Politecnico di Torino

DEPARTMENT OF COMPUTER ENGINEERING, CINEMA AND MECHATRONICS

MASTER'S DEGREE IN CINEMA AND MEDIA ENGINEERING

DEVELOPMENT OF AN AI GAME FOR CHILDREN WITH DYSCALCULIA

Brugo Gaia
278146

Thesis Coordinator
Prof. Andrea Sanna

Thesis Advisor
Prof. Vito De Feo

Accademic Year 2021- 2022

To my beautiful family

ABSTRACT

Dyscalculia is a Specific Learning Disorder that affects the ability to attribute quantitative meaning to numbers and mathematical concepts. In fact, dyscalculic children have problems learning what are the mechanisms of calculation and counting. A part of the scientific research that deals with the study of this disorder is focused on finding ways to train dyscalculic children, in order to improve their mathematical calculation skills, and to reduce the difficulties that may emerge in many aspects of their daily lives. This project aims to design a videogame for children with numerical difficulties to train them and try to prevent a future diagnosis of dyscalculia and reduce the possible effects that this disability could have. It is based on the results of a study conducted in 2018 by Manuela Piazza, Vito De Feo, Stefano Panzeri, and Stanislas Dehaene that found alternative hypotheses on possible mental models regarding numerical abilities. It introduces an alternative to the classic idea that with age and education, children have a progressive refinement of the representation of the number. It seems, indeed, that young children may fail in counting also because they are influenced by non-numerical variables, such as the size of the items to be counted. The non-numerical variables seem to interfere with the sense of number leading to a wrong count. With growth and education, children learn to focus on number filtering out any interference from non-numerical variables. It seems that dyscalculic children struggle to improve this filtering ability when they grow up and go to school. The aim of this thesis project is to use these new discoveries to design a game application in which a child can train and improve their filtering ability. The game will be based on an AI software that will assign counting tasks to the player and, analyzing their performance in doing the tasks, will train them in filtering out non-numerical variables and achieving better and better numerical skills. The game is designed to be easy to interpret for preschoolers and, hopefully, it will help in the prevention and, maybe, also in the treatment of dyscalculia.

CONTENTS

INTRODUCTION

Mathematical skills are, nowadays, of extreme importance in different aspects of life, not only at school or in the workplace, but also in simple daily life. Children from early childhood show that they have conception of number and numerical skills. Research proposes that the sense of number in the infant is based on two systems: the first that considers only small numbers and traces them perceptually, the second that approximates the number system to represent the largest numerosities. [1]
As these systems grow, they also evolve by increasing both the numerical conception of multiple numbers and the linear mapping of the sense of number. Thanks to study and education they increase and refine these brain mechanisms by refining knowledge and mental representations of the number. However, all this is delayed or very tiring for some individuals who are suffering from a disorder called Dyscalculia.

1.1 SPECIFIC LEARNING DISORDERS

The Specific Learning Disorders, shortened with the acronym SLD (in italian DSA), are neurodevelopmental disorders, in other words they have their beginning during the period of the development. They are specific because there is a difference between the general intellectual functioning and school skills. The former, in fact, turns out to be in the norm, while the latter are significantly below.

They are, therefore, neurobiological disorders that persist throughout life. They impair the ability to read, write, calculate, listening and verbal expression. These disorders are:

- Dyslexia- disability of reading skills
- Disortography- disturbance of spelling skills

- Dysgraphia- disorder of writing ability
- Dyscalculia- disturbance of the ability of number and calculation

The Specific Learning Disorders are not a disease but a different functioning of the brain, which requires longer times and with more attention effort to perform a certain skill. All of this is innate and accompanies the individual throughout life. So those who suffer from it can never heal permanently but the difficulties can be compensated with time and a good strengthening activity.

1.2 DYSCALCULIA AND ITS CAUSES

Developmental Dyscalculia (DD) is a disorder that affects the understanding of quantities, the ability to recognize numerical symbols, the ability to perform basic arithmetic operations. Diagnosis of DD is recommended by the 'Diagnostic and Statistical Manual of Mental Disorders' 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]

Studies illustrate that it is also present a distinction between two categories of DD referred to as 'primary dyscalculia' and 'secondary dyscalculia'. While the primary is related to impaired growth of the brain structure that processes numerical information, the secondary refers to mathematics deficiency which originates from exterior factors such as deficient education, behavioral attention problems, and low socio-economic status. [1]

Scientific research on this disability is constantly improving and is not very thorough with compared to dyslexia. So the causes of this disorder are multiple and can be found both from a neurological and a social point of view. In fact, several researchers have shown that dyscalculia could also develop due to hereditary family factors. DD may be related to other specific learning disorders, described above, or may be associated with attention deficit hyperactivity disorder. Finally, factors such as brain injuries that occurred through accidents can also result in a type of acquired dyscalculia. The latter does not fall under the SLD since it has not had an evolutionary onset but it is linked to a brain trauma or some kind of disease.

1.3 SYMPTOMS OF THIS DISABILITY

People with dyscalculia have some or all of these signs:

- Difficulty working with numbers
- Confused by math symbols
- Difficulty with basic facts (adding, subtracting, multiplying and dividing)
- Often will reverse or transpose numbers (36: 63)
- Difficulty with mental math
- Difficulty telling time
- Difficulty with directions (as for playing a game)
- Difficulty grasping and remembering math concepts
- Poor memory for layout of things (for example, numbers on a clock)
- Limited strategic planning skills (like used in chess)
- Relies on tangible supports such as fingers, tally marks
- Slowness in given answers to math questions
- Difficulty with estimation and approximation
- Difficulty finding different approaches to one problem
- Trouble with visualizing patterns, different parts of a math problem, or identifying critical information needed in problem solving
- History of academic failure contributing to the development of learned helplessness in mathematics [3]

Dyscalculia, therefore, has a big impact on the patient's mathematical skills. The sense of number and the interference between the numerical and non-numerical dimension may not be properly analyzed by the child. In dyscalculics the so-called "sense of number", that is, the ability to quantify and numerically estimate objects in everyday images or scenes, is severely impaired. This also greatly affects tasks that involve manipulating symbolic numbers, such as arithmetic operations. [4]

1.4 TRAINING TO IMPROVE NUMERICAL SKILLS

A part of the scientific research that deals with the study of this disorder is trying to find ways to train dyscalculics, in order to improve their mathematical calculation skills, and to reduce the difficulties that may emerge in many aspects of their daily lives. Children with

dyscalculia than their peers need more tools that can help them learn and understand numerical and arithmetic concepts of mathematics. That's why using tablets or computers with applications aimed precisely at improving and consolidating these aspects can help and speed up learning. Electronic devices are also very useful because they have the ability to analyze in real time the progress made by children and this is an important role to understand the right rehabilitation method to reduce this disability.

1.5 USING TECH APPS WITH KIDS

Video games nowadays are used by many individuals already in childhood, with technological progress children, in fact, are increasingly flooded with electronic feedback everywhere both at home and outside. This leads to continuous research and creation of better apps for children related to education. Educational apps therefore offer the usefulness of using targeted and personalized processes for improvement in the area of belonging; in fact, apps with gamification principles capture the user who pushed to overcome the levels of increasing difficulty improves and learns. Another important aspect is that touch-screen tablets are much more recommended among all electronic objects for learning in early childhood children because they are lightweight, easy to carry and do not need particular outputs, like keyboards and mouses.

A study conducted recently in England [5] shows just how the use of tablets with mathematical apps inside has achieved significant improvements in learning. In this study, children divided into three groups were tested: the first group uses the app in addition to normal math lessons, the second group uses them only during school hours, and the last group did not use them. The results showed significantly better learning gains in children in the first two groups than in children who followed only the standard teaching method.

These data are very reassuring also for the technological process of prevention and improvement in individuals suffering from some learning disorder especially because thanks to all this it can be create ad hoc paths much faster by analyzing response after response while the child has fun.

THEORETICAL MODEL

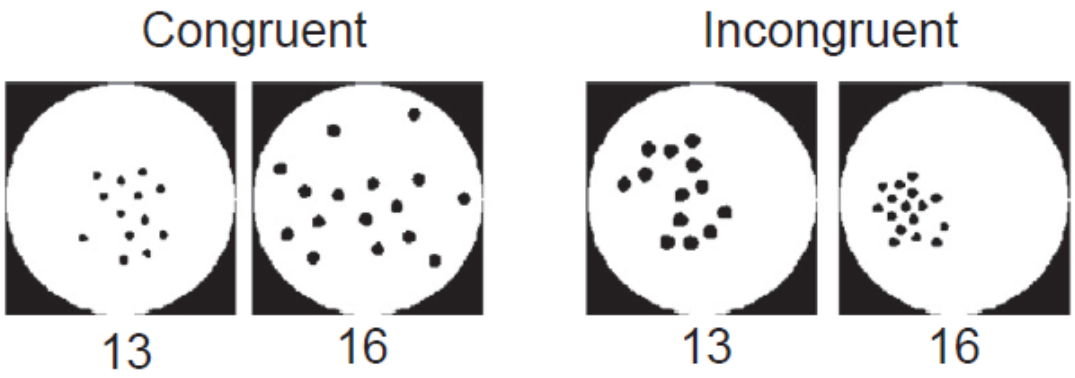
The prject is based on research conducted by M. Piazza, V. De Feo, S. Panzerid and S. Dehaene and published in 2018. In this study, the researchers demontrate that with age and education there is an increase of the ability to focus on number and filter out potentially interfering information on the non-numerical dimensions, called by their filtering hypothesis. [6] This study also showed how individuals with dyscalculia, who had the same development as healthy subjects, have problems with filtering between numerical dimensions and other dimensions. This can be used as a basis for diagnosing this disability already in preschool age.

2.1 THE EXPERIMENT

The test was carried out by 156 people (44 Italian kindergartens; 29 Italian school age children; 20 Italian educated adults; 25 Italian dyscalculic children and 38 Mundurucú, children and adults, that is a population with low computing capacity) and it provided a thorough insight into the possible cause of this condition.

Participants were given different sets of numbers represented with dots and asked to select which set was bigger in number without counting them. In fact, they had to choose in a limited period of time so they did not get the chance to count. As the test progressed, both the number of points and other non-numerical variables, such as the size of the dots, were changed. This produced what the researchers labeled as “congruent and incongruous” trials. In congruent trials the set that has the largest number also has the largest value of the non-numerical variables and therefore with a greater value while the incongruent trials were the opposite with the series with the smallest number that always had one or more non-numerical variables with a greater value. The following figure

provides a visual representation of these different types of tests.



2.1.1 An example of Congruent and Incongruent Trials [6]

The researchers defined the stimuli space using one numerical variable, representing the numbers of dots, and four non-numerical variables (all expressed in terms of number of pixels), which univocally define number:

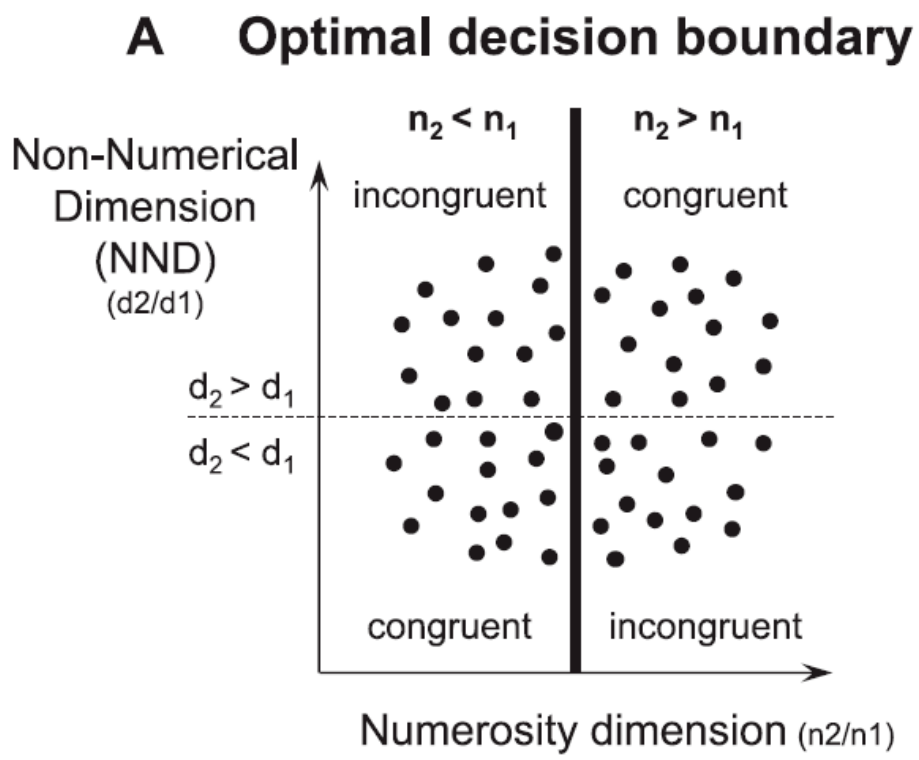
- Item Surface Area, also known as ISA, defined as the area (in terms of number of pixels) occupied by a single dot;
- Total Surface Area, also known as TSA, defined as the item surface areas multiplied by the number of items;
- Field Area, also known as FA or "convex hull", which indicates the portion of the space where dots actually fall into;
- Sparsity, also known as SPARS, which gives us information about how sparse the dots are inside their area, in other words it's the field area divided by the number of items.

For analysis purposes, these four non-numerical variables were merged using the principal component analysis into a single summary measure called 'Non-Numerical Dimension (NND) of the stimulus space hosting the sets. [6]

2.2 THEORIES USED FOR THE ANALYSES

The researchers analyzed the data considering that the normal progression in a person's decision-making process improves with growth, arriving in adulthood to differentiate small numerical changes of 15-20%. In the following figure each dots represents a single trial that is a combination of two numbers and their relative non-numerical characteristics. The horizontal axis represents the logarithmic ratio of the numbers while the vertical one represents the logarithmic ratio of the non-numeric variables and the vertical line

indicates the “Optimal Decision Boundary”. It represents a line in a two-dimensional space in which are located, as shown in the figure, the dots that are part of two distinct classes and exceeding this line the class will be changed. Each point within this graph, and the following ones, represents a trial that took place within the experiment. The first class consists of all the points where the major numeric variable is present in the first area, $n_2<n_1$, and the second class has within it all the points where the largest numeric variable is present in the second area, $n_2>n_1$. This boundary is defined as optimal when the decision boundary perfectly divides the samples of the class on its left with the samples of the class on its right. If the points, present in the two-dimensional space, are close to the line, their numerosities varies rarely; as these points move away, both to the right and to the left of the line, their distance increases.



2.2.2 Graph representing Optimal Decision Boundary [6]

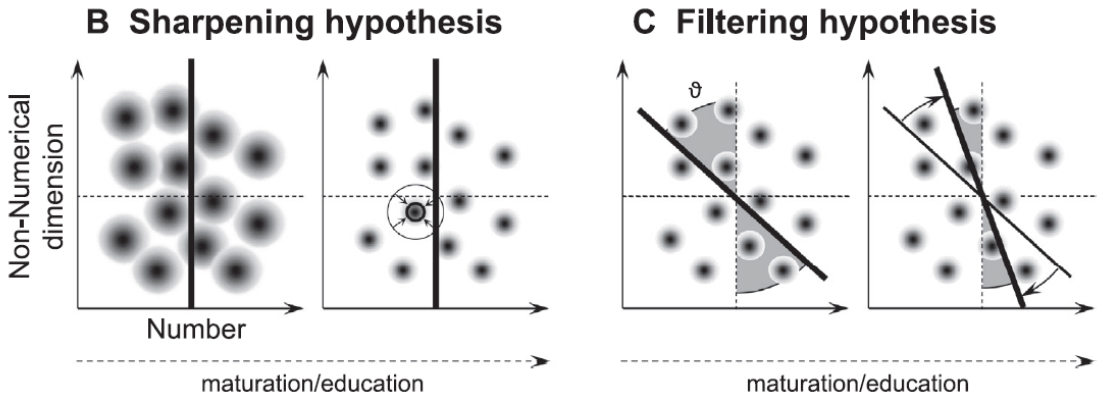
In this study, two hypotheses were also considered: the sharpening hypothesis and the filtering hypothesis.

- The sharpening hypothesis holds that with maturation and the educational process the mental representation of numbers is sharpened, in fact with the growth the visual acuity of the child with respect to the numerical difference becomes increasingly clear

and less blurred;

- The filtering hypothesis, on the other hand, proposes that already at an early age children take into account both numerical and non-numerical variables, but as they become adults, their decision-making system learns to focus on numbers and neglects intrusion from non-numerical variables.

The next figure shows both graphs that relate to both hypotheses. In these panels the width of the dots represents the noise of internal representations of number and of another non-numerical dimension (NND). [6] The sharpening model shows that the representations are initially very noisy, but then become sharper with age and education. The filtering mechanism holds that the change does not lie in numerical representation, but in the ability to gradually learn to ignore NND variables and focus only on numbers.



2.2.3 Graphs representing Sharpening and Filtering Hypothesis [6]

As can be seen in the sharpening model the Optimal Decision Boundary remains constant at every stage of life, while in the filtering model it shows a gradual rotation of the decision border towards the optimal line that leads to a constant reduction of the angle (ϑ) between the decision and the optimal slope that makes the decision-making process of the person more accurate. [6]

If the sharpening hypothesis were accurate and unique by growing and improving one's visual acuity, there would be an improvement in both congruent and incongruent trials and would not differ in anything while studies show that it improves much more in incongruents. That is why the filtering hypothesis must be taken into account and is the one that most characterizes the development in the child, since the presence and

development of dyscalculia leads to a disability does not allow to focus only on the relevant size, that is, the number.

2.3 ANALYSES AND RESULTS

The researchers at the end of the tests analyzed separately the results of the experimet in which there was congruence between the non-numerical and the numerical dimensions and the incongruent ones. Congruent trials achieved a greater overall magnitude in all experimental groups.

The data was then analyzed through two methods: the one based on logistic regression and the one that uses Shannon information. The first, based on model, indicates how much the choice between right and left, that is which is the right set, depends on all the variables in play; the second, without a model, uses mutual information [7] by creating a more general approach to estimate the effect of the number and the non-numerical variables. All this was done to better understand the relative weight of each variable. The results distinguish the effect of age from that of education, and the latter was noted to play an important role.

Usually education, in fact, will lead in a reducing of the effect that non-numerical characteristics have on decision-making. The results of the data analysis then validate the filtering hypothesis, showing that the improvement in the comparison of the number is based on a greater ability to focus on the number than the other factors.

2.4 APPLY THIS KNOWLEDGE TO THE PROJECT

Using the filtering hypothesis of this research, the project will be based on the experimental activity using a game application. The goal is to use it to specifically train children to distinguish differences in sets of numbers while ignoring non-numerical variables.

Thanks to these trails, the child's mind is trained to better ignore irrelevant external factors and pay attention only to the numerical quantity. In addition, the game will have a built-in artificial intelligence that will evaluate the child's performance after providing several

random tasks with different levels of difficulty.

By analyzing the responses, the AI agent will model the part of the child's brain responsible for counting numbers with new and more targeted tasks. The purpose of this modeling is to understand the level of mathematical intelligence that the child possesses to improve their filtering. The game purposes to bring the decision-making boundary towards the most optimal position possible thanks to targeted trials that it can propose to the child and, in doing so, helps to correct the onset of dyscalculia in advance.

STATE OF THE ART TECNOLOGY

Nowadays children are increasingly fascinated and surrounded by technology; in fact, from an early age they interact with it. For this reason, the researchers thought they could engage dyscalculic children through video game apps rather than traditional methods, such as books and simple exercises. These applications are intended to try to improve and consolidate the numerical and arithmetic consciousness in these children. Scientific research is always evolving and apps will have to update on new discoveries in this area. In this chapter three gaming apps will mainly be described since these apps, although with some limitations, are based on published research and scientific studies compared to other applications on this topic that do not base their roots on tested experiments. So, to help small people suffering from dyscalculia, these three gaming apps were created:

- The Number Race
- The Number Chatcher
- Rescue Calcularis.

The first two were created by INSERM-CEA Cognitive Neuroimaging Unit, a world-leading research institute French in mathematical cognition. [8] [9] These two apps are one the continuum of the other since the first focuses on smaller single-digit numbers and the second focuses on larger two-digit numbers. The latest app analyzed was created by Swiss researchers. [10]

3.1 THE NUMBER RACE

This game is designed primarily for children aged 4 to 8 years, who are coming into contact with numbers and mathematics. For younger children the game will teach the

basic concepts of number and arithmetic; instead, for older ones it will allow to quantify the sense of number and will practice arithmetic.

The game focuses on three macrogroups of topics: number formats, counting between numbers from 1 to 40 and addition and subtraction operations in the range 1-10.
The structure of the game has been studied both to be used to improve and empower children without specific disorders and for dyscalculic children considering that this disorder is a primary deficit that affects the associations between numerical quantities and symbols and their representations.

