qualcosa, qual-

cosa che mi caratterizzasse e aiutasse a definire la mia identità, che, a Marzo del 2019, praticamente non c'era, o meglio era stata stravolta completamente. A Torino ho trovato non solo una casa temporanea, ma un luogo in cui sento di essere cresciuta e diventata la persona che voglio essere, un posto in cui ho scoperto e riscoperto me stessa, di nuovo, in un modo più maturo e in un certo senso adulto. Indipendentemente da come andrà a finire, un pezzo del mio cuore rimarrà per sempre a Torino.

Prima di tutto, ringrazio le mie cugine ingegnere, in particolare mia cugina Micaela, che ha ispirato la mia scelta di trasferirmi a Torino e in qualche modo me ne ha dato anche il coraggio, sostenendomi sotto molti punti di vista.

Ringrazio il mio padrino, Giorgio, per le infinite telefonate, per i preziosissimi consigli e per pensare a me quando io non ci penso. Sei il migliore del mondo, grazie.

Grazie alla mia amatissima famiglia della GiFra. Ognuno di voi mi ha cambiato, e sono fortunatissima a chiamarvi fratelli e sorelle, siete la mia famiglia del Nord. Non avrei mai pensato di trovare un'altra famiglia in un posto diverso, e voi avete giocato un ruolo importantissimo nel farmi ritrovare la strada di casa. La mia cara presy, Ludovica. Sei la persona più pazza, come posso fare senza scriverti in continuazione di quanto siamo sclerate? Spero il nostro rapporto rimanga sempre così! La mia dolcissima Chiara. Sei la persona più divertente che conosca e la tua empatia mi ispira ad essere una persona migliora. La mia sorellina Lidia. Tu mi hai mostrato che è possibile essere amici in famiglia, e anche quanto questo sia naturale! Sono così fortunata a chiamarti sorella e amica. Infine, a Enzo. Non ho mai avuto un fratello maggiore, e Dio sa quanto ne avessi bisogno. E poi Lui mi ha mandato te, e sono così fortunata ad averti come fratello maggiore e guida. Grazie, per questo.

Ringrazio Brunella, la mia cara coinquilina, anzi coinqui-amica. Mi hai dato letteralmente una casa, spalancandomi le porte di quella che sarà per sempre casa nostra, regalandomi un posto all'interno della nostra piccola famiglia, e non potrò mai esserti abbastanza riconoscente per questo.

Ringrazio Alessandra, la mia coinqui-amica all'attivo. Sei una delle persone più belle che conosca, paziente, empatica, gentile. Mi hai fatto da sorella maggiore avendo sempre la parola giusta per me nel momento esatto in cui ne avevo bisogno. Mi hai fatto sempre sentire una persona di grande valore, hai creduto in me sempre, durante ogni esame e colloquio. Sono grata di averti nella mia vita.

Grazie alle mie amiche: Anna, Antonella, Camilla, Dalila, Domenica, Martina e Roberta. Vi voglio un bene infinito, e passare del tempo con voi è sempre tempo speso bne. Avervi nella mia vita significa sorridere e ridere ogni volta che siamo insieme. Grazie per ogni chiacchierata, ogni singolo momento di ogni volta che torno a casa, grazie per farmi sentire come se nulla fosse cambiato, anche se mi sono trasferita. Grazie perchè, nonostante la lontananza, mi fate sentire sempre accolta e a casa.

Grazie a Dalila. Più che amica, dovrei definirti sorella. Sei parte della mia famiglia e quasi tutti i più bei ricordi che ho della mia vita dall'età dell'infanzia li ho condivisi con te. Sei una certezza, e ovunque ci sei fai di quel posto casa, grazie.

Grazie a Martina. Mi hai aiutato nei momenti più duri della mia esistenza, ricordandomi costantemente la persona che sono, vedendo sempre in me qualcosa di buono, qualcosa di bello da salvare, credendo in me sempre. Sei la persona che meglio mi comprende, anche meglio di me, e sarò sempre grata per te.