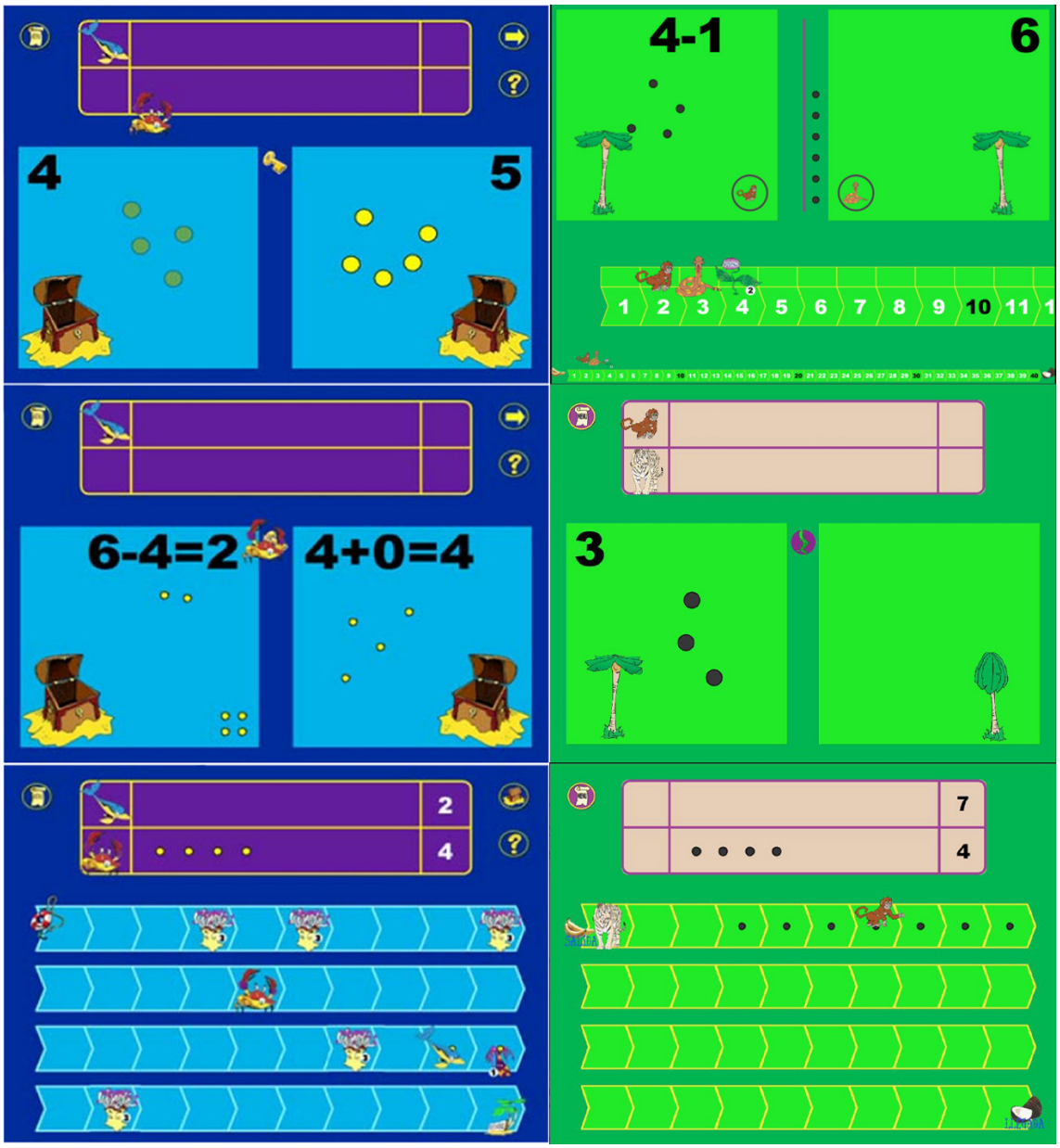
3.1.4 The Numer Race: Game Cover

The difficulty of the game dynamically adapts to the player's performance thanks to an adaptive algorithm that aims to ensure the best possible learning. The increasing difficulty is also obtained by decreasing the response time for numerical comparison or by making the quantities only in symbolic format or with the presence of simple arithmetic operations, as it is shown in ***fig. 3.1.5***. All this is always analyzed in relation to the progress and feedback of the child.

In the game there are two main screens in which children must perform certain activities: in the first they must compare between two sets which is the major, sometimes having to perform mathematical operations; in the second the child must move his/hers character and that of the opponent according to the number of coins of both, if he/she arrives first he/she receives a prize.

The application was created and tested by researchers A.Wilson and S. Dehaene [11] for

two precise purposes: the first is to test children on the sense of number, that is, the ability to non-verbally describe the number as a quantity, and the second is to depict the number itself. The game, therefore, seeks to achieve both goals through levels using numerical quantities and their spatial representation and through how non-symbolic and symbolic numerical representations are compared. The researchers tested the game on a group of nine children between the ages of 7 and 9.



3.1.5 The Numer Race: Some Game Screens

The results showed an improvement in basic number cognition and also noted that the links between number representations were consolidated, in part. There was no real

improvement in arithmetic concepts but only a better awareness of the rules. However, this study also found limitations, such as the limited number of participants and the absence of a control group.

3.2 THE NUMBER CATCHER

The target of this app is children aged 5 to 10 years, but it can also be used by adults in its highest levels. Children through the game will learn arithmetic and the concept of number. It therefore through games on the ability of calculation, such as addition and subtraction, on numerical formats, on the basic principle ten and on the logic of two multi-digit numbers allows to practice and improve knowledge on the basic concepts of number and arithmetic.

3.2.6 The Numer Catcher: Game Cover

The numbers within the game are represented with Arabic symbols, with their sound translation and are displayed as quantities, arranged on a line. The score they get is based on the time it took to complete a level and the number of mistakes they made. In this way, the game consolidates the brain connections that handle the various numerical representations.

Through continuous practice it is possible to partially rehabilitate the function of the brain circuit that dyscalculic children have. Each screen proposes to the player to fill the vehicle always in different ways and it moves the player to the next screen reaching the correct number of objects selected from the accumulated packs at the top of the screen.

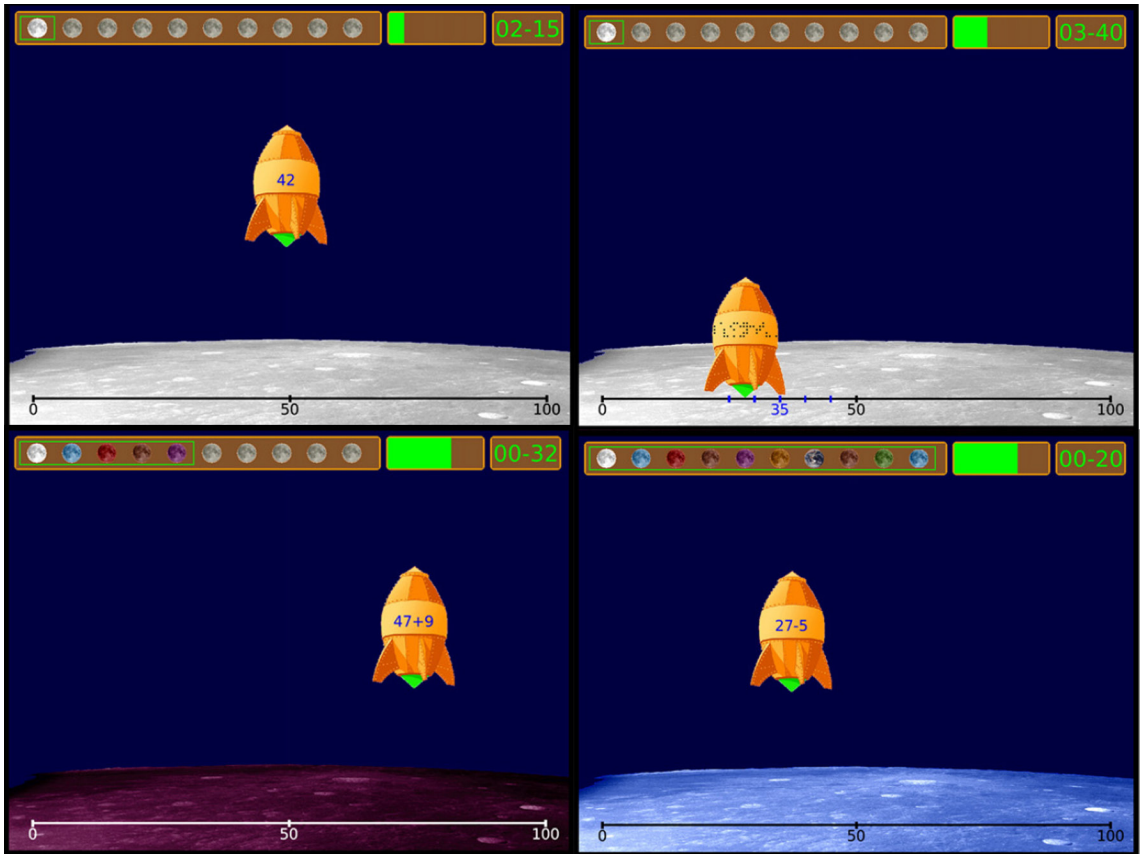
3.2.7 The Numer Catcher: Some Game Screens

3.3 RESCUE CALCULARIS

The software aims to improve the representations of the number by favoring the association between the numerical quantity and the space. In addition to this, it aims to understand the concepts of numerical ordinariness, estimation and arithmetic.

The training is integrated through a story to retain the child. The player must save his/hers home planet, called “Calcularis”, which few and limited reserves of energy. The child-astronaut flies with his/hers spaceship to the planet “Heureka” to collect the supergas but since this planet is far away he/she has to stop on ten planets to refuel his/hers spaceship. The two planets are 30 light-years apart which corresponds to 30 different levels with increasing difficulty. For each level the child must guide the spaceship to the correct position on the number line. If he/she makes a mistake, he/she will have to repeat the level until he/she solves it so as to support learning. [10]

The tests conducted showed, for both the experimental group and the control group, a refinement in indicating the correct position of a number on the number line after the game. Dyscalculic children have achieved much greater results and this is a good sign of an improvement on disability. In addition to this, the game brought a better mental representation of the number. It is possible, however, that all these results and improvements are not only due to an improvement in mathematical knowledge but also to the effect of training.



3.3.8 Rescue Calcularis: Some Game Screens

3.4 DIFFERENCES OF THESE APPLICATIONS WITH THE GAME

The previous games listed are excellent tools to refine and weaken the disorders given by dyscalculia, as shown by the tests carried out on them. However, they have some differences with the app created in this project because they do not refer to new discoveries about this disability. In fact, in the playful process of these applications the interferences that can cause the non-numerical variables on the granting of the number

itself are not refined and tested. Another discordance is the fact that they are designed to train the brains of dyscalculic children, of school age, on calculations and numbers and not for the prevention of the occurrence of dyscalculia on kindergarten children, who show early signs of dyscalculia. They are not even based on artificial intelligence and therefore cannot, by analyzing the feedback between play and child, help in the diagnosis. Last difference is that the tests carried out are not entirely satisfactory since they have not been tested on a large sample.

So to summarize the main dissimilarity of these games for dyscalculic children are mainly, compared to what the project aims to achieve:

- not neglecting and not testing the interference of non-numerical variables respect to the sense of number;
- the target audience is different because these apps are for school-age children suffering from dyscalculia and not for the prevention of its occurrence on younger children;
- the fact that they are not based on artificial intelligence;
- the sample of testers was very small and therefore the tests are not entirely sufficient.

AIMS

Mathematics characterizes, nowadays, many aspects of people's daily lives. Many sectors and activities require an intelligence and fluidity typical of mathematical processes. It is all this that can cause discomfort and frustration in people suffering from dyscalculia. Being a disorder that only in recent years is at the center of numerous research and debates, a definitive way has not yet been found to decrease the effects of this disability can have in the individual.

What this project tries to do is mainly attempt to prevent the formation of any form of dyscalculia in the preschool child. In fact, the main target of the application are children of 4/5 years who could or who begin to show difficulties in counting. The project builds on the latest findings in this area. The theory is that this disorder originates from early childhood in the child who learns to count and understand numerical quantities since there is an interference with numerical and non-numerical stimuli.In fact, if the results of 7 or 8-year-old dyscalculic children are compared with those of 3 or 4 year old dyscalculia-free children, they are the same and both groups tend to find the same problems in the same concepts.

The project, in fact, tries to train the child with congruent and incongruent tasks: the former have numerical and non-linear dimensions, the latter are the opposite and are the ones that cause the most confusion to children. With the help of a game environment suitable for the target audience and age-appropriate visual and sound feedback, the idea is to try to entertain the player as much as possible on the platform so as to allow him/her to improve and expand the filtering methodologies.

The use of artificial intelligence, within the game, is intended to provide a targeted and tailored experience for each player. It gives a continuous and constant evaluation in real time on the answers given by the child; therefore, if it notices that the player is having problems on certain trials with the interferences that non-numerical variables make to the detriment of numerical ones, it lowers the level of difficulty. In this way it allows a gradual improvement of the filtering method.

To conclude, the key problem behind this project is that dyscalculic children will never be able to heal but only improve the concepts of number and quantity; so as to be able to integrate themselves, as much as possible, into all aspects of daily life. To do this, the goal is to prevent the onset of any form and symptom related to this disorder by training children from kindergarten to recognize and focus on the quantity and numerical size as compared to any other variable.

4.1 TARGET

This project is designed to adapt to the needs of researchers to test on the filtering methodology. The target audience is mainly kindergarten children, so between 3 to 5 or 6 years who have shown initial difficulties in counting objects. Since dyscalculia is usually detected in the early years of elementary school the task of this game is to prevent its onset. That's why it caters to younger children with a graphical interface and simple narrative gimmicks made just for them. This type of target needs a simple and immediately understandable playful environment so as to be able to enjoy as much as possible the experience to improve and train the filtering between numerical variables and non-numerical variables.

It is not excluded a use to older children, attending the first years of school, to mitigate and reduce the effects of this disorder.

4.2 IDEA

Starting from the tests carried out and explained in the article "Learning to focus on number" [6] in this project a different expedient and setting has been created but at the

base of the same mental exercise.

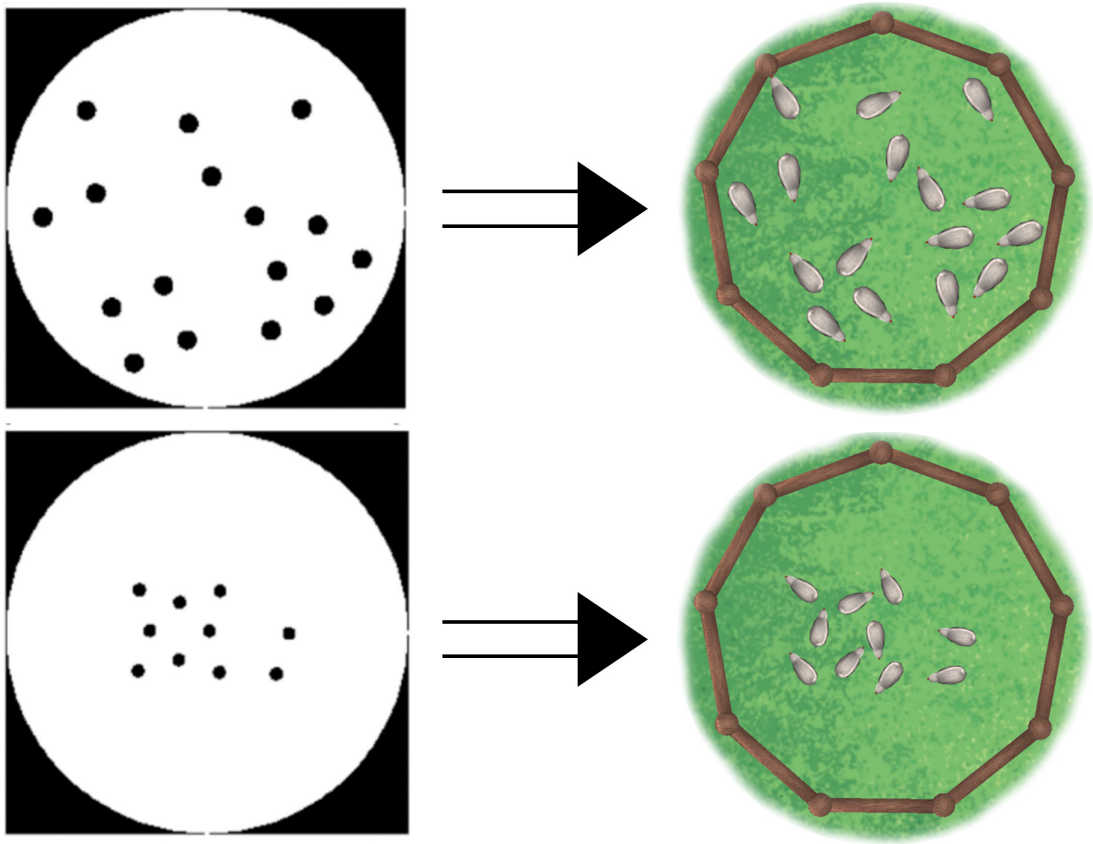
The first creator of the project Fletcher Hurn, who implemented a demo version of the application both on the server side and on the client side, conceived the farm as the main theme. This idea is a winning hypothesis given the age of the target. In fact, the typical peasant animal and plant world can be identified and recognized by young children. The idea was to transform the circles with fences and the points with animals, for example chickens, that came in to eat in them. By changing their size or that of the enclosures and the distance between the various animals, the non-numerical quantities are modified and they act as an interference, for the child, with respect to the number of animals inside the enclosure.

Beginning from the demo version, in this project, the game has been modified entirely starting from the graphic style. In the conception phase of the videogame environment, an inspirational moodboard was created with illustrations that depicted the peasant world in which the child wants to be immersed. Finally, from it a palette of colors has been created that will be found within the game.

4.2.9 Moodboard and Palette

GRAPHICS AND DESIGN

The application was created to be used by kindergarten children almost entirely individually and therefore without the help of parents or adults. This is to allow him/her a total learning from the point of view of the sense of number and the method of filtering. The game recreates the experiment described in depth in the article “Learning to focus on number”. [6]



5.10 From experiment sets to app graphics

In the game the two sets that contain the points have been transformed into fences and the points have been transformed into animals, so that a background storyline can be

devised to involve the player. In fact, every aspect of the application refers to variables and data of the starting test, as seen in ***fig 5.10***. The difference between the numerical and non-numerical variables, which in the research was characterized by the number of dots and the size of the enclosure, the points and their mutual distance, in the game is described by the number of chickens, the size of the fenced area, the size of the animals and their mutual distance.

In addition to this, graphics typical of gaming applications have been added to the game, such as the various visual and sound feedback or the pause and start menus. All these precautions were made to make the game more engaging and to allow the child to play in settings and with animals that he/she could recognize and empathize, so as to be able to continue playing with more desire and liking. This continuous focus on the game will lead him/her to an improvement regarding the filtering between the numerical component with the non-numerical variables. Through visual and sound feedback the child will understand if the choices made will be right or wrong so as to understand the mistakes and be able to improve.
Being the target so young the game is designed to be child-friendly with videos and visual content accompanied by the sounds and phrases of the farmer James who explains both the purpose of the game and in the various trials, that is the various mini levels that will follow one another within the app, he will indicate if the answer is correct or not. Only in the various menus there are the presence of writings that a child of the kindergarten could not be able to read, this is because in the game the child does not need help from older people and he/she has to make the final decision; instead, the various menus are only for the parent or those who assist the child to move within the app.

5.1 GAMEPLAY

The game is designed to transport the child into a peasant world. This environment was chosen because it is very simple for a preschooler to recognize places and characters. On this farm, farmer James has a mission for the player, to help him/her estimate how many animals enter the enclosures. In fact, the animals, in the trials represented by chickens, will enter the two enclosures and the child will have to indicate in which of the two sets there are the largest number of chickens.

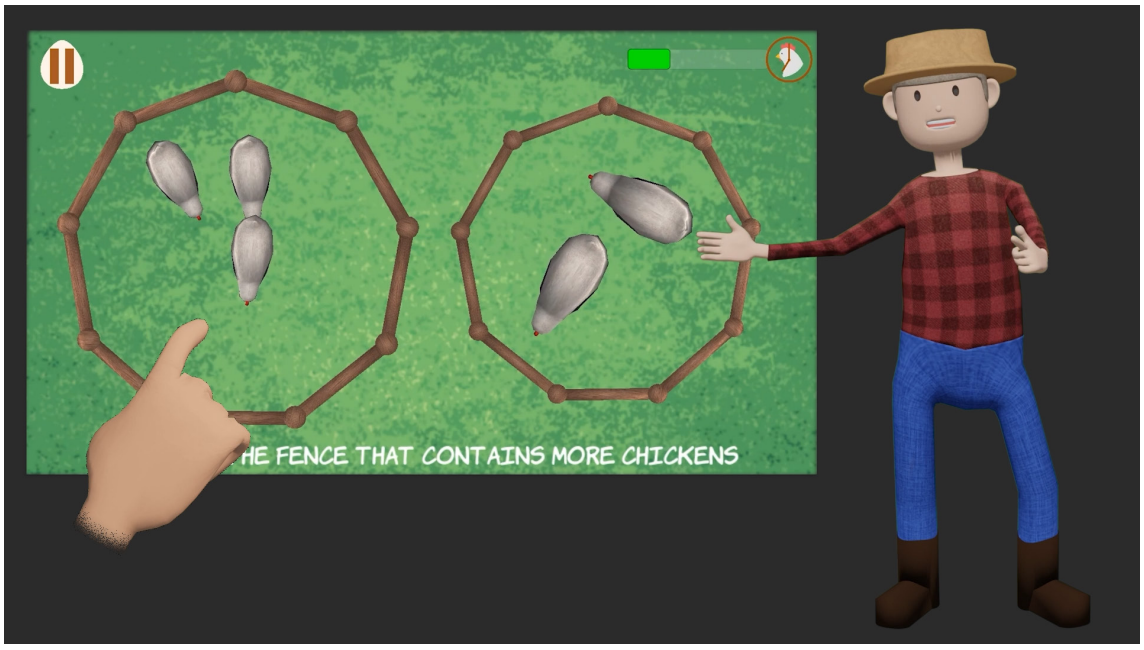
The application has been called “The Number Farm” so as to present from the beginning what the children will be tested on. In the main menu, an UI graphic is presented with the title of the application, the farm in the background and a set of random numbers to characterize and visually show the world to which reference is made, that is, calculus and mathematics.

5.1.11 Main Menu

The screen is designed to be used by the child together with the parent or caregiver. In fact, there are writings without sound feedback and not all the specific target of the game can read and therefore it needs external help.