Ringrazio le mie nonne. Siete le mie mamme al quadrato, vi adoro e siete il motivo principale per cui prendo l'aereo per tornare qua col sorriso a 57 denti. Vedere l'orgoglio nei vostri occhi per i miei successi è una delle luci della mia vita. Mia nonna Zina, di cui vorrei poter avere un giorno la capacità di non perdersi mai d'animo. Mia nonna Carmelina, di cui vorrei sempre avere l'empatia e la parola giusta sempre, per tutti. Non potrei essere più fortunata.

Infine, i ringraziamenti più banali forse, ma quelli più veri.

A mia sorella, Irene, luce principale della mia vita e mia grande speranza, a cui dedico ogni cosa bella che ho, nella speranza di vederti felice e soddisfatta della tua vita e delle tue scelte almeno tanto quanto lo sono io adesso. Sei i miei sorrisi più veri.

A mio padre, Ignazio. Sei la persona più intelligente che io conosca, e la persona che mi spinge a superare ogni mio limite. E' grazie alle tue metafore sui fuorigioco e i calci d'angolo che mi aiuti a venire a capo dei maggiori dilemmi della mia vita. Grazie per renderli così semplici.

A mia madre, Laura. Non sei solo mia mamma, sei mia sorella e la mia amica, sempre pronta a venirmi in soccorso e ad aiutarmi, mettendomi sempre al centro.

Mamma, papà, sono così grata a Dio di avermi fatto venire al mondo nella vostra famiglia. Voi tre siete la mia roccia, il mio scudo e la mia fortezza, il motivo per cui tutto questo sogno assurdo e incredibile esiste ed è vero. Avete creduto in me senza mai vacillare un secondo, mi avete lasciato il mio tempo e il mio spazio e mi avete compreso, senza avere pretese su di me. Questo non è scontato, neanche in famiglia. Non vi sarò mai riconoscente abbastanza.

Ho dedicato il mio precedente lavoro di tesi a mio nonno Nino. Lui mi ha insegnato l'orgoglio e la forza di combattere per i propri obiettivi. Quando tornavo dopo un esame, immediatamente voleva sapere come fosse andato e, sebbene a volte non era tanto contento del voto, neanche provando a nasconderlo, non ha mai smesso di credere che io fossi la numero 1. Non ha avuto l'occasione di vedermi incoronata, ma io non ho smesso di sentire la sua presenza, neanche per un instante, e so che è qui con me adesso e che sta scoppiando di orgoglio per la sua piccola Oreficina.

Dedico questo mio lavoro di tesi a mio nonno Ignazio. Lui mi ha insegnato

che nessuno ti regala niente, che tutto quello per cui lavori è tuo e che se anche in un certo momento nessuno ti riconosce il tuo merito, il tempo è in grado di regolare i conti e di far venire i nodi al pettine, basta avere solo un po' di pazienza. Mi ha insegnato anche che la cosa più importante è la famiglia, e la tenerezza e la dedizione con cui si è dedicato a me nel corso della mia vita la custodisco nel mio cuore con grande affetto e gratitudine.

Vorrei poter concludere questi ringraziamenti dicendo "e infine, grazie a me, perché ce l'ho fatta". La verità è che nessuno si salva da solo: quello che ha salvato me è stato avere accanto persone che hanno creduto in me, anche e soprattutto nei momenti in cui io non ci ho creduto. Senza di voi, senza i vostri incoraggiamenti, senza il vostro ricordarmi che sono una persona che vale, avrei mollato al secondo giorno di università (chi c'era il primo anno di triennale sa). Quindi brindo a voi, che fate parte della mia vita, ai vostri successi e vi ringrazio per avermi donato il mio sogno: essere ingegnera, ma più di tutto, essere la persona che sognavo di diventare.

Bibliography

[1] Shelli R. Kesler et al. *Changes in frontal-parietal activation and math skills performance following adaptive number sense training: Preliminary results from a pilot study.* DOI: `https://doi.org/10.1080/09602011.2011.578446`.

[2] Gavin R. Price and Daniel Ansari. *Dyscalculia: Characteristics, Causes, and Treatments.* DOI: `http://dx.doi.org/10.5038/1936-4660.6.1.2`.

[3] K. Kucian - U. Grond - S. Rotzer - B. Henzi - C. Schonmann - F. Plangger - M. Galli - E. Martin - M. von Aster. *Mental number line training in children with developmental dyscalculia.* DOI: `10.1016/j.neuroimage.2011.01.070`.

[4] Anna J. Wilson - Stanislas Dehaene - Philippe Pinel - Susannah K Revkin - Laurent Cohen and David Cohen. *Principles underlying the design of 'The Number Race', an adaptive computer game for remediation of dyscalculia.* DOI: `10.1186/1744-9081-2-19`.

[5] Anna J. Wilson - Susannah K Revkin - David Cohen - Laurent Cohen and Stanislas Dehaene. *An open trial assessment of 'The Number Race', an adaptive computer game for remediation of dyscalculia.* DOI: `10.1186/1744-9081-2-20`.

[6] Manuela Piazza - Vito De Feo - Stefano Panzeri and Stanislas Dehaene. *Learning to focus on number.* DOI: `https://doi.org/10.1016/j.cognition.2018.07.011`.

[7] Saedaa Khanum et al. *Effects of Non-Symbolic Approximate Number Practice on Symbolic Numerical Abilities in Pakistani Children.* DOI: `10.1371/journal.pone.0164436`.

[8] Nastasya Honoré and Marie-Pascale Noel. *Improving Preschoolers' Arithmetic through Number Magnitude Training: The Impact of Non-Symbolic and Symbolic Training.* DOI: `10.1371/journal.pone.0166685`.

[9] Marisol Cueli et al. *Influence of initial mathematical competencies on the effectiveness of a classroom-based intervention.* DOI: `10.1111/bjep.12239`.

[10] Paloma Gonzalez-Castro - Marisol Cueli - Debora Areces and Celestino Rodrıguez. *Improvement of Word Problem Solving and Basic Mathematics Competencies in Students with Attention Deficit/Hyperactivity Disorder and Mathematical Learning Difficulties.* DOI: 10.1111/ldrp.12106.

[11] Maria Chiara Passolunghi and Hiwet Mariam Costa. *Working memory and early numeracy training in preschool children.* DOI: 10.1080/09297049.2014.971726.

[12] Jörg-Tobias Kuhn and Heinz Holling. *Number sense or working memory? The effect of two computer-based trainings on mathematical skills in elementary school.* DOI: 10.5709/acp-0157-2.

[13] *Dyscalculia: Know the Signs.* URL: https://www.healthline.com/health/dyscalculia.

[14] T. Navagalli and P. Fidelis Pauline Juliet. *Technology for Dyscalculic Children.*

[15] Liane Kaufmann - Guilherme Wood - Orly Rubinsten and Avishai Henik. *Meta-Analyses of Developmental fMRI Studies Investigating Typical and Atypical Trajectories of Number Processing and Calculation.* DOI: https://doi.org/10.1080/87565641.2010.549884.

[16] Manuela Piazza - Andrea Facoetti - Anna Noemi Trussardi - Ilaria Berteletti - Stefano Conte - Daniela Lucangeli - Stanislas Dehaene and Marco Zorzi. *Developmental trajectory of number acuity reveals a severe impairment in developmental dyscalculia.* DOI: 10.1016/j.cognition.2010.03.012.

[17] Elisa Castaldi - Anne Mirassou - Stanislas Dehaene - Manuela Piazza and Evelyn Eger. *Asymmetrical interference between number and item size perception provides evidence for a domain specific impairment in dyscalculia.* DOI: https://doi.org/10.1371/journal.pone.0209256.

[18] Laura A. Outhwaite - Marc Faulder - Anthea Gulliford and Nicola J. Pitchford. *Raising Early Achievement in Math With Interactive Apps: A Randomized Control Trial.* DOI: http://dx.doi.org/10.1037/edu0000286.

[19] Magda Praet and Annemie Desoete. *Enhancing young children's arithmetic skills through non-intensive computerised kindergarten interventions: A randomised controlled study.* DOI: http://dx.doi.org/10.1016/j.tate.2013.12.003.