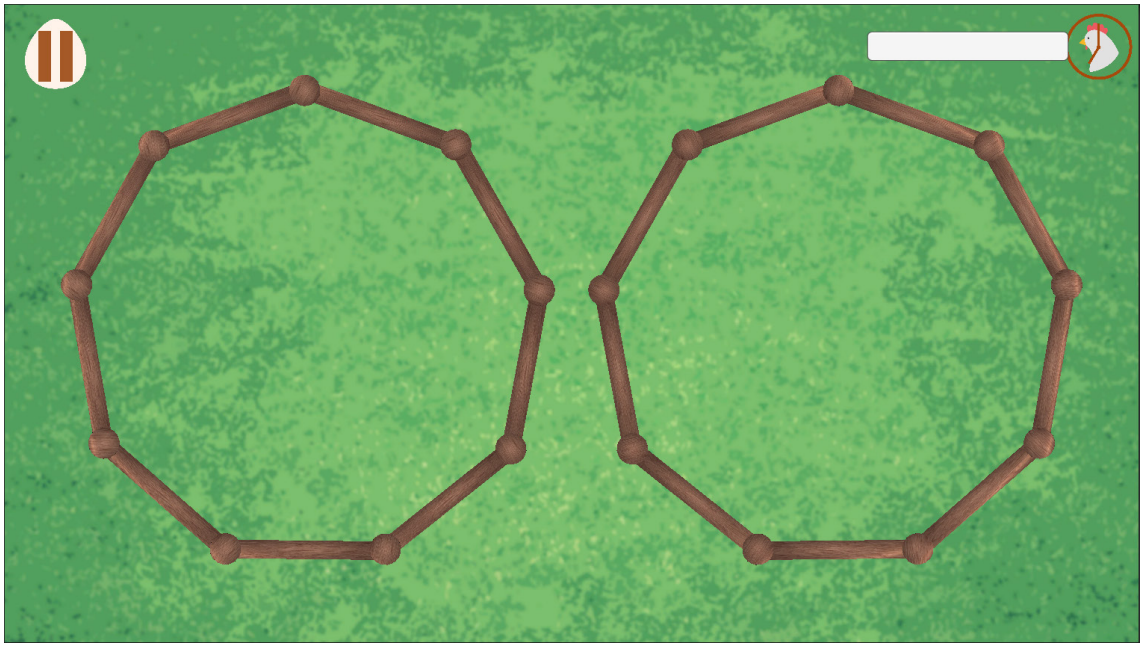
Once the play button is pressed, the role of the parent is no longer necessary, except to close the application or adjust the settings, since the playful path is characterized by visual and sound feedback designed precisely for the age of the child. Before the start of the trials, the child is shown an introductory video in which the farmer James explains the purpose of the game. In a first part of the video is shown the farm and the farmer in the foreground who introduces himself and introduces the purpose of the game, as can be seen in *fig 5.1.12*.
In the second part of the video James explains the game better and what the child must do. He shows the child that he/she will have to click in which enclosure there will be more chickens but explains that there will also be a timer that will decrease quickly.

5.1.12 Screenshot of the first part of the video



5.1.13 Screenshot of the second part of the video

In the video James' voice is very quiet and spells out all the words perfectly so as to allow the maximum possible compression to any child who approaches the app. This is precisely to let the player learn the rules of the game on his/hers own and to enter the playful environment without the help of an adult. At the end of the video, the main scene of the game will appear. In it there are no writings but it is all visual so as to be understood by all children.



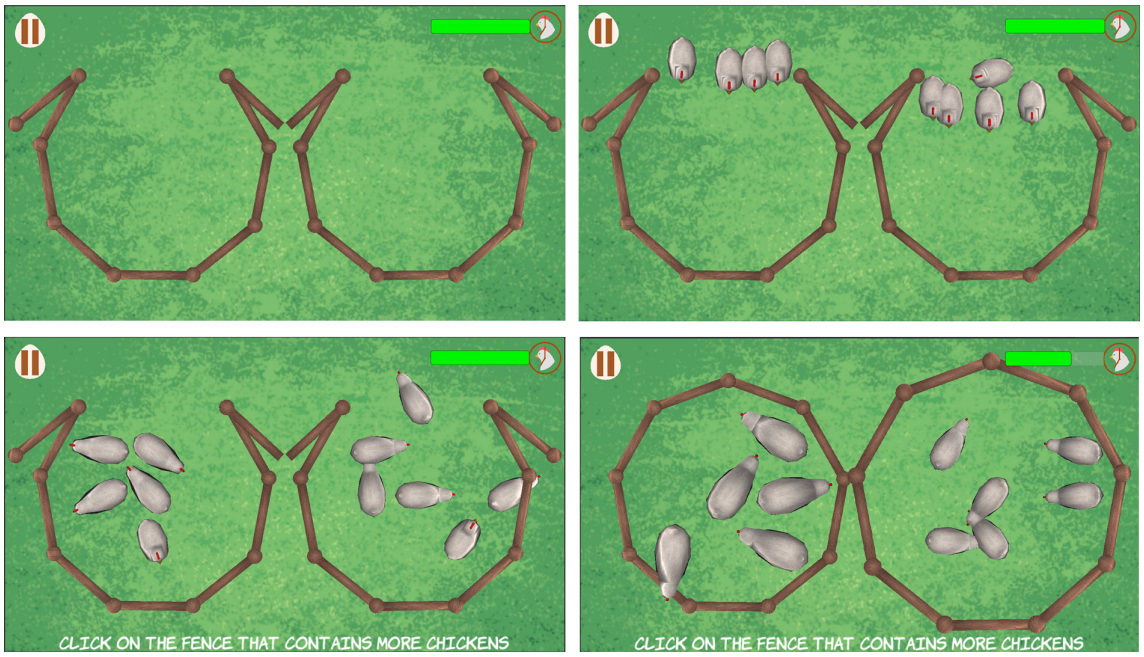
5.1.14 Main screen of the game level

At the top right there is a slider that shows the time that the child has in total to respond and will decrease with the passage of time. In fact, the child will have a maximum time in which he/she can click and respond, it is decided by the server and sent to the client along with all the data that make up the numerical variables and not put into play. This time is composed of two components: a part of the time available constitutes the time in which the chickens inside the fences will be visible; the remaining is the time left to the child to make the decision.

5.1.15 Timer

The elapsed time is represented by a slider, shown in ***fig. 5.1.15***, which decreases depending on the seconds elapsed; at first the slider is green, once the chickens are gone it will change color to yellow to visually show the almost end of the time available. All accompanied by the ticking of the timer and the change of the clock icon with the chicken and without it.

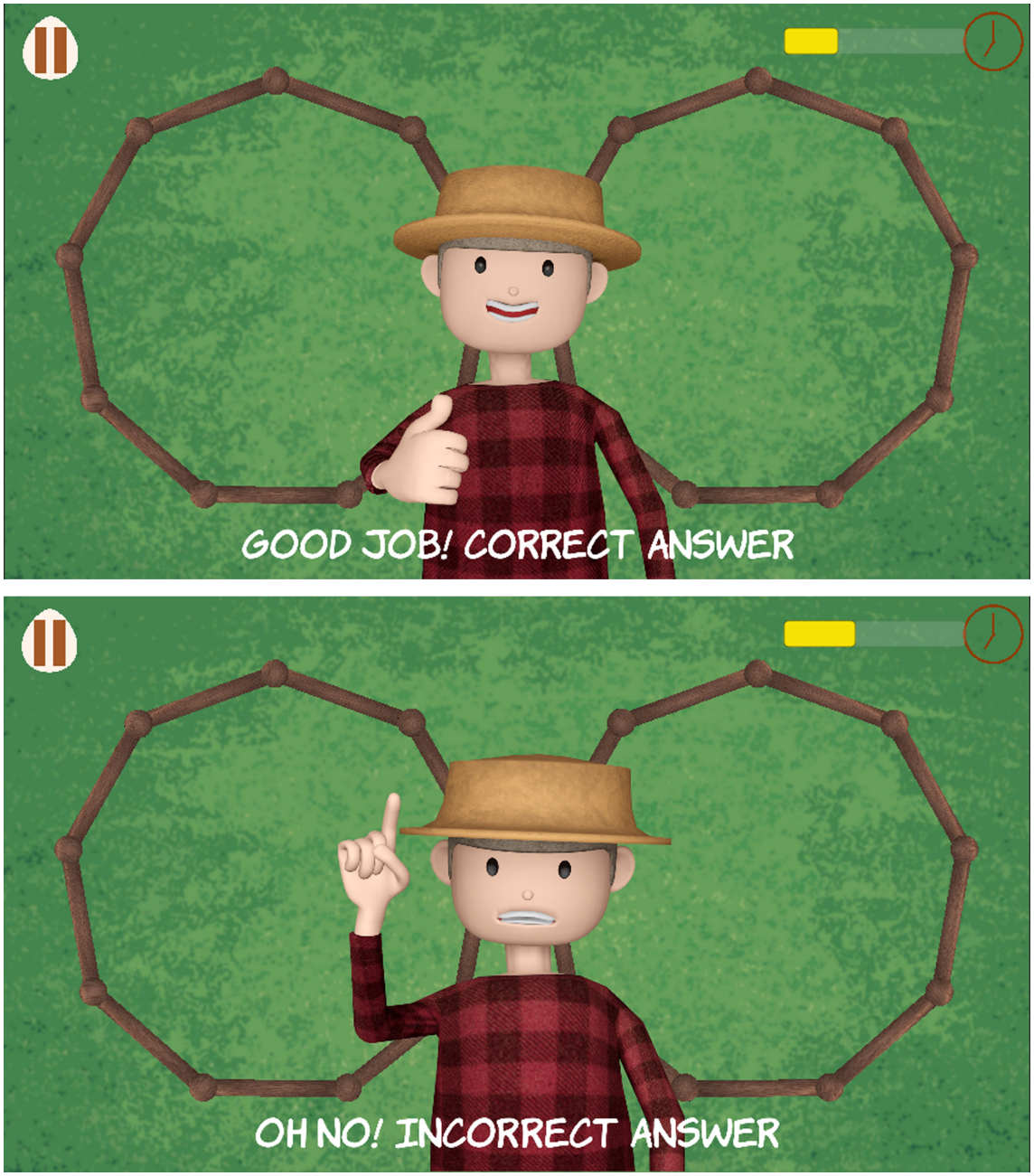
As shown in the next figure, the various levels follow one another in the game with the opening of the fences and with the entrance of the chickens from the top of the screen. The chickens are placed inside the fences and once the latter close the time to decide will begin. Immediately after the closure of the fence James will intervene by telling the child to click on which fence there are the most chickens, this phrase will be displayed through an inscription in the game to which an audio is connected.



5.1.16 Entry of chickens into the fences

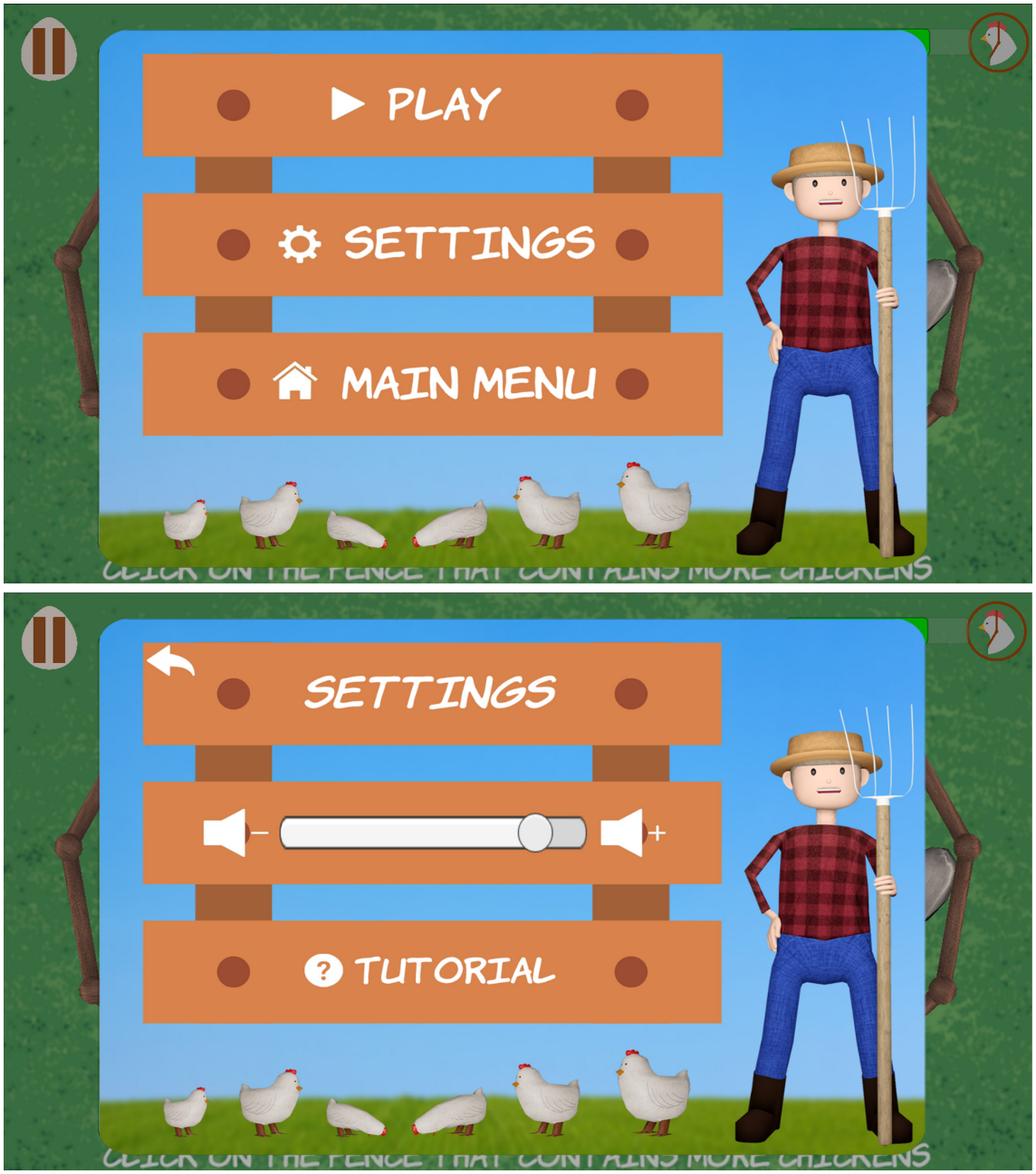
Depending on the child's response there will be two types of feedback one positive and one negative: the first corresponds to an UI with the happy farmer James complimenting the player, in the second James is sad about the wrong answer of the child.

The game will continue like this for each trial depending on the data sent by the server. To conclude, at the top left there is the pause menu button that allows the player to control the settings and to exit the level.



5.1.17 Feedback for the correct answer and for the incorrect answer

As with the main screen, the pause menu has also been designed to be used with the help of a person who can read, who is assisting the child. In addition to being able to control the general audio in the settings the player can follow the video tutorial in which James explains the rules of the game.

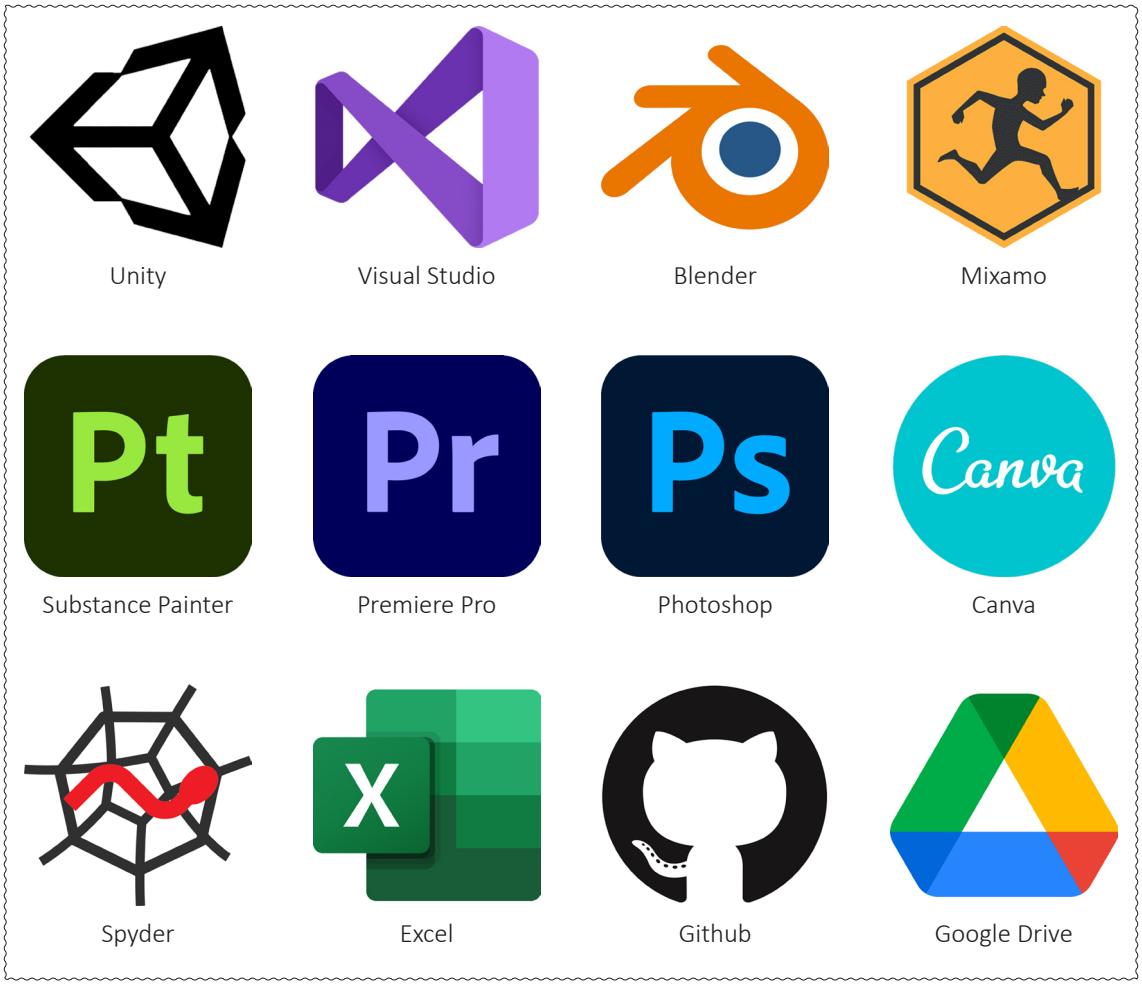


5.1.18 Pause menu

5.2 TECHNOLOGIES USED

The game was designed using the Unity game engine (version 2020.1.9f1), thanks to its highly appreciated gaming performance and compatibility with different operating systems, and using the C# programming language via the Visual Studio editor. The main character James, the animals, the objects as well as all the animations in the game were

created using the modeling software Blender 2.93. Some animations were imported from Mixamo's online library and instead the textures were created using the Substance Painter software. As for the rest of the UIs present, they were created using the programs of the Adobe suite, namely Photoshop and Premiere Pro, and through the online graphic design tool Canva. On the server side, the development took place through the Python language with the use of the Spyder editor and all the analysis and creation of the space of solutions was done through Excel. Finally, Github and Google Drive were used for the storage and sharing of the material.

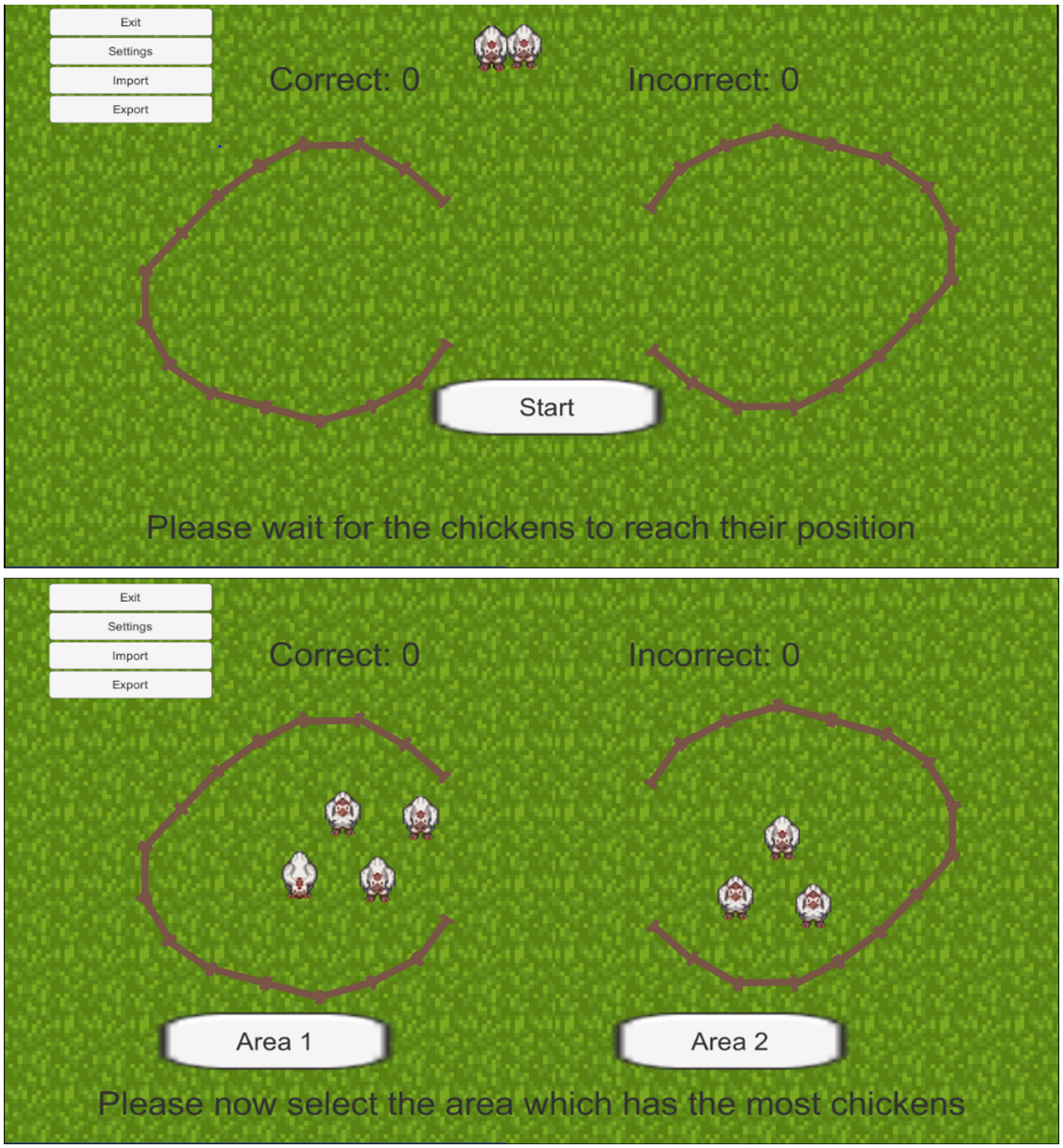


5.2.19 Logos of the software used

As for the two main software used, Blender and Unity, they were mainly used for their performance and for the possibility of not having to use particular licenses, in fact they are two free and open source software. Blender was mainly used for modeling using the main features in the edit mode and for animation using both the dope sheet and the NLA editor to organize the animations mainly of the chicken. As for the video, the graph

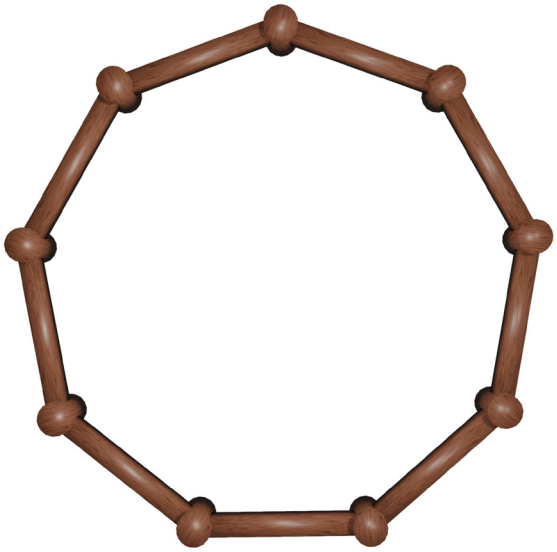
editor was also used to insert an audio and combine it with a shape key to have a fast and immediate lip sync to be inserted into the final video. Unity has been used in a complete way using all the main features. Prefabs of the various models have been created so that animations and materials used can be modified faster. The Animation and Audio Mixer panels were used to create animations and create a sound mixer and finally a special library, Sockets, was used to have the server-client connection.

5.3 GRAPHIC DESIGN

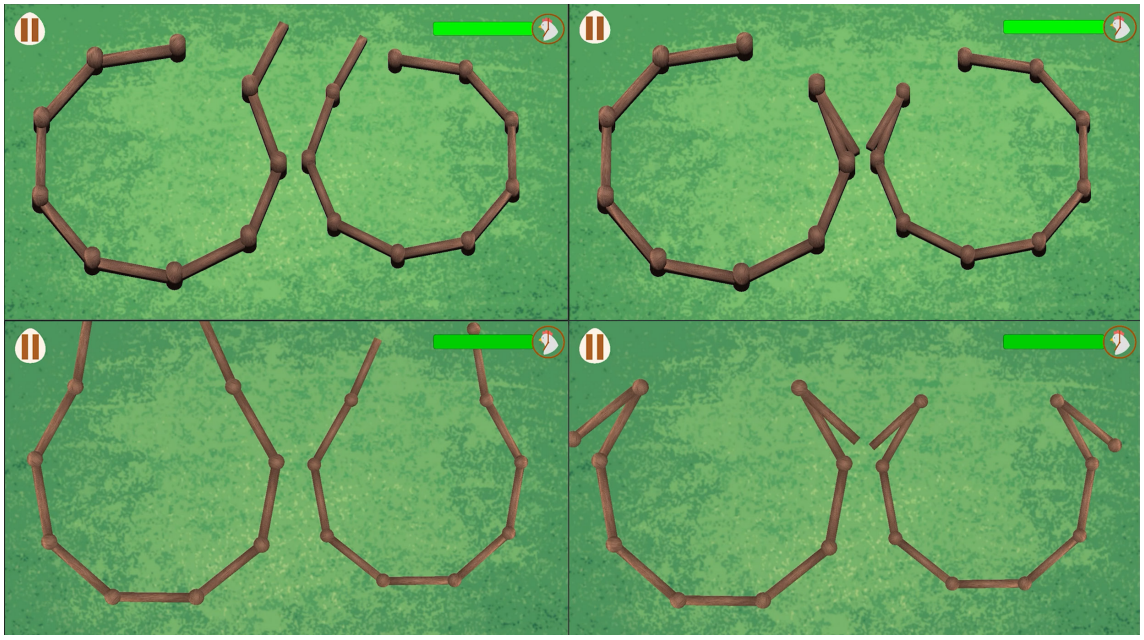


5.3.20 Demo version graphics

The game has been designed and created to have a cartoon style to make it more easily recognizable to the target of use. Starting from the design of the demo version of the game created by Fletcher Hunt using open source graphics by Daniel Eddeland for chickens [12]. Starting from this demo version, new models have been recreated for the game setting. First of all, the fence was created with an already circular shape so to have an almost perfect circumference that recalled the circles that contained the points in the original experiment.



5.3.21 Model of the fence



5.3.22 Animations of the opening of the fence

Then on Unity its opening and closing animation was realized; at first the opening of the fences took place laterally then it was chosen, in relation to the script that finds the positions of the chickens inside them, the opening from above so as to allow a fluid movement to the chickens that enter their enclosure. As well as the closing animation which is exactly the same animation only flipped.

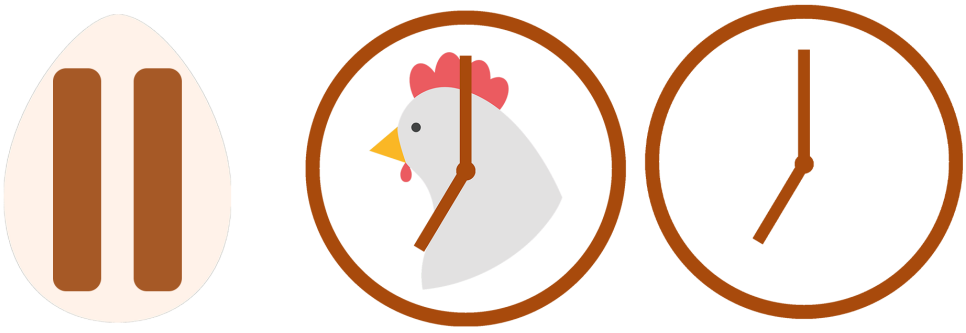
As can be seen from ***fig. 5.3.22***, the main camera was also changed during the design to the final 2D version because shadows were created on the objects and some chickens, when they reached the final positions, were covered by the fence.

After creating the fences the chicken were created, always with a cartoon design. A complete and super realisting model has not been created and with many vertices but for example the eyes and wings have been added as textures since the view of the game is from above in 2d, so it was preferred to decrease the vertices of the 3d model.
The various animations were created through Blender and once finished they were inserted on Unity. Two different animations have been made: the first of walking that is used when moving the chickens towards their final position in the enclosure and the second used once they arrive at their final position that simulates the act of pecking feed on the ground. To make these animations, an armature was manually made consisting of a series of bones: one controlled the head, five the body, one the tail and two in their respective legs.

5.3.23 Chicken armature

5.3.24 Chicken animations: idle, walking, pecking

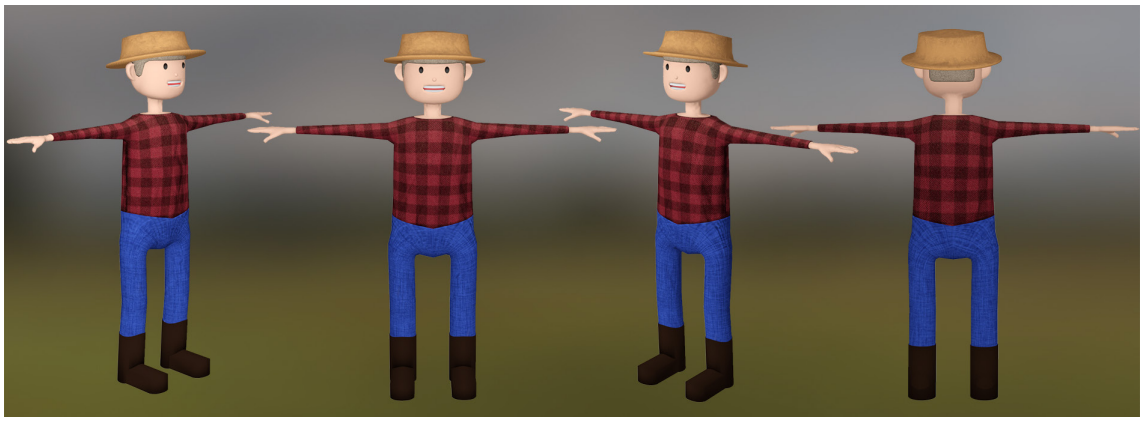
After modeling those models, the UIs that characterize the pause menu button and the timer slider were created.



5.3.25 Pause button and Timer UI

The first graphics have been created by inserting the symbol of the pause inside an egg, always in order not to lose the style of the farm and characterize even more the game graphics. The second graphic was created to represent the timer with the playing time,

so that all children could understand that the slider characterizes the passage of time available. Once the time has passed when the chickens in the enclosure can be seen, time sent by the server, the UI changes from the clock icon with the chicken to that of the clock only.



5.3.26 Some renders of James in T-pose

Once all the main graphics of the main level of the game have been created and tried to characterize the app more by creating the whole side story. Therefore the farmer James was modeled and animated to have a character relate to the child while he/she plays. The realization has tried to respect the graphic mood of the game also respecting the classic imagination of a farmer, so the checkered shirt, jeans and hat, as in the figure.

5.3.27 James armature

After modeling and texturizing the whole character was placed on Mixamo, an online library of animations and models, so as to have a skeleton inserted perfectly and to be able to access more easily some animations used in the video.

Before creating the initial video, the final UIs for the main game environment were produced. Starting from the visual feedback that appears depending on the child's response, to which two sound feedbacks are related. In the first James is happy and compliments the child; while in the second he has a sad face related to a disgruntled answer, these two UIs can be seen in ***fig. 5.1.17*** on ***page 32***. In addition to these, the graphics that characterize the pause menu have been created in which on a fence there are the buttons of the settings and in the background there is James standing with a pitchfork and a series of chickens, even these graphics have already been reported on ***page 33*** at ***fig. 5.1.18***.

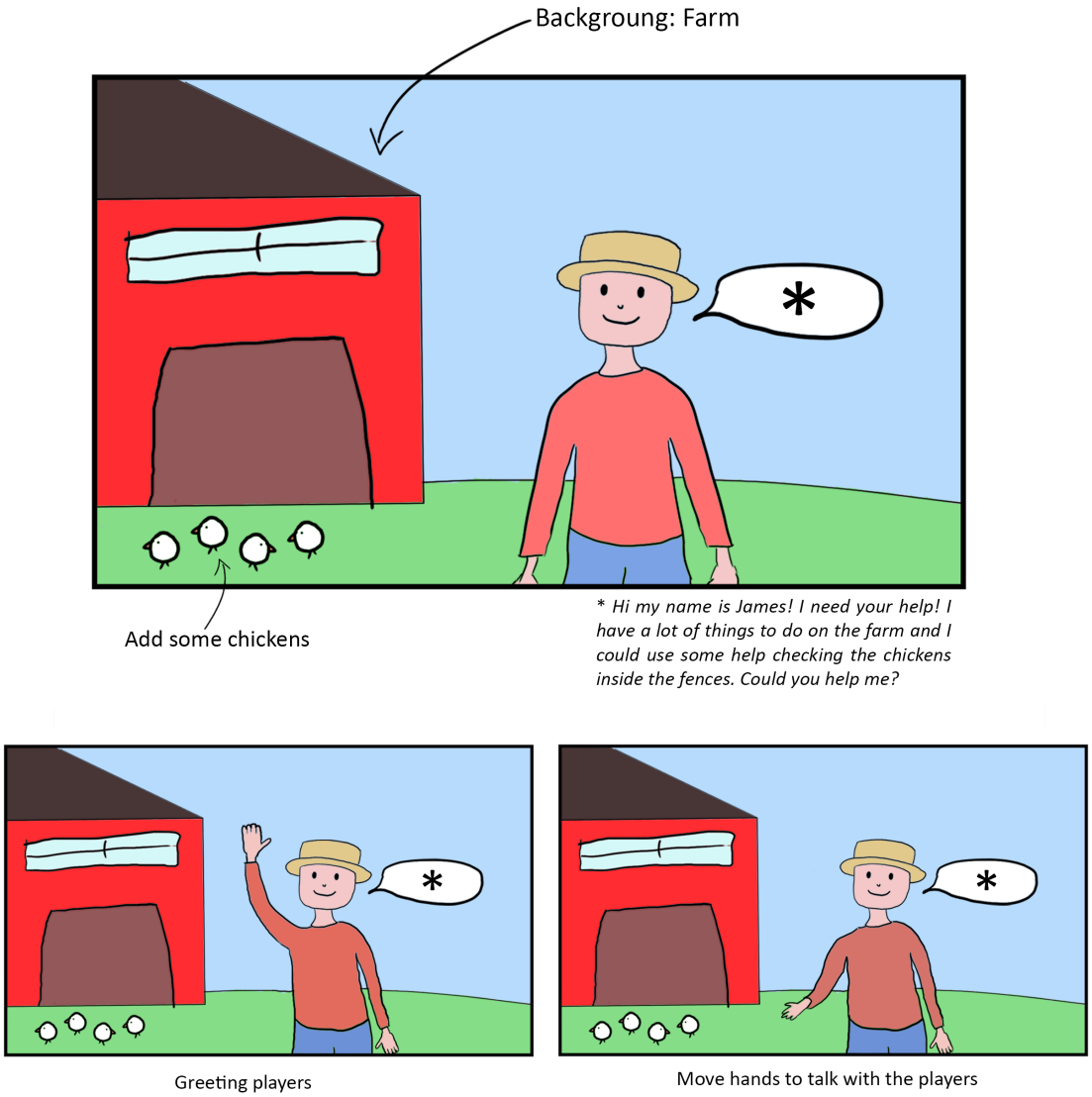
At a last moment a last UI was added to give feedback to the child and also to the parent. This UI occurs when there are connection problems with the main server and is always composed of James but also of a button that allows to retry the reconnection to the server.

5.3.28 Server connection error feedback

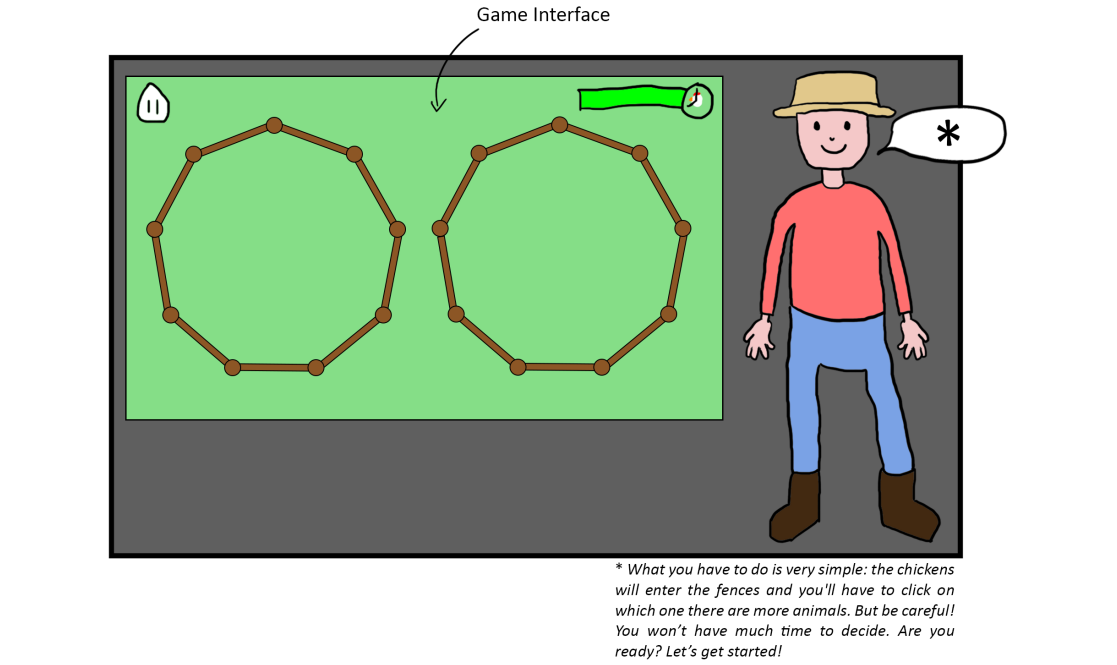
In the realization of these UI were used animations downloaded from Mixamo and specific frames were chosen that best characterized the feedback to be transmitted. In addition, expressions have been modeled to better characterize the character of James. To make

them, no specific shape keys have been added on Blender but the facial vertices have been modeled on the various models downloaded from Mixamo.

After completing the graphics it was time to design and animate the video tutorial to explain the game to the child. The video was conceived as a union of two parts: the first more introductory in which the child was taken into the world of James' farm and a second in which the farmer explains the rules of the game. The settings and shots are different in fact, the first part has a fixed shot with a medium plane on James with his farm in the background; in the second part there is a shot with James in full figure on the side that explains the game and indicates the application screen to the players.



5.3.29 Storyboard of the first part of the video



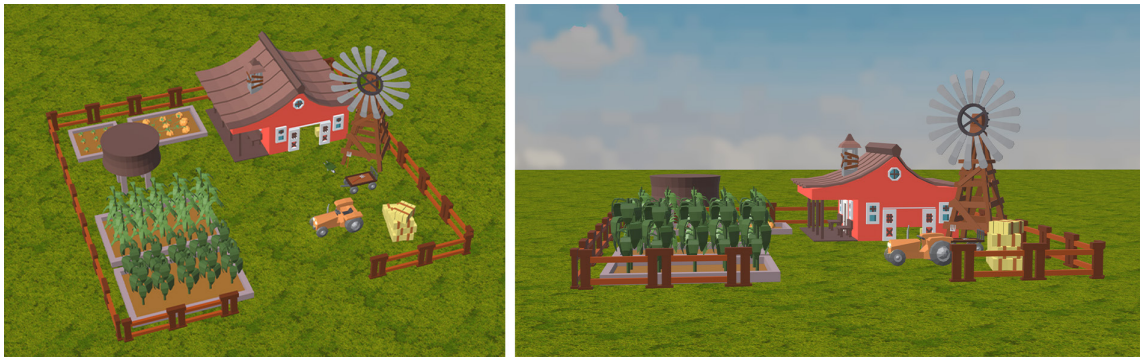
5.3.30 Storyboard of the first part of the video

After conceiving and realizing the storyboard of the idea there was the modeling and animation part of the video. The farm was downloaded from the Turbosquid site for free and was created by Mitylernal. [13]

5.3.31 3D model downloaded from Turbosquid

It was chosen precisely for the low poly cartoon style and was perfect for the initial setting of the video.

Starting from the already modeled farm, some elements have been modified, such as the sky and the ground, and some models of objects have been added or moved to make everything more inherent with the application and with the chosen framing. Both the ground and the sky have been downloaded for free and without copyright so that they can be included in the final rendering on Blender.

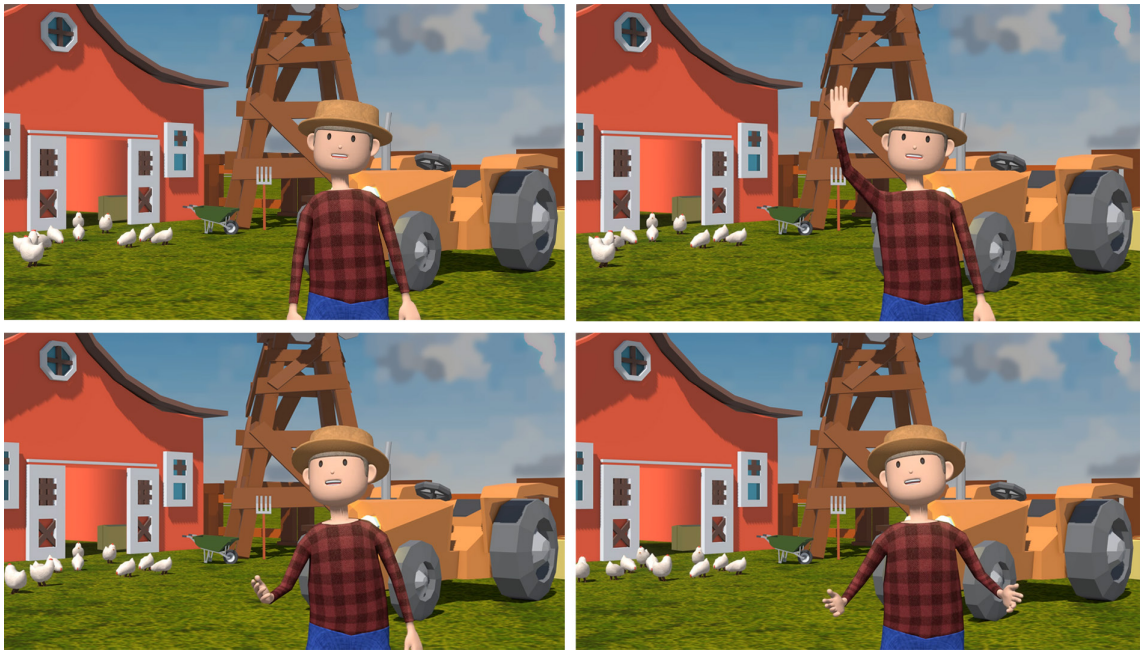


5.3.32 Farm pack

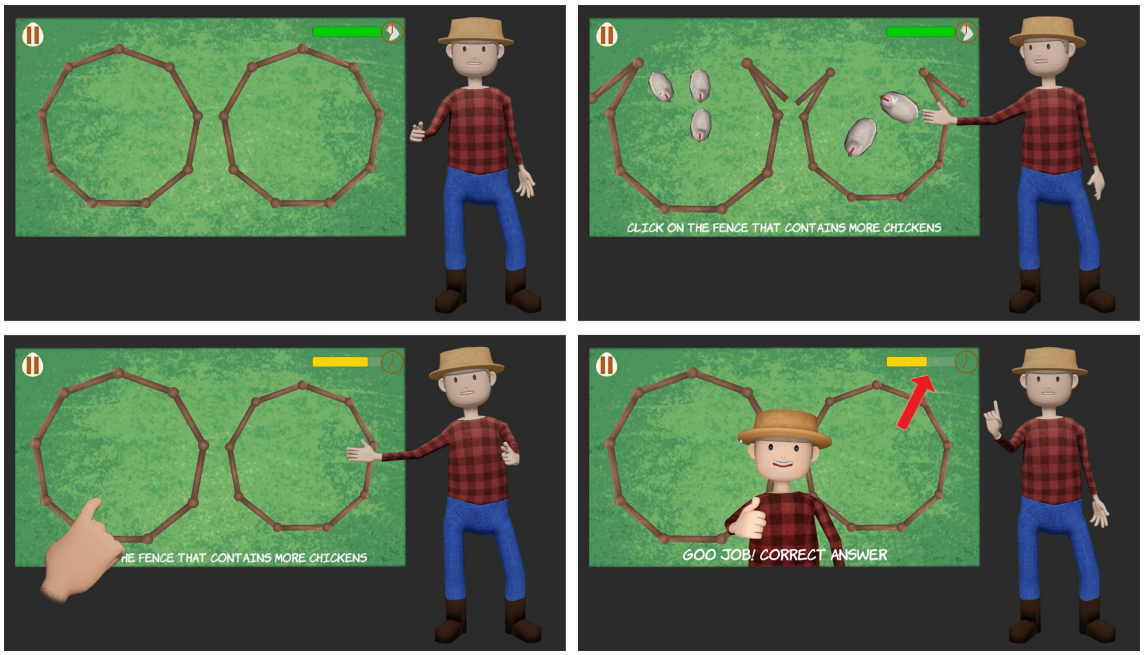
Shape keys have been added to James' model so as to automatically have a lip synchronization in a simple way with the reference audio via Blender. In addition to this and the movements while talking, some walking or eating chickens were added to the background of the shooting to characterize the farm even more.

In this first part of the video the farmer introduces himself to the player and asks him/her for help in counting the chickens in the fences given the many jobs that there are to do on the farm. The second part, on the other hand, concerns more specifically the explanation in a visual way, as well as auditory, of what must be done during the game. In fact, James will explain that the chickens will enter the enclosures, and the child will have to click on which enclosure the most animals are located. Also remember that the player will not have much time to respond because there will be a timer. In this part of the video James is full-length and indicates to his left where an example of a game trial is taking place.

This video is shown before the various levels of play so as to explain everything to the child. The second part of the video can then be reviewed by clicking on the tutorial item in the pause menu.



5.3.33 Screenshot of the first part of the video



5.3.34 Screenshot of the second part of the video

5.4 SOUND DESIGN

Given the age of the players it was decided to add a lot of sound feedback to allow the child to understand what is happening. This was intended to allow a simple and immediate use for the player. So sound gimmicks were found that characterize different feedback to be

given to the child. First of all it was decided to explain the purpose of the game and how to play through the voice of James in the initial video. The same voice has been used to accompany the UIs that indicate the wrong answer or the right one. In addition to James' voice, the sound of the ticking of the clock that starts as soon as the timer starts has been added. At first other more environmental sounds had been thought of but then this idea was discarded so as not to further distract the child and make him/her focus only on the task.

Since English was chosen as the main language of the application, a simple voice was needed without any particular accents or modulations. So it was chosen as the voice for the farmer James that of Professor Anthony Vickers, Emeritus Professor of the University of Essex in Colchester. Anthony's voice immediately seemed perfect precisely because it did not present strange modulations of timbre and dialectal or cultural influences. In addition to this, the words and phrases have been scanned perfectly and in a simple way for the children, still having a timbre modulation that reflects the expressions and visual feedback. In fact, it is not monotonous and devoid of emphasis and sound expression.
It is not excluded later to add other languages and therefore other male voices for James in the different languages of reference.

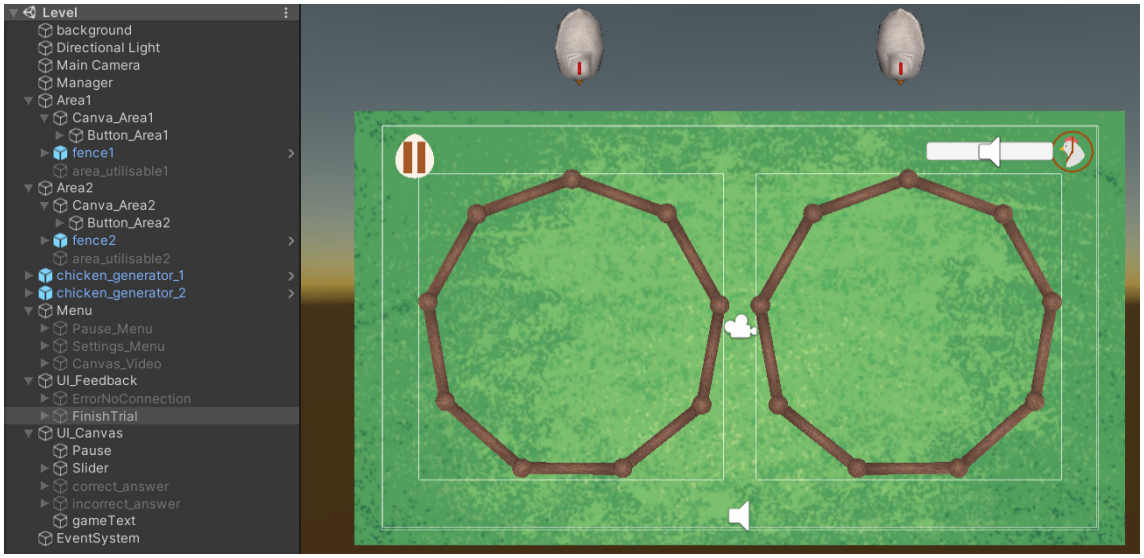
Instead as for the sound of the timer is searched and downloaded from the site Freesound. This site features a collaborative repository of Creative Commons licensed audio samples with over 500,000 sounds. Among all these, a simple sound was chosen that could be repeated in a loop for as long as the child has available to give an answer. The sound in the application starts at normal speed and once the chickens disappear and the color of the slider changes the sound feedback is speeded up to emphasize the little time missing.

In addition to the first part of the video, in addition to James' voice, an ambient audio was added in the background, always downloaded from the Freesound library. This was done to better characterize this part of the video and make it as natural and real as possible.

These sound and visual feedbacks have been added precisely to allow children who do not know how to read to be able to play independently and improve their sense of number through the continuous trials generated by the server.

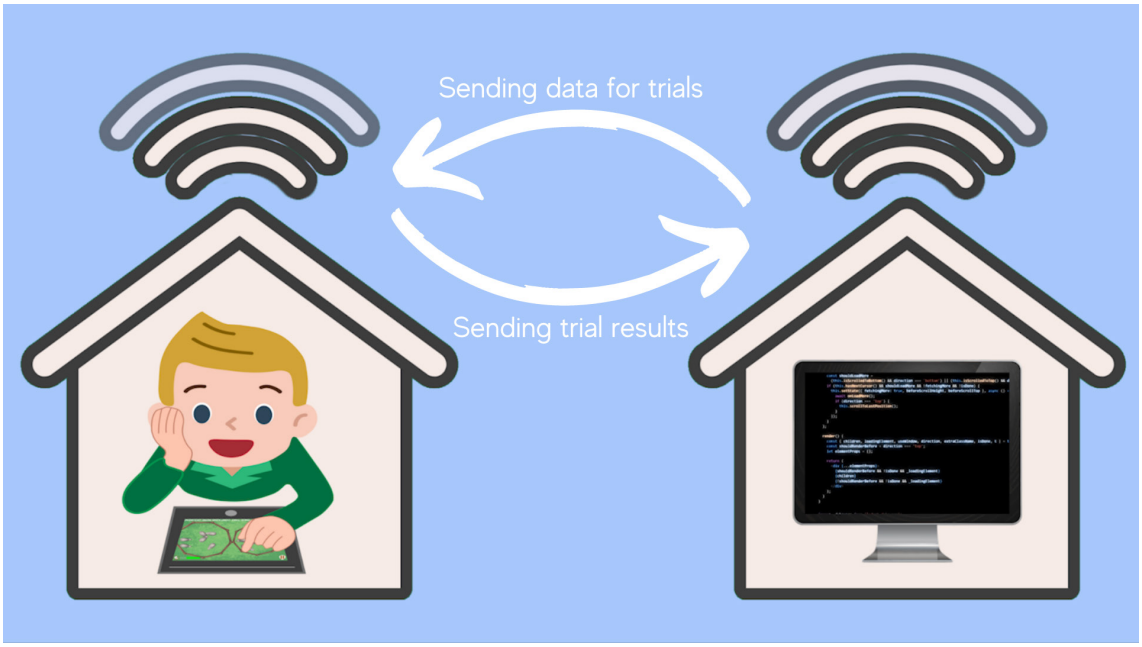
PROGRAMMING BEHIND THE GAME

After describing all the graphics and the conception of the application from the player's point of view, in this chapter will be described in depth how the game dynamics were designed. Starting with describing how the main level is composed on Unity. In this scene two areas have been placed: each one contains, for the respective fence, a canvas with inside the invisible button that the child will have to click to give the answer and a circumference that during the game will never be used but that has served to calculate the usable area in which the chickens can be arranged. Above these two areas were placed two generators consisting of two gameObjects of the chicken model that have the function of generating a defined number of chickens that will enter the corresponding area. These are the main objects of the layer, in addition to these the scene in Unity has canvases that make up the various UIs already described in the previous chapter and an empty object called Manager that manages almost all the scripts that make up the game.



6.1.35 How the scene is built on Unity

6.1 CLIENT-SERVER COMMUNICATION



6.1.36 Graphical representation of client-server communication

The game was designed to relate a server that generates trials, according to an algorithm that refers to the filtering hypothesis and the research at the base of the original experiment, and a client that takes this data and processes it creating a real game. In other words, the test data is generated in the server and it sends this information to the client which is the unity engine and after the player has completed the various trials, the client sends the results as a response to the server.

For this architecture, WebSockets was used to run it with the Python programming language using Spyder IDE (Integrated Development Environment) since it is the most suitable for data correlation, mathematical problems and graphs. Both the client and the server incorporate the socket library to create a connection, using the same port, and this library helps prevent users from being blocked when there are multiple users connected.

The communication takes place through a string in which a first part describes the message to be sent and then a JSON is placed side by side that represents the data to be sent. JSON format is used because it easily serializes objects into both programming languages; this also implies that classes are defined with identical variables on both the client and the server.

```python
import socket
from client_handler import ClientHandler
from dummy_client_handler import DummyClientHandler

class Create_Game:

    def __init__ (self, trials_matrix):
        self.trials_matrix = trials_matrix

    def run(self, simulation_on, nnd_selector, alpha, sigma, ServerSocket, host, port, DB, ThreadCount):

        # REAL GAME
        if simulation_on == 0:
            try:
                # bind() method assigns an IP address and a port number
                # to a socket instance. So, in a way, it binds the port number
                # with the IP address
                ServerSocket.bind((host, port))
            except socket.error as e:
                print(str(e))

            print('Waiting for a Connection..')
            # In order to accept a connection, the server must stay in a
            # 'listening' position
            ServerSocket.listen(5)

            while True:
                # When the client wants to establish the connection, the server
                # must accept it --> method .accept(), which returns:
                    # Client: is a new socket object usable to send and receive
                    # data on the connection
                    # Address: is the address bound to the socket on the other
                    # end of the connection.
                Client, address = ServerSocket.accept()
                print('Connected to: ' + address[0] + ':' + str(address[1]))
                player_id = DB.get_player(address[0])
                if player_id == - 1:
                    player_id = DB.add_player(address[0])
                # gameThread becomes an Object of class ClientHandler, which
                # is a thread, so to make it actually running, we must call the
                # start() method on it, which calls automatically the run() method
                gameThread = ClientHandler(Client, DB, player_id, self.trials_matrix)
                response_vector = gameThread.start()
                ThreadCount += 1
                print('Thread Number: ' + str(ThreadCount))
            # Once all is done, close the DB instance and the Socket one
            DB.close()
            ServerSocket.close()
```

6.1.37 Server-side class where the connection occurs

```python
def get_reply(self, data):
    print("2 GET REPLY")
    # Must terminate reply with \n
    if data.strip() == 'TRIAL':
        print("ENTRATA NEL POSTO RANDOM")
        trial = self.generate_random_trial()
        return json.dumps(trial.__dict__) + '\n'
    elif "TRIALS:" in data:
        return self.handle_trials_message(data.split(":"))
    elif "COMPLETE:" in data:
        print("Those are the complete trials received by the client")
        print(data)
        return self.handle_complete_message(data[9:].strip())
    elif "SETTINGS:" in data:
        return self.handle_settings_message(data[9:])
    return 'Server Says: ' + data
```

6.1.38 Function within the ClientHandler class of the server where the various algorithms are defined depending on the messages received

As for the design of the architecture from the server side was implemented by another student who focused on this part of the project. Below will be some code screens on the

server side related to client/server communication to have an overview generate various communication levels.

The next figure shows the class where the connection to the server is made. This is recalled by the script that checks all the correct functioning of the various trials in the game.

```
using System;
using System.IO;
using System.Net.Sockets;
using UnityEngine;
using System.Collections.Generic;

3 riferimenti
public class ClientToServer : MonoBehaviour
{
    const int port = 65432;
    private readonly StreamReader reader;
    private readonly StreamWriter writer;

    1 riferimento
    public ClientToServer(){
        TcpClient tcpClient = new TcpClient("localhost", port);
        NetworkStream stream = tcpClient.GetStream();
        reader = new StreamReader(stream);
        writer = new StreamWriter(stream);
        writer.AutoFlush = true;
    }

    1 riferimento
    public Stack<TrialData> GetTrials(){
        writer.WriteLine("TRIALS:5");
        string line = reader.ReadLine();
        Debug.Log("GET TRIALS: " + line);

        try
        {
            TrialData[] trials = JsonUtility.FromJson<TrialsArray>("{\"trials\":" + line + "}").trials;
            return new Stack<TrialData>(trials);
        }
        catch (Exception e)
        {
            Debug.Log(e);
        }
        return new Stack<TrialData>();
    }

    1 riferimento
    public void CompleteTrials(){
        TrialsResults results = new TrialsResults(TrialsManager.instance.completedTrialResults);
        String resultsJson = JsonUtility.ToJson(results);
        writer.WriteLine("COMPLETE:" + resultsJson);

        string line = reader.ReadLine();
        TrialsManager.instance.ClearResults();
        if (line != "SUCCESS") Debug.Log("Unable to complete trials on server");
    }

    0 riferimenti
    public void Dispose(){
        reader.Close();
        writer.Close();
    }
}
```

6.1.39 Client-side class where the connection occurs

In fact, in ***figure 6.1.39*** the ConnectWithClient() function is shown inside the script called TrialsManager that through a try and catch method that tries to establish the connection, if there should be an error sends an error message and activates the UI described in the previous chapter on ***page 40***. Through the "Try Again" button this function is invoked until the connection is established.

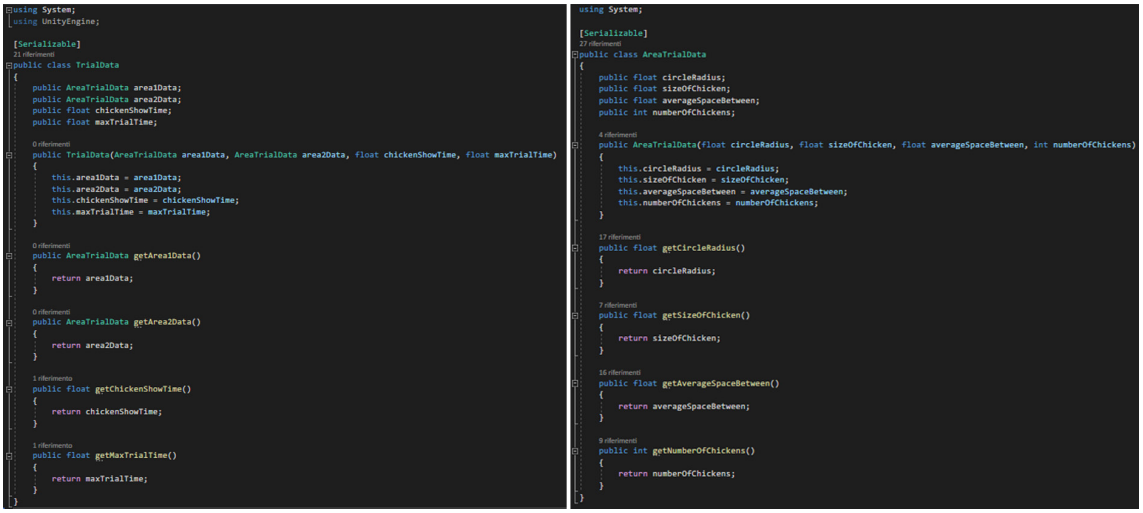
```
public void ConnectWithClient()
{
    try
    {
        client = new ClientToServer();
        Debug.Log("Logged in successfully.");
        connectionStarted = true;
        errorImage.SetActive(false);
        this.upcomingTrials = client.GetTrials();

    }
    catch (Exception e)
    {
        Debug.Log(e.Message);
        errorImage.SetActive(true);
        connectionStarted = false;

    }
}
```

6.1.40 TrialsManager() function that invokes the ClientToServer() connection function

So in the ClientToServer class there are various functions that are intended to send various messages to the server to have and give the data. In fact, to create the connection a new TcpClient is created that will connect with exactly the same port and the same hostname. Through a StreamReader and a StreamWriter the client can send messages to receive or send data, and this is precisely what happens in the GetTrials() functions, which receives the data array from the server, and CompleteTrials(), which sends an array of results to the server that will analyze them.
Both functions are invoked in the general TrialsManager class, the first immediately after establishing the connection to download the data and insert them into a special list that will be used in the game and the second after finishing the trials session so that the data can be send to the server that will analyze them.

The data saved on the client side is within a Stack collection, that refers to the serializable class TrialData, which saves the different data of the original array in variables that can be used in the game. In TrialData the first two variables refer to another serializable class

AreaTrialData, which allows to define every aspect of the relative enclosure, the third variable chickenShowTime indicates the time in which the chickens will be visible in their enclosures and the last maxTrialTime indicates the total time that the child has available to give an answer.



```
using System;
using UnityEngine;

[Serializable]
21 riferimenti
public class TrialData
{
    public AreaTrialData area1Data;
    public AreaTrialData area2Data;
    public float chickenShowTime;
    public float maxTrialTime;

    0 riferimenti
    public TrialData(AreaTrialData area1Data, AreaTrialData area2Data, float chickenShowTime, float maxTrialTime)
    {
        this.area1Data = area1Data;
        this.area2Data = area2Data;
        this.chickenShowTime = chickenShowTime;
        this.maxTrialTime = maxTrialTime;
    }

    0 riferimenti
    public AreaTrialData getArea1Data()
    {
        return area1Data;
    }

    0 riferimenti
    public AreaTrialData getArea2Data()
    {
        return area2Data;
    }

    1 riferimento
    public float getChickenShowTime()
    {
        return chickenShowTime;
    }

    1 riferimento
    public float getMaxTrialTime()
    {
        return maxTrialTime;
    }
}
```

```
using System;

[Serializable]
27 riferimenti
public class AreaTrialData
{
    public float circleRadius;
    public float sizeOfChicken;
    public float averageSpaceBetween;
    public int numberOfChickens;

    4 riferimenti
    public AreaTrialData(float circleRadius, float sizeOfChicken, float averageSpaceBetween, int numberOfChickens)
    {
        this.circleRadius = circleRadius;
        this.sizeOfChicken = sizeOfChicken;
        this.averageSpaceBetween = averageSpaceBetween;
        this.numberOfChickens = numberOfChickens;
    }

    17 riferimenti
    public float getCircleRadius()
    {
        return circleRadius;
    }

    7 riferimenti
    public float getSizeOfChicken()
    {
        return sizeOfChicken;
    }

    16 riferimenti
    public float getAverageSpaceBetween()
    {
        return averageSpaceBetween;
    }

    9 riferimenti
    public int getNumberOfChickens()
    {
        return numberOfChickens;
    }
}
```

6.1.41 TrialData and AreaTrialData serializable classes

6.2 GENERATION OF A NEW TRIAL IN UNITY

Each new level of the game is always set in another script that has the function of management called DataManager. This class implemented a SetNewTrialData public function with an object of the TrialData class. In this function the values related to the objects put into play are attributed, first of all two chicken generators are placed perfectly above the respective enclosures so as not to have problems in their entry.

Then the data is set to the respective areas and also scaled according to the size that the server wants to have. After that all the positions that can be within the playing area with a certain distance are calculated, this action was done since the game froze while looking for random points within the fence. While the animation that makes the fence models open is started, as many chickens are generated as there are two numberOfChickens variables sent by the server. The last lines of code of this function refer to an instance that refers to variables defined within the TrialsManager that are used to check later what the right answer will be and the boolean that indicates if the trial has started or the game is still paused.

```
public void SetNewTrialData(TrialData trialData)
{
    Random.InitState((int)System.DateTime.Now.Ticks);

    chickens_generators[0].transform.position = new Vector3(-4, 7, 0);
    chickens_generators[1].transform.position = new Vector3(4, 7, 0);

    data = trialData;

    areasData[0] = trialData.area1Data;
    areasData[1] = trialData.area2Data;

    areas[0].transform.localScale = new Vector3(areasData[0].getCircleRadius(), areasData[0].getCircleRadius(), areasData[0].getCircleRadius());
    areas[1].transform.localScale = new Vector3(areasData[1].getCircleRadius(), areasData[1].getCircleRadius(), areasData[1].getCircleRadius());

    //Find Points Inside Areas
     for (int i=0; i< areas.Length; i++)
     {
         this.CreateGrid(areas[i], areasData[i]);
     }

    //Start Animation to open the fences
    this.OpenFence(true);

    //Initialize Chickens
    for (int i = 0; i < areasData.Length; i++)
    {
        for (int j = 0; j < areasData[i].getNumberOfChickens(); j++)
        {
            GameObject newChicken = Instantiate(chickens_generators[i]);
            newChicken.GetComponent<Chickens>().SetChicken(areas[i], i + 1, areasData[i]);
            activeChickens.Add(newChicken);
        }
    }

    TrialsManager.instance.area1Value = areasData[0].getNumberOfChickens();
    TrialsManager.instance.area2Value = areasData[1].getNumberOfChickens();
    TrialsManager.instance.trialStarted = true; //Start Trial
}
```

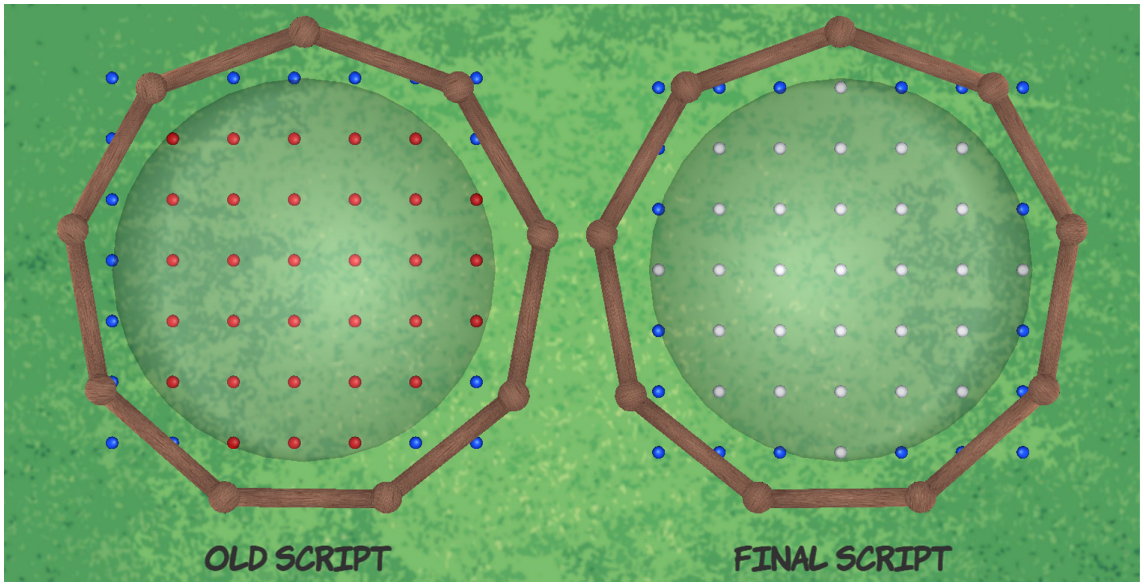
6.2.42 Function that sets data in the DataManager class

The chickens will move, once the trialStarted Boolean variable has become true, to their final position which is taken randomly from the list created in the CreateGrid method (GameObject area, AreaTrialData areaData) always present in the DataManager class. This method was created because during the design of the game there was some problems to find the positions randomly but with a relative distance, given by the variable averageSpaceBetween invites from the server. Since the number of chickens was quite high Unity had problems looking for and processing all possible positions and calculating their distances, a faster solution was found both on the computational side and on the temporal side.

Designing a grid in which there were all possible positions spaced exactly one apart from each other seemed like the easiest decision. A grid was first designed that started from the top right vertex for each reference area and once all the points were had, it was checked that they were part of the usable circumference. From a visual point of view this algorithm was not performing when it came to a number of chickens equivalent to the maximum possible positions in the enclosure, this is because the calculation of the positions was not perfectly centered with the walkable area. Then an algorithm was designed, in the figure, which starting from the center considers all the positions at a given distance in a radial way.

```
private void CreateGrid(GameObject area, AreaTrialData areaData)
{
    float radius_area = areaData.getCircleRadius() * 3f; //3f because the radius of area usable is (you can see it also in unity) 0.5*6=3
    int div = (int)(radius_area / areaData.getAverageSpaceBetween()); //how much you can divide the radius in same space with same distance between (=getAverageSpaceBetween)
    Vector3 centre_area = area.transform.position;
    List<Vector3> vectors= new List<Vector3>();

    if (vectors.Count == 0)
    {
        vectors.Add(new Vector3(centre_area.x, centre_area.y, 0));
    }

    //calculation all the point usable in the grid
    for (int i = 0; i < div+1; i++)
    {
        //calculation of points around the center
        for (int j = 1; j <= div; j++)
        {
            float x_plus = vectors[0].x + (j * areaData.getAverageSpaceBetween());
            float x_minus = vectors[0].x - (j * areaData.getAverageSpaceBetween());
            float y_plus = vectors[0].y + (i * areaData.getAverageSpaceBetween());
            float y_minus = vectors[0].y - (i * areaData.getAverageSpaceBetween());

            vectors.Add(new Vector3(x_plus, y_plus, 0));
            vectors.Add(new Vector3(x_minus, y_plus, 0));
            vectors.Add(new Vector3(x_plus, y_minus, 0));
            vectors.Add(new Vector3(x_minus, y_minus, 0));
        }

        if (i != 0)
        {
            Vector3 v_i_plus = new Vector3(vectors[0].x, vectors[0].y + (i * areaData.getAverageSpaceBetween()), 0);
            vectors.Add(v_i_plus);

            Vector3 v_i_minus = new Vector3(vectors[0].x, vectors[0].y - (i * areaData.getAverageSpaceBetween()), 0);
            vectors.Add(v_i_minus);
        }
    }

    //Take only the points that are inside the circle
    foreach (Vector3 v in vectors)
    {
        float d_x = (v.x - centre_area.x) * (v.x - centre_area.x);
        float d_y = (v.y - centre_area.y) * (v.y - centre_area.y);

        if ((Math.Sqrt(d_x + d_y) <= radius_area))
        {
            if (area == areas[0])
            {
                if (!createdPositionsArea1.Contains(v)) createdPositionsArea1.Add(v);
            }
            else
            {
                if (!createdPositionsArea2.Contains(v)) createdPositionsArea2.Add(v);
            }
        }
    }
}
```

6.2.43 Function that creates the final grid of all possible positions



6.2.44 Visual differences between the two algorithms that calculated the grids of the final positions

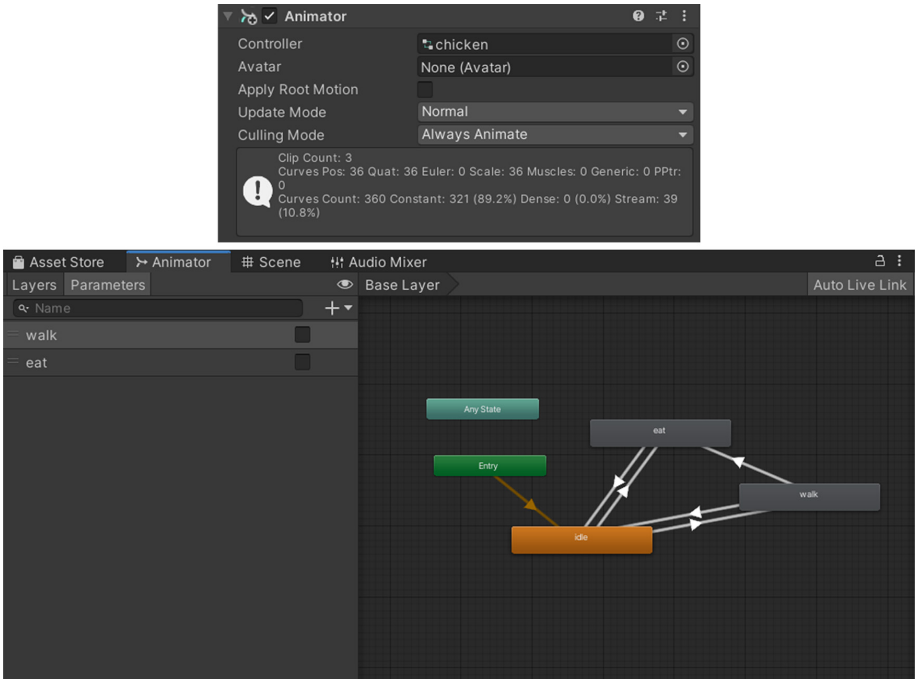
In *fig.6.2.44* it is visually shown how both algorithms that calculated the grid worked, the spheres were placed on all the calculated positions, those colored blue in both areas

are the discarded positions because not within the usable area, shown in the figure with a circle in transparency. The red spheres indicate the positions chosen with the old algorithm and instead the white ones are the positions calculated in the final game and the consistent difference between the definitive positions can be noticed.

Once the chickens reach their final position they will arrange themselves at that point with a random rotation and will switch from the animation of the walk to the one in which they eat. From that position they will never move and will remain to eat in loop all the time chickenShowTime. The chicken has an animator component that controls the various animations and the booleans that set which animation should be shown.

```
if (!arrived)
{
    if (startWalk)
    {
        animator.SetBool("walk", true);
        this.transform.position = Vector2.MoveTowards(this.transform.position, this.positionFinal, 5f * Time.deltaTime); //move chicken to Final Position
    }

    this.NoCollisionInsideCircle();

    if (Vector3.Distance(this.transform.position, this.positionFinal) < 0.01f)
    {
        arrived = true;

        //Set Random Rotation
        GetComponent<Rigidbody>().rotation = Quaternion.Euler(new Vector3(Random.Range(0f, 360f), 270f, 90f) * 1);

        animator.SetBool("eat", true);
    }
}
```

6.2.45 Script in the update of the Chickens class that schedules the arrival of chickens in the final positions



6.2.46 Chicken animator in Unity

6.3 USER-GAME INTERACTIONS

Once the chickens have reached their final positions time will begin to flow. Another script controls the buttons that allow the player to respond to the various trials and receive their feedback. Once the timer is gone, the slider will begin to descend so as to visually show the child the time is passing.

```
void Update()
{
    if (TrialsManager.instance.connectionStarted)
    {
        if (!TrialsManager.instance.trialStarted)
        {
            if (!firstTrial)
            {
                firstTrial = true;
                this.FirstTrial();
            }
        }
        else if (TrialsManager.instance.chickensReady)
        {
            if (!buttonsEnabled)
            {
                this.Buttons(true);
                buttonsEnabled = true;

                //Start timer
                timer.maxValue = TrialsManager.instance.chickenShowTime * 1000;
                timer.value = TrialsManager.instance.chickenShowTime * 1000;
                stopwatch.Start();
                timer.gameObject.GetComponent<AudioSource>().Play();
                timer.gameObject.GetComponent<AudioSource>().pitch = 1f;

                gameText.text = "Click on the fence that contains more chickens";
                gameText.GetComponent<AudioSource>().Play();
            }

            //Decrement timer
            timer.value = timer.maxValue - stopwatch.ElapsedMilliseconds;

            //Controll if time is > of ChickenShowTime
            if (stopwatch.IsRunning && stopwatch.ElapsedMilliseconds > (TrialsManager.instance.chickenShowTime * 1000) && !elapsedChickenShowTime)
            {
                foreach (GameObject c in istance_DataManager.activeChickens)
                {
                    c.SetActive(false);
                }

                //set value and color of slider
                clockImage.sprite = clockSprite[0];
                timer.maxValue = TrialsManager.instance.maxTrialTime * 1000;
                timer.value = TrialsManager.instance.maxTrialTime * 1000;
                timer.gameObject.transform.Find("Fill Area").Find("Fill").GetComponent<Image>().color = Color.yellow;
                timer.gameObject.GetComponent<AudioSource>().pitch = 1.2f;
                elapsedChickenShowTime = true;
            }

            //Controll if time is > of maxTrialTime --> incorrect answer for trail
            else if (stopwatch.IsRunning && stopwatch.ElapsedMilliseconds >= (TrialsManager.instance.maxTrialTime * 1000))
            {
                stopwatch.Stop();
                timer.gameObject.GetComponent<AudioSource>().Stop();
                double elapsedTime = stopwatch.Elapsed.TotalMilliseconds;
                TrialsManager.instance.AddTrialResult(elapsedTime, false);
                gameText.text = "";
                this.HandleLoss();
                this.Buttons(false);

                StartCoroutine(NewTrial());
                isCoroutine = true;
            }
        }
    }
}
```

6.3.47 Update() of the ButtonManager class

All this happens in the Update() of the ButtonManager class. If the timer is the same as the chickenShowTime in the game all the animals in the enclosures will disappear at the same time and the color of the timer will change from green to yellow, there will

also be an increase in the speed of sound. If the child does not give the answer within the maximum time indicated with the maxTrialTime variable, the counter of the wrong answers will be increased. In fact, for each answer given, two counters will be increased and the results will be saved in a list that will then be sent to the server, in addition to this the child's response time will also be saved so that the artificial intelligence can analyze all the results.

```
private void ButtonSelected(int selectedArea, int unselectedArea)
{
    stopwatch.Stop();
    timer.gameObject.GetComponent<AudioSource>().Stop();

    Debug.Log("Area selected: "+ selectedArea + " Time elapsed: " + stopwatch.Elapsed);

    bool correct = selectedArea > unselectedArea;
    double elapsedTime = stopwatch.Elapsed.TotalMilliseconds;
    TrialsManager.instance.AddTrialResult(elapsedTime, correct);
    gameText.text = "";
    gameText.GetComponent<AudioSource>().Stop();

    if (correct)
    {
        HandleWin();
    }
    else
    {
        HandleLoss();
    }

    this.Buttons(false);

    StartCoroutine(NewTrial());
    isCoroutine = true;

}

1 riferimento
private void HandleWin()
{
    UIImage[0].SetActive(true);
    UIImage[0].GetComponent<AudioSource>().Play();
    TrialsManager.instance.correctCount += 1;
}

2 riferimenti
private void HandleLoss()
{
    UIImage[1].SetActive(true);
    UIImage[1].GetComponent<AudioSource>().Play();
    TrialsManager.instance.incorrectCount += 1;
}
```

6.3.48 Functions that check if the answer is right or wrong and the sending of the related feedback to the screen

If before the time expires the player clicks one of the two buttons that are present in the areas, then makes his/hers choice, the answer given will be checked and depending on the result either the positive or negative feedback will be activated. Formerly the answer has been given and saved the result in the right list of results will be started a coroutine that will open a new trial each time until the end of the list downloaded from the server. Once this last list is finished, the results will be sent to the server via a corresponding message during the communication between client and server. The latter will analyze the answers and send another series of data that however adapt better and better to the player so as to model the game on the individual and allow him/her to improve.

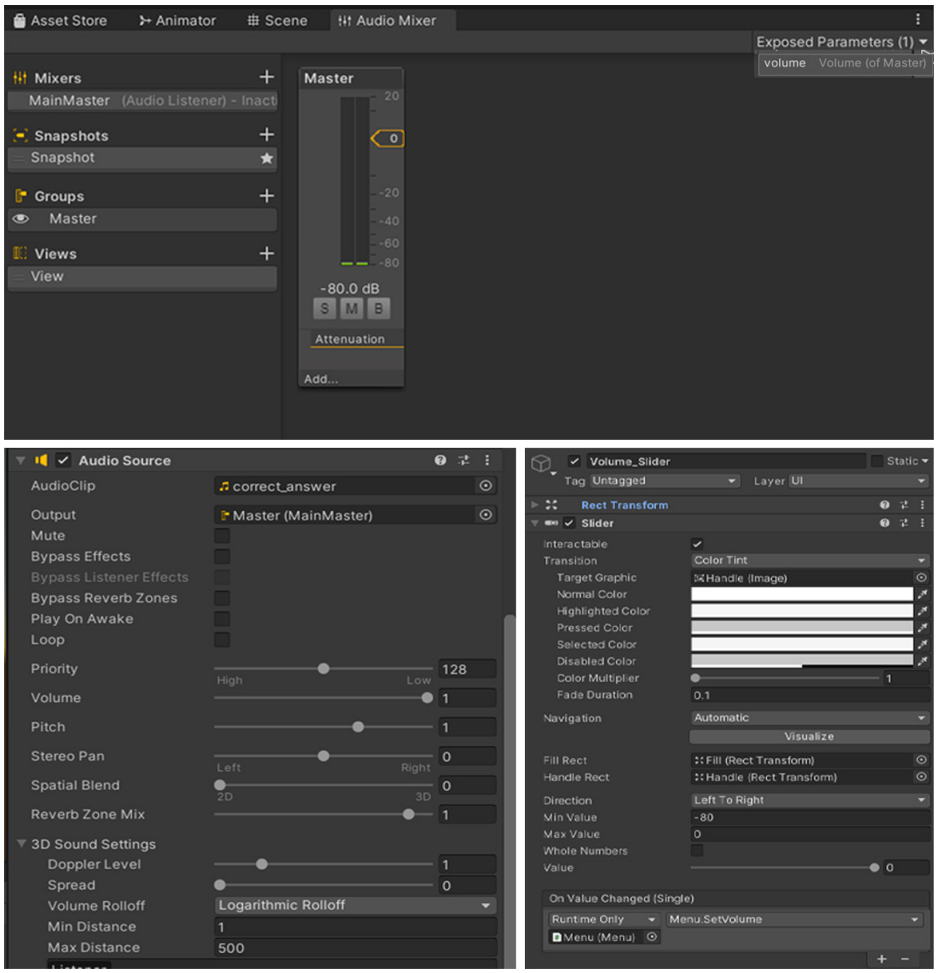
Timing is one of the most important factors in the game as choices must be made by players without allowing them to count chickens. That's why it's so important to send the server the amount of time the player also uses to select a response. It will be used later in AI assistance to know if the level was simple or difficult for the player and train the experience to allow to improve the ability of the child to focus on the sense of number.

The results are structured in a format of zeros and ones (zero indicates that the answer is wrong and one indicates the correct answer). Storing the submitted results is what led to the introduction of a related database, as the information as th player continues in his/hers gaming experience may become too large to be stored in a file, using a MySQL database seemed more effective.

As for, to conclude, instead the pause interfaces and the various menus have been designed so that once the pause menu is clicked, the game freezes and the child assisted by a parent can decide whether to enter the settings or exit the levels of the game. If the players decide to enter the settings they can access and see again the second part of the video, the one that has the purpose more of explaining the rules of the game, or they can change the general volume of the game. To change the overall volume of the application, a master mixer, connected with all the audios within the scene and also with the various videos, has been added to Unity.

By connecting the slider in the settings menu with the mixer depending on the value that the slider will have, the master will also adapt. In addition to these things, the player will

be able to access the main menu and close the game or return to the trial, which will start all over again, that is from the entry of the chickens into the enclosures and the timer will start from scratch.



6.3.49 Unity interfaces for setting the volume of sounds

PARAMETERS USED IN THE LITERATURE AND IN UNITY

From the examination of the document “Learning to focus on numbers” [6], in paragraph 2.2 there is an extensive description of how the space of stimuli is made. In particular, it is composed of one numeric variable and four non-numeric variables, which are:

- Number of Dots, which is the only numerical variable
- Item Surface Area, also known as ISA, non-numeric variable, defined as the area (in terms of number of pixels) occupied by a single dot
- Total Surface Area, also known as TSA, non-numeric variable, defined as the surface areas of the element multiplied by the number of elements.
- Field Area, also known as FA or “convex hull”, a non-numeric variable, which indicates the portion of space where the points actually fall
- Sparsity, also known as SPARS, non-numerical variable, which gives us information about how scattered the points are within their area, in other words it is the area of the field divided by the number of elements.

It is possible to condense the non-numerical variables all together into a single variable, called “Non-Numerical Dimension”, briefly NND, which is defined as the first main component (estimated by Principal Component Analysis) of the set of four previously defined non-numerical variables. During the original experiment the researchers found that the weights of non-numerical variables can be expressed through the following relationship:

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

It can be highlighted here that NND is, in this way, influenced only by non-numerical variables, but there is also a numerical variable to take into account. In fact, this previous

relationship can be modify by taking advantage of some other relationships between nonnumeric variables. During the experiment described in the original research, the following relationships were found that applied to each trial:

- TSA = ISA * Number-Of-Dots
- SPARS = FA / Number-Of-Dots

Thanks to these new relationships, it was possible to further reduce the space and represent the NND with only 3 variables, one of which is the numerical one while two of which are the non-numerical ones, which can be chosen. In this way, two new ways of representing the NND are obtained:

I. NND = (0.577 + 0.467*Number) * (FA/Number) * (0.487 + 0.473*Number) * ISA

II. NND = (0.577 + 0.467*Number) * SPARS + (0.487 + 0.473*Number) * (TSA/Number)

In the first expression basically the SPARS and TSA were removed, while in the second it is the opposite, that is, FA and ISA were removed. Of course, in both expressions there is something that was not present before, which is the numerical variable, here referred to simply as Number.

Starting from these variables, the client part uses other variables that are more suitable for the game in Unity for a matter of compatibility, since the units of measurement and the environment of the programs deviated:

- "number_of_chickens", the only numerical variable that is the number of chickens that are in each enclosure, has a 1:1 correspondence with the numerical variable Number-Of-Dots of the original research;
- "size_of_chicken", one of the non-numerical variables that indicates the size of each chicken in the hen house and conceptually, is equal to the ISA: ***size_of_chicken = ISA***;
- "circle_radius", another non-numerical variable that affects the size of the fence in the game and is directly proportional to the FA: ***circle_radius = [RADQ(FA / 3)] * π***

  FA represents the area where the points fall, while the circle_radius actually indicates the radius of the area where the chickens fall. This relationship is the result of the inverse calculation of the equation that calculates the area of a circumference A= π*(r^2) and, for a matter of compatibility with the Unity environment, it was necessary to add a multiplicative factor for the radius of 3.
- "average_space_between" indicates the average space between each chicken after it

has settled inside the hen house this variable is also connected with an initial variable, that is, to the SPARS, in a purely theoretical way;

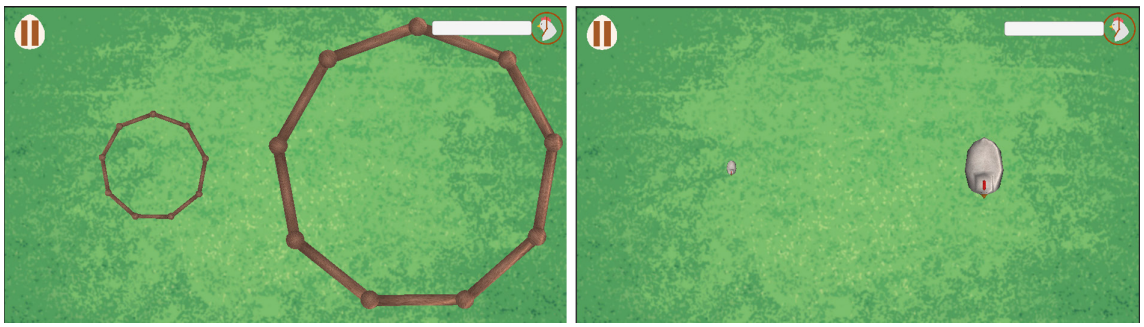
- "chicken_show_time", the first time variable that indicates the total time that chickens show on the screen before disappearing. Chickens disappear to prevent the player from simply counting the number of chickens.
- "max_trial_time", which instead is the maximum time that can last the test before being automatically considered as incorrect.

Starting from all these reflections, it was decided to use number, FA and ISA as the main variables to analyze the values of the relative variables studied and found in the literature. From the set of values found in the original experiment, the reference values were calculated, such as the maximum and minimum values for each parameter taken into consideration and the relationships between the left and right values, to begin the analysis and to understand how to represent everything on Unity since the reference units of measurement are different.

7.1 RELATE FA AND ISA WITH UNITY'S QUANTITIES

Visually checking the Unity application have assumed some mainly visual limits in the game on Unity, this is due to a mainly graphic and stylistic fact since either too large values or too small values are unusable on the game for the target audience. For this reason, analyzing the various values that non-numerical variables can have, these values were found:

- Circle_radius: 0.45 is minimum applicable and 1.2 is the maximum
- Size_of_chicken: 4 is the minimum and 10 the maximum applicable



7.1.50 The minimum and maximum values of circle_radius and size_of_chicken

As shown in the previous figure the minimum values are extremely small this is due to a limit for the values in literature. In fact, those values were analyzed and the ratios between the values for the right area and those for the left area were calculated, the same ratios

	Left Num	Right Num	ratio Num	Left FA	Right FA	ratio FA	ratio radius	Left ISA	Right ISA	ratio ISA
Maximum value ratio	32.00	20.00	1.6	31818	4892	6.5	2.55	246.46875	76.340909	3.23
Minimum ratio, value close to 1	16.00	17.00	0.94	35941	35949	1	1	106.28125	106.83333	0.99

Left RAD	Right RAD	1 < Ratio < 2.55
1.15	1.15	1
1.1	1.1	1
1.05	1.05	1
1	1	1
0.95	0.95	1
0.9	0.9	1
0.85	0.85	1
0.8	0.8	1
1.15	1.1	1.05
1.1	1.05	1.05
1.05	1	1.05
1	0.95	1.05
0.95	0.9	1.06
0.9	0.85	1.06
0.85	0.8	1.06
1.15	1.05	1.1
1.1	1	1.1
1.05	0.95	1.11
1	0.9	1.11
0.95	0.85	1.12
0.9	0.8	1.13
1.15	1	1.15
1.1	0.95	1.16
1.05	0.9	1.17
1	0.85	1.18
0.95	0.8	1.19
1.2	1	1.2
1.15	0.95	1.21
1.1	0.9	1.22
1.05	0.85	1.24
1	0.8	1.25
1.2	0.95	1.26
1.15	0.9	1.28
1.1	0.85	1.29
1.05	0.8	1.31
1.2	0.9	1.33
1.15	0.85	1.35
1.1	0.8	1.38
1.2	0.85	1.41
1.15	0.8	1.44
1.2	0.8	1.5
0.75	0.75	1
0.7	0.7	1
0.65	0.65	1
0.6	0.6	1
0.55	0.55	1
0.5	0.5	1
0.45	0.45	1
0.8	0.75	1.07
0.75	0.7	1.07
0.7	0.65	1.08
0.65	0.6	1.08
0.6	0.55	1.09
0.55	0.5	1.1
0.5	0.45	1.11
0.85	0.75	1.13
0.8	0.7	1.14
0.75	0.65	1.15
0.7	0.6	1.17
0.65	0.55	1.18
0.9	0.75	1.2
0.6	0.5	1.2
0.85	0.7	1.21
0.55	0.45	1.22
0.8	0.65	1.23
0.75	0.6	1.25
0.95	0.75	1.27
0.7	0.55	1.27
0.9	0.7	1.29
0.65	0.5	1.3
0.85	0.65	1.31
1	0.75	1.33
0.8	0.6	1.33
0.6	0.45	1.33
0.95	0.7	1.36
0.75	0.55	1.36
0.9	0.65	1.38
1.05	0.75	1.4
0.7	0.5	1.4
0.85	0.6	1.42
1	0.7	1.43
0.65	0.45	1.44
0.8	0.55	1.45
0.95	0.65	1.46
1.1	0.75	1.47
1.05	0.7	1.5
0.9	0.6	1.5
0.75	0.5	1.5
1.15	0.75	1.53
1	0.65	1.54
0.85	0.55	1.55
0.7	0.45	1.56
1.1	0.7	1.57
0.95	0.6	1.58
1.2	0.75	1.6
0.8	0.5	1.6
1.05	0.65	1.62
1.15	0.7	1.64
0.9	0.55	1.64
1	0.6	1.67
0.75	0.45	1.67
1.1	0.65	1.69
0.85	0.5	1.7
1.2	0.7	1.71
0.95	0.55	1.73
1.05	0.6	1.75
1.15	0.65	1.77
0.8	0.45	1.78
0.9	0.5	1.8
1	0.55	1.82
1.1	0.6	1.83
1.2	0.65	1.85
0.85	0.45	1.89
0.95	0.5	1.9
1.05	0.55	1.91
1.15	0.6	1.92
1.2	0.6	2
1.1	0.55	2
1	0.5	2
0.9	0.45	2
1.15	0.55	2.09
1.05	0.5	2.1
0.95	0.45	2.11
1.2	0.55	2.18
1.1	0.5	2.2
1	0.45	2.22
1.15	0.5	2.3
1.05	0.45	2.33
1.2	0.5	2.4
1.1	0.45	2.44
1.15	0.45	2.56

only adults

both adults and children

Left SIZE	Right SIZE	0.99 < Ratio < 3.23
4	4	1
4.5	4.5	1
5	5	1
5.5	5.5	1
6	6	1
6.5	6.5	1
7	7	1
7.5	7.5	1
8	8	1
8.5	8.5	1
9	9	1
9.5	9.5	1
10	10	1
10	9.5	1.05
8.5	8	1.06
9	8.5	1.06
9.5	9	1.06
7.5	7	1.07
8	7.5	1.07
6.5	6	1.08
7	6.5	1.08
6	5.5	1.09
5.5	5	1.1
5	4.5	1.11
10	9	1.11
9.5	8.5	1.12
4.5	4	1.13
8.5	7.5	1.13
9	8	1.13
8	7	1.14
7.5	6.5	1.15
7	6	1.17
6.5	5.5	1.18
10	8.5	1.18
9.5	8	1.19
6	5	1.2
9	7.5	1.2
8.5	7	1.21
5.5	4.5	1.22
8	6.5	1.23
5	4	1.25
7.5	6	1.25
10	8	1.25
7	5.5	1.27
9.5	7.5	1.27
9	7	1.29
6.5	5	1.3
8.5	6.5	1.31
6	4.5	1.33
8	6	1.33
10	7.5	1.33
7.5	5.5	1.36
9.5	7	1.36
5.5	4	1.38
9	6.5	1.38
7	5	1.4
8.5	6	1.42
10	7	1.43
6.5	4.5	1.44
8	5.5	1.45
9.5	6.5	1.46
6	4	1.5
7.5	5	1.5
9	6	1.5
10	6.5	1.54
8.5	5.5	1.55
7	4.5	1.56
9.5	6	1.58
8	5	1.6
6.5	4	1.63
9	5.5	1.64
7.5	4.5	1.67
10	6	1.67
8.5	5	1.7
9.5	5.5	1.73
7	4	1.75
8	4.5	1.78
9	5	1.8
10	5.5	1.82
7.5	4	1.88
8.5	4.5	1.89
9.5	5	1.9
8	4	2
9	4.5	2
10	5	2
9.5	4.5	2.11
8.5	4	2.13
10	4.5	2.22
9	4	2.25
9.5	4	2.38
10	4	2.5
2.5	2.5	1
3	3	1
3.5	3.5	1
4	3.5	1.14
3.5	3	1.17
3	2.5	1.2
4.5	3.5	1.29
4	3	1.33
3.5	2.5	1.4
5	3.5	1.43
4.5	3	1.5
5.5	3.5	1.57
4	2.5	1.6
5	3	1.67
6	3.5	1.71
4.5	2.5	1.8
5.5	3	1.83
6.5	3.5	1.86
5	2.5	2
6	3	2
7	3.5	2
7.5	3.5	2.14
6.5	3	2.17
5.5	2.5	2.2
8	3.5	2.29
7	3	2.33
6	2.5	2.4
8.5	3.5	2.43
7.5	3	2.5
9	3.5	2.57
6.5	2.5	2.6
8	3	2.67
9.5	3.5	2.71
7	2.5	2.8
8.5	3	2.83
10	3.5	2.86
7.5	2.5	3
9	3	3
9.5	3	3.17
8	2.5	3.2

only adults

both adults and children

7.1.51 All combinations of circle_radius and size_of_chicken divided by adult-only combinations and children's and adult combinations

were reached with these limit values.

However, it has been assumed that for the slightly smaller target the minimum values can be raised to 0.8 for the circle_radius and 4 for size_of_ chicken so as to be more visible and recognizable for younger players. In fact, starting from the values in literature by calculating the minimum values, that is, those closest to 1 because this implies maximum confusion for the child who is found to have similar data in both areas, and the maximum values have been found the respective ratios that can be obtained on Unity. In the ***figure 7.1.51*** are shown precisely these relationships and which can be applied to children and which to older people.

After finding all the limits of the two non-numerical variables it was time to find an equation that would allow to move from the domain of FA and ISA to the domain of circle_radius and size_of_chicken.

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

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

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

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

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

7.1.52 Process of formulas and calculations to find how to convert the values of one domain to those of the other

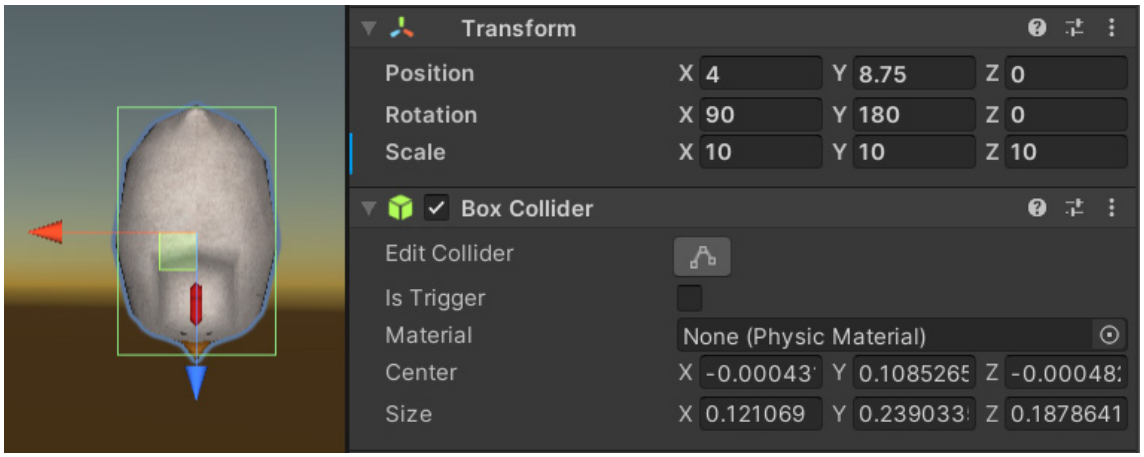
Using the formulas just show and through the maximum values in literature and those found in Unity the following conversion reports are found:

$$CR = 6 * 10^{-3}\sqrt{FA} \quad : \quad SC = 0.57\sqrt{ISA}$$

7.1.53 Conversion formulas: circle_radius to Field Area and size_of_chicken to Item Surface Area

7.2 THE CALCULATION OF THE AVERAGE SPACE BETWEEN

Having found these conversion ratios for variables that modify the area of fences and the size of chickens, a conversion ratio between average_space_between and sparsity was sought, at least initially, and the limits on Unity, which indicate the distance between two chickens, were calculated, therefore. To find the minimum value, the box collider relative to the chicken on Unity was analyzed and since its collider is a parallelepiped, tedious to the cube, it was decided to take into account the minimum value inside it. As shown in the figure, it was chosen the value of x that the minimum among the three; it indicates the whole side of the parallelepiped so to find the distance between the elements this value was divided by 2 and then multiplied by the scale of the chicken, that is, 10.



7.2.54 Collider box applied to chicken model on Unity

Once the limit values of the distance between one chicken and the other were set, the game was tested to check that all the limits worked. The idea was to create a list of possible solutions applicable on Unity for any value of the three non-numeric variables described so far in relation to the number. This was done because on Unity to find the positions a grid is created considering the avergae_space_between as a variable that defines how many segments the radius of the usable area can be divided into; so a lot of attention has been paid to analyzing the trend of this variable.

Then all the combinations between circle_radius, size_of_chicken and average_space_between were tested and the maximum number of chickens present within the usable area was found for each triad. Found all the numbers, they were reported on an Excel file to have a visual match of all these compatibility tests done. In the following image, the results have been reported through four columns:

- the first relating to the values of the circle_radius which have been increased from the minimum limit of 0.05 to the maximum limit;
- the second column refers to the size_of_chicken whose values have been increased by 0.5 between the two limits of the range;
- the third and fourth columns refer to the range of average_space_between values in which the same maximum number of chickens can be had;
- the last column refers to the maximum number of positions found by the grid.

All Possible Combination For Only Adults				
		max	min	max
circle_radiu	size_of_chick	average_space_between		number_of_chic
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
0.5	2.5	0.67	0.6	21
0.5	2.5	0.75	0.68	13
0.5	2.5	1.06	0.76	9
0.5	2.5	1.5	1.07	5
0.5	2.5	3.5	1.51	1
0.55	2.5	0.73	0.6	21
0.55	2.5	0.82	0.74	13
0.55	2.5	1.16	0.83	9
0.55	2.5	1.65	1.17	5
0.55	2.5	3.5	1.66	1
0.6	2.5	0.6	0.6	27
0.6	2.5	0.63	0.61	25
0.6	2.5	0.8	0.64	21
0.6	2.5	0.89	0.81	13
0.6	2.5	0.9	0.9	12
0.6	2.5	1.27	0.91	9
0.6	2.5	1.79	1.28	5
0.6	2.5	1.8	1.8	4
0.6	2.5	3.5	1.81	1
0.65	2.5	0.61	0.6	37
0.65	2.5	0.64	0.62	29
0.65	2.5	0.65	0.65	28
0.65	2.5	0.68	0.66	25
0.65	2.5	0.87	0.69	21
0.65	2.5	0.97	0.88	13

All Possible Combination For Both Adults & Kids				
		max	min	max
circle_radiu	size_of_chick	average_space_betwee		number_of_chic
0.8	4	0.66	0.66	45
0.8	4	0.75	0.67	37
0.8	4	0.79	0.76	29
0.8	4	0.8	0.8	28
0.8	4	0.84	0.81	25
0.8	4	1.07	0.85	21
0.8	4	1.19	1.08	13
0.8	4	1.2	1.2	12
0.8	4	1.69	1.21	9
0.8	4	2.39	1.7	5
0.8	4	2.4	2.4	4
0.8	4	3.5	2.41	1
0.85	4	0.63	0.62	49
0.85	4	0.7	0.64	45
0.85	4	0.8	0.71	37
0.85	4	0.84	0.81	29
0.85	4	0.85	0.85	28
0.85	4	0.9	0.86	25
0.85	4	1.14	0.91	21
0.85	4	1.27	1.15	13
0.85	4	1.8	1.28	9
0.85	4	2.55	1.81	5
0.85	4	3.5	2.56	1
0.9	4	0.65	0.64	57
0.9	4	0.67	0.66	49
0.9	4	0.74	0.68	45
0.9	4	0.85	0.75	37
0.9	4	0.89	0.86	29
0.9	4	0.95	0.9	25
0.9	4	1.2	0.96	21

●●●

0.7	10	3.5	2.1	1
0.75	10	2.25	1.6	5
0.75	10	3.5	2.26	1
0.8	10	2.39	1.7	5
0.8	10	2.4	2.4	4
0.8	10	3.5	2.41	1
0.85	10	1.8	1.8	9
0.85	10	2.55	1.81	5
0.85	10	3.5	2.56	1
0.9	10	1.9	1.9	9
0.9	10	2.69	1.91	5
0.9	10	3.5	2.7	1
0.95	10	2.01	2.01	9
0.95	10	2.84	2.02	5
0.95	10	2.85	2.85	3
0.95	10	3.5	2.86	1
1	10	2.12	2.12	9
1	10	3	2.13	5
1	10	3.5	3.01	1
1.05	10	2.22	1.58	9
1.05	10	3.14	2.23	5
1.05	10	3.5	3.15	1
1.1	10	1.65	1.65	13
1.1	10	2.33	1.66	9
1.1	10	3.3	2.34	5
1.1	10	3.5	3.31	1
1.15	10	1.72	1.72	13
1.15	10	2.43	1.73	9
1.15	10	3.44	2.44	5
1.15	10	3.5	3.45	1
1.2	10	1.8	1.61	13
1.2	10	2.54	1.81	9
1.2	10	3.5	2.55	5

1.2	9.5	1.8	1.61	13
1.2	9.5	2.54	1.81	9
1.2	9.5	3.5	2.55	5
0.8	10	2.39	1.7	5
0.8	10	2.4	2.4	4
0.8	10	3.5	2.41	1
0.85	10	1.8	1.8	9
0.85	10	2.55	1.81	5
0.85	10	3.5	2.56	1
0.9	10	1.9	1.9	9
0.9	10	2.69	1.91	5
0.9	10	3.5	2.7	1
0.95	10	2.01	2.01	9
0.95	10	2.84	2.02	5
0.95	10	2.85	2.85	3
0.95	10	3.5	2.86	1
1	10	2.12	2.12	9
1	10	3	2.13	5
1	10	3.5	3.01	1
1.05	10	2.22	1.58	9
1.05	10	3.14	2.23	5
1.05	10	3.5	3.15	1
1.1	10	1.65	1.65	13
1.1	10	2.33	1.66	9
1.1	10	3.3	2.34	5
1.1	10	3.5	3.31	1
1.15	10	1.72	1.72	13
1.15	10	2.43	1.73	9
1.15	10	3.44	2.44	5
1.15	10	3.5	3.45	1
1.2	10	1.8	1.61	13
1.2	10	2.54	1.81	9
1.2	10	3.5	2.55	5

7.2.55 Some examples of how the triads of non-numerical variables were found and joined with the maximum number of chickens possible for each triad

After creating all this new set of possible solutions applicable to Unity it was tried to apply the same relationship that binds the Sparsity with the Field Area to the average_

space_between. After numerous calculations and tests by relating the area of the usable enclosure area, the number of chickens could not be traced back to a value of average_ space_between identical to that reported in the list of possible combinations working on Unity. So it has been defined that average_space_between and sparsity are only linked in a theoretical and not practical way.

This is not a completely wrong thing since for the calculation of the NND it was possible to decrease the degree of non-numerical variables from four to two non-numerical variables and the numerical variable that indicates the number of chickens. These two non-numeric variables that related give the NND formula can be just FA and ISA that have their reference on Unity through the conversion formulas with circle_radius and size_of_chicken.

$$NND = (0.577 + 0.467 * NUMBER) * \frac{CR^2}{NUMBER * (6 * 10^{-3})^2} + (0.487 + \ 0.473 * \text{NUMBER}) * \left(\frac{SC}{0.57}\right)^2$$

7.2.56 Equation to calculate the NND with the number_of_chickens, size_of_chicken and circle_radius

Then another method was sought to derive the average_space_between starting from one or both non-numerical variables. The minimum distance, which there may be between two chickens, has been related to the size of them; since the positions within the area, that could be used for each average_space_between, were always the same and did not take into consideration the size of the chicken. In fact, the right maximum positions for the different size_of_chicken have been decided graphically and have been determined which triads are acceptable and which are not. It was thus decided to calculate the average_space_between in relation to the various sizes.

For each size_of_chicken the absolute minimum average_space_between was found; once a first table was created with these values, the line passing through two points was calculated, taking as the first point the minimum, that is SC=2.5 and ASB = 0.6, and as the second the maximum value present in the table, that is SC=10 and ASB=1.61.

$$\frac{y - 0.6}{1.61 - 0.6} = \frac{x - 2.5}{10 - 2.5} \rightarrow y - 0.6 = 0.135(x - 2.5) \rightarrow y = 0.135x + (0.6 - 0.34) \rightarrow y = 0.135x + 0.26$$

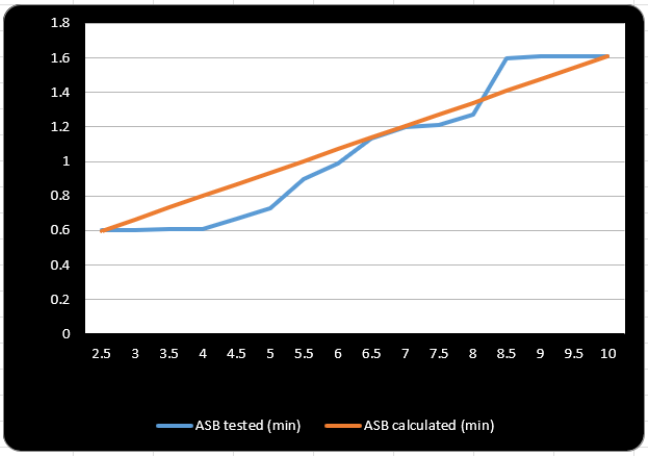
$$average\_space\_between = 0.135 * size\_of\_chicken + 0.26$$

7.2.57 Equation of the line through two points with minimum and maximum values of couples of average_space_between and size_of_chicken

Applied this formula to the couples of size_of_chicken and average_space_between and calculated a new value of average_space_between, so that it could be easily determined

on the server, it was noted that not all the new values of the line were above those calculated on Unity, shown in the next figure.

Size of Chickens	ASB tested (min)	ASB calculated (min)
2.5	0.6	0.5975
3	0.6	0.665
3.5	0.61	0.7325
4	0.61	0.8
4.5	0.67	0.8675
5	0.73	0.935
5.5	0.9	1.0025
6	0.99	1.07
6.5	1.13	1.1375
7	1.2	1.205
7.5	1.21	1.2725
8	1.27	1.34
8.5	1.6	1.4075
9	1.61	1.475
9.5	1.61	1.5425
10	1.61	1.61



7.2.58 Graph and table for calculating the average_space_between, using the values of size_of_chicken

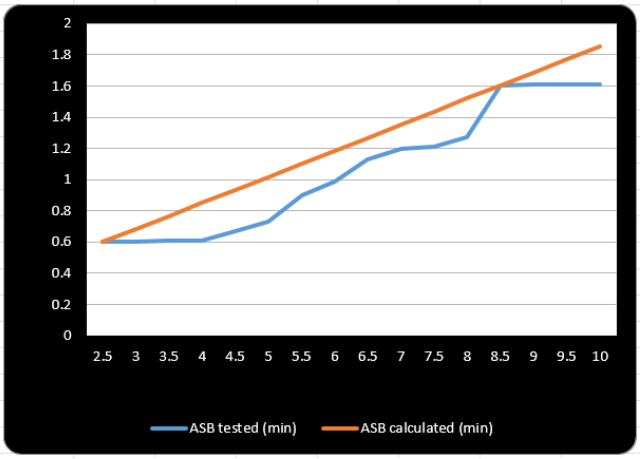
Thus another maximum point was chosen to calculate the relative line; this point is SC=8.5 and ASB= 1.6. With this point all values calculated by the two-point straight line method are above the relative points calculated on Unity. This calculation was used to find a mathematical relation to the space of combinations created on Unity so that the server can send in the simplest and correct way all the data to the client.

$$\frac{y-0.6}{1.6-0.6}=\frac{x-2.5}{8.5-2.5}\rightarrow y-0.6=0.167(x-2.5)\rightarrow y=0.167x+(0.6-0.4175)\rightarrow y=0.167x+0.183$$

$$average\_space\_between=0.167*size\_of\_chicken+0.183$$

7.2.59 New equation of the line through two points with minimum and maximum values of couples of average_space_between and size_of_chicken

Size of Chickens	ASB tested (min)	ASB calculated (min)
2.5	0.6	0.6005
3	0.6	0.684
3.5	0.61	0.7675
4	0.61	0.851
4.5	0.67	0.9345
5	0.73	1.018
5.5	0.9	1.1015
6	0.99	1.185
6.5	1.13	1.2685
7	1.2	1.352
7.5	1.21	1.4355
8	1.27	1.519
8.5	1.6	1.6025
9	1.61	1.686
9.5	1.61	1.7695
10	1.61	1.853



7.2.60 Graph and table for the right and definitive calculation of the average_space_between, using the values of size_of_chicken

7.3 CREATING THE SPACE OF POSSIBLE SOLUTIONS

After having found all the conversion formulas that allow the server to easily convert the data of the literature in order to create and send a series of trials that take inspiration and basis from the original experiment, it was time to create and realize the space of all possible solutions.

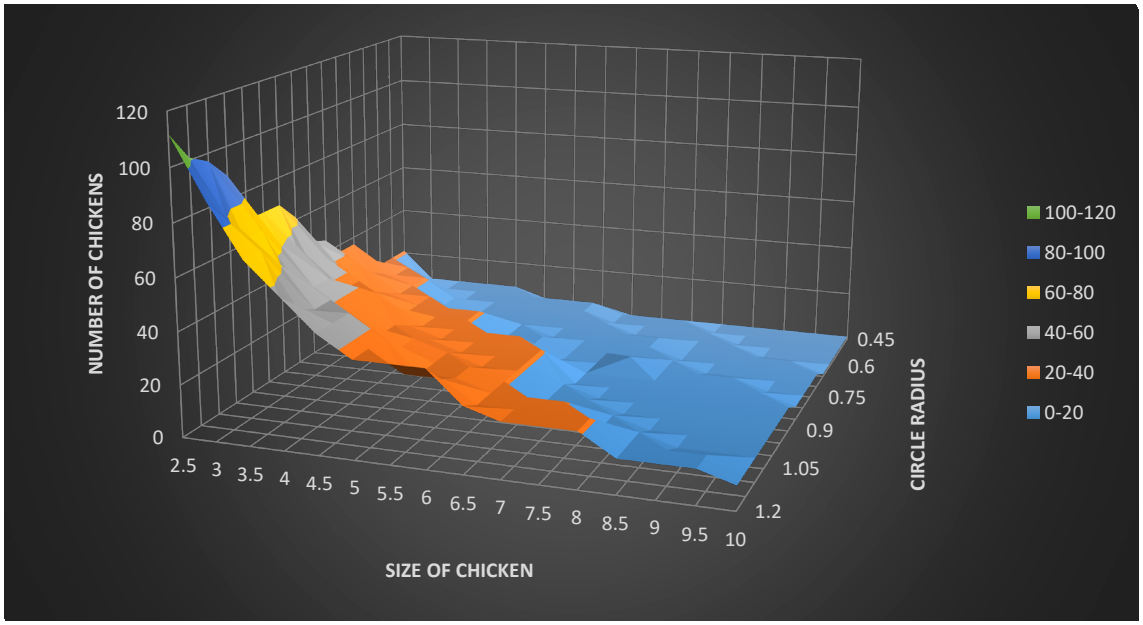
Analyzing the table, shown in part in ***figure 7.2.55***, the maximum possible number of chickens was taken for each pair of circle_radius and size_of_chicken. Taken these triads was created a table that would relate all the values obtained. All possible combinations, that can be made on Unity ,have been considered to create the table so:

- all values between 2.5 and 10 with regard to size_of_chickens;
- all numbers from 1 to 112 for the number_of_chickens;
- all values between 0.45 and 1.2 with regard to circle_radius.

	1.2	1.15	1.1	1.05	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45
2.5	112	101	97	89	69	69	69	61	48	45	37	37	27	21	21	21
3	89	81	69	69	61	57	45	45	37	37	29	25	21	21	13	9
3.5	69	69	61	49	45	45	37	37	29	25	21	21	21	13	9	9
4	57	49	45	45	37	37	37	28	21	21	21	13	13	9	9	9
4.5	45	45	45	37	29	29	25	21	21	21	13	9	9	9	9	9
5	37	37	37	29	29	25	21	21	21	21	13	9	9	9	9	5
5.5	37	29	25	25	25	21	21	21	13	13	13	9	9	9	9	5
6	37	29	25	25	21	21	21	21	13	13	9	9	9	9	9	1
6.5	25	21	21	21	21	21	21	13	9	9	9	9	9	5	5	1
7	21	21	21	13	13	13	9	9	9	9	9	9	5	5	5	1
7.5	21	21	21	13	13	9	9	9	13	9	9	9	5	5	5	1
8	21	21	13	13	9	9	9	9	9	9	5	5	5	5	1	1
8.5	13	13	13	9	9	9	9	9	9	5	5	5	5	1	1	1
9	13	13	9	9	9	9	9	9	9	5	5	5	1	1	1	1
9.5	13	13	9	9	9	9	9	9	5	5	5	5	1	1	1	1
10	9	9	9	9	9	9	9	5	5	5	5	1	1	1	1	1

7.3.61 Table of all possible solutions found on Unity

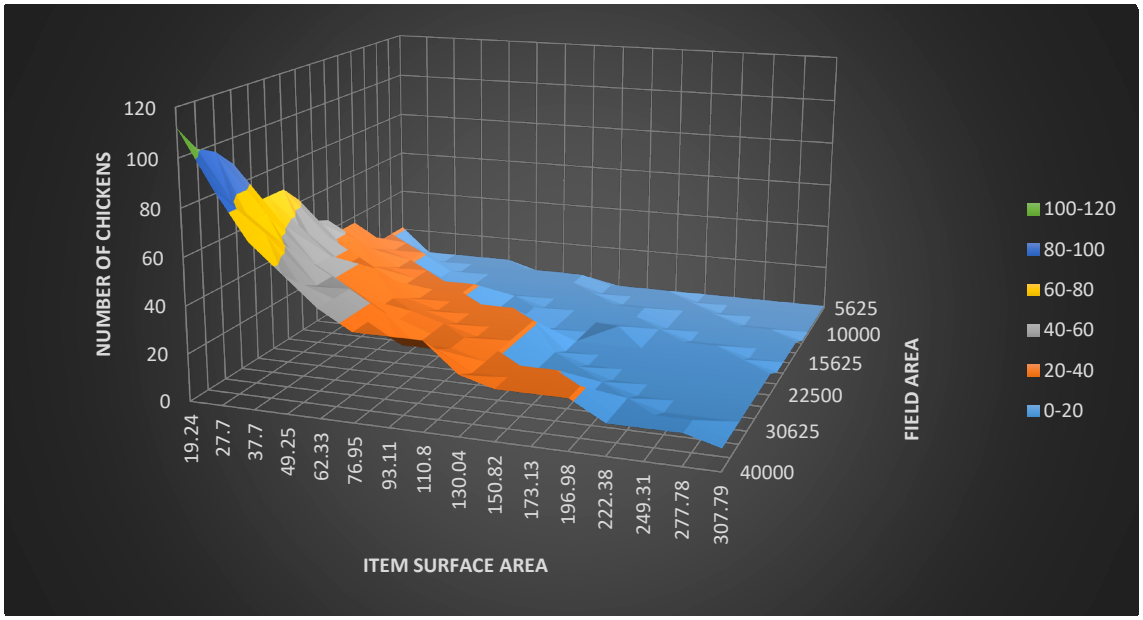
From this table, thanks to Excel, the space of possible solutions has been created three-dimensionally. The logic is that on the server side, the point moves below this graph depending on the child's answers by getting up as a result of right answers and then increasing the difficulty or lowering in the graph to have simpler trials if the child is struggling and responding incorrectly. After creating the relative graph, shown in ***figure 7.3.62***, it was checked that there was correspondence with all the various values of the triads with Field Area and Item Surface Area since the server will use these values and convert them only shortly before sending them to the client.



7.3.62 3D graph of all possible solutions found on Unity

	40000	36736.11	33611.11	30625	27777.78	25069.44	22500	20069.44	17777.78	15625	13611.11	11736.11	10000	8402.78	6944.44	5625
19.24	112	101	97	89	69	69	69	61	48	45	37	37	27	21	21	21
27.7	89	81	69	69	61	57	45	45	37	37	29	25	21	21	13	9
37.7	69	69	61	49	45	45	37	37	29	25	21	21	21	13	9	9
49.25	57	49	45	45	37	37	37	28	21	21	21	13	13	9	9	9
62.33	45	45	45	37	29	29	25	21	21	21	13	9	9	9	9	9
76.95	37	37	37	29	29	25	21	21	21	21	13	9	9	9	9	5
93.11	37	29	25	25	25	21	21	21	13	13	13	9	9	9	9	5
110.8	37	29	25	25	21	21	21	21	13	13	9	9	9	9	9	1
130.04	25	21	21	21	21	21	21	13	9	9	9	9	9	5	5	1
150.82	21	21	21	13	13	13	9	9	9	9	9	9	5	5	5	1
173.13	21	21	21	13	13	9	9	9	13	9	9	9	5	5	5	1
196.98	21	21	13	13	9	9	9	9	9	9	5	5	5	5	1	1
222.38	13	13	13	9	9	9	9	9	9	5	5	5	5	1	1	1
249.31	13	13	9	9	9	9	9	9	9	5	5	5	1	1	1	1
277.78	13	13	9	9	9	9	9	9	5	5	5	5	1	1	1	1
307.79	9	9	9	9	9	9	9	5	5	5	5	1	1	1	1	1

7.3.63 Table of all possible solutions that refer to the literature



7.3.64 3d graph of all possible solutions that refer to the literature

USABILITY TEST

After designing and creating the application in the last design phase, usability tests were carried out on the specific target. The System Usability Scale (SUS) is a simple ten-item scale that offers a global view of subjective assessments of usability. [14] Thanks to these questions it was possible to test the game from a design point of view to understand if the graphics, the feedback and all the elements that characterize the app are suitable for children, who will be the main end users of the game. This usability test, created by John Brooke, consists of ten sentences:

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system [14]

Each statement can have a value ranging from one, strongly disagreed, to five, strongly agree, and now it is calculated for each sentence a relative score that added to all those of the other sentences will give a percentage indicating the usability rate of the application.

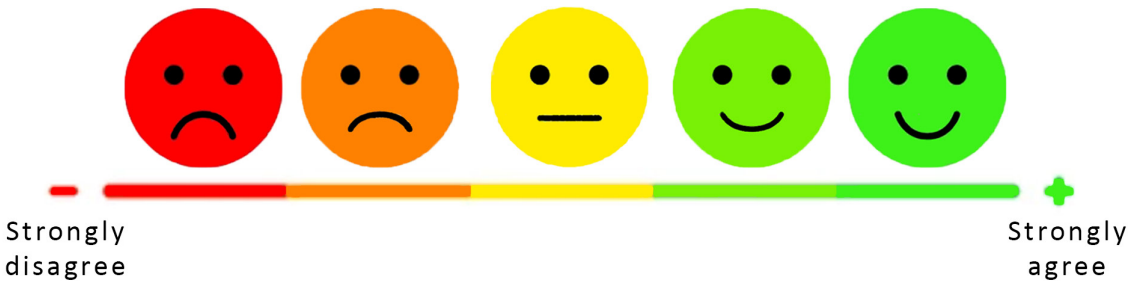
These tests were carried out during school hours at “Asilo Infantile Gaetano Zuccone”

in Quarona. Ten children between the four and six years played the game and then they answered first the ten statements of the test of Brooke and then to some questions that served to understand if they had problems during the game and also to have ideas for future graphic changes that mainly involve the setting of the game. Since kindergarten is bilingual, the children did not have much trouble understanding a game that is not in their native language, that is, Italian.

Since the target of the application is very small the questions have been revised to make them as understandable as possible for a child. Then they were rewritten and explained verbally also with the help of the teachers during the test session:

1. I think I'd like to use this game often
2. I found the game very difficult
3. I thought the system was easy to use
4. I needed some help to be able to use this game
5. I found the aspects of the game to feel good together
6. I thought this game didn't make too much sense
7. I think my friends and I are able to use the game fast
8. I found the game very complicated to use
9. I felt very confident using the system
10. I had to learn a lot of things to be able to play

The teacher also suggested given the age of the children to help them respond to quantify the relative value for each statement compared to that with numbers with five smileys that start from an angry to an extremely happy one.



8.65 Smileys used to allow children to express their answers

Thanks to this stratagem the children did not have much difficulty in understanding almost

all the questions and answering them as freely as possible. If children did not understand the sentence or did not know what to answer, an intermediate score of three for that statement was given for correctness. Being small another factor to take into account is that some children were a bit shy to answer questions asked by a person they do not know. Despite this, the game was liked and no major inaccuracies or elements of the application were found that were not very recognizable or difficult to understand.

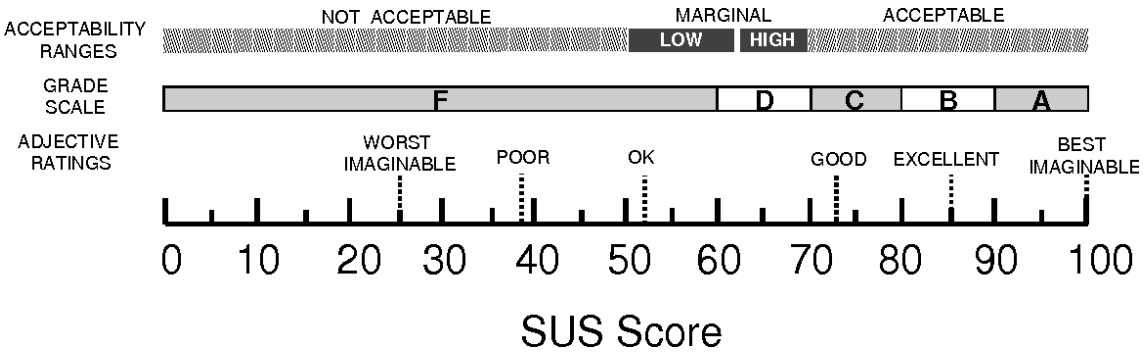
From the point of view of the architecture of the game some children had to listen more than once the initial video with the explanation of what they had to do this can have two meanings mainly:

1. not being their native language English perhaps they had not grasped some words or phrases said by James and therefore they had to listen to the video again to better understand the purpose of the game;
2. the part of the video that explain to the children what they have to do and all the elements of the game is a bit too fast and from a visual point of view the children do not understand perfectly what they have to do.

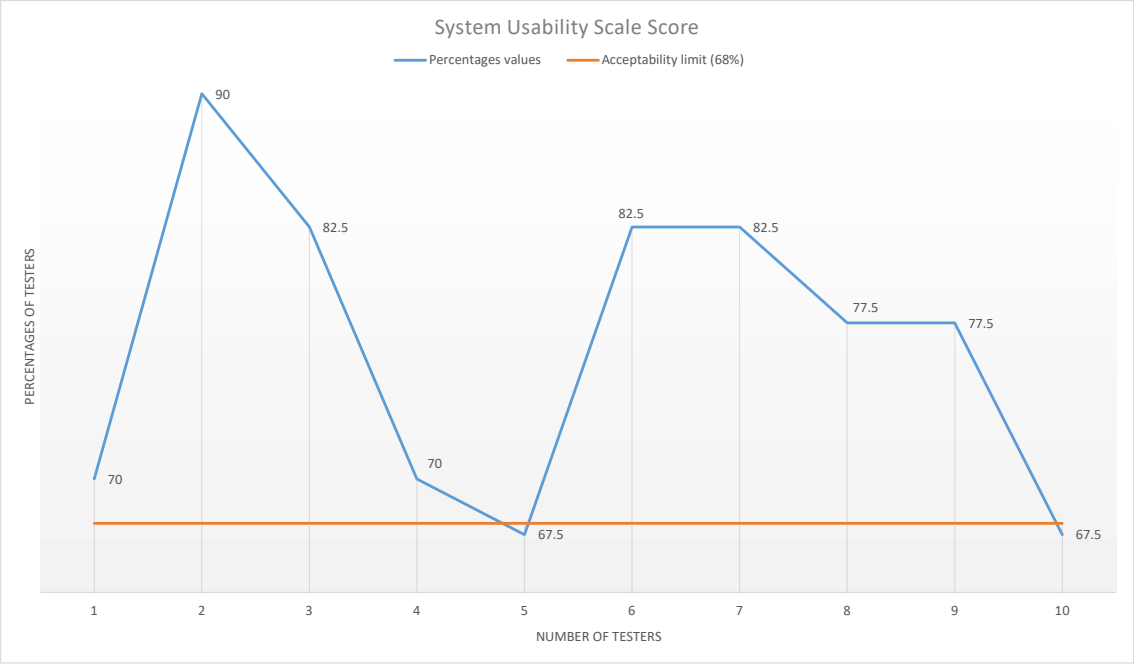
For both points there is a simple solution since the general idea is to create a game with multiple languages so as to be accessible to most children; for the second point the video can be easily change and make it slower and more visual for future players.

Another element present in the internal level of play that was not understood by the children was the button that pauses the game allows to reach the pause menu. This is because there is no explanation within the main video and this lack is due to the superficiality of those who created the video to think that it is a child knew the pause icon. Also for this aspect can be easily solve it by explaining this thing at an early stage within the video. In addition to this, an additional graphic could also be created that explains in detail each element of the game both visually and auditorially so as to allow children to understand all the elements that characterize the application so as not to ask for help from parents or teachers or those who are assisting them during the game.

Despite these problems, quite solvable in a short time, the game was liked and according to the SUS scale almost all the values are above the limit of acceptability, in fact the average System Usability Scale score is 68 [15], totaling an average percentage of 76.75.



8.66 Grade rankings of SUS scores from "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale" [15]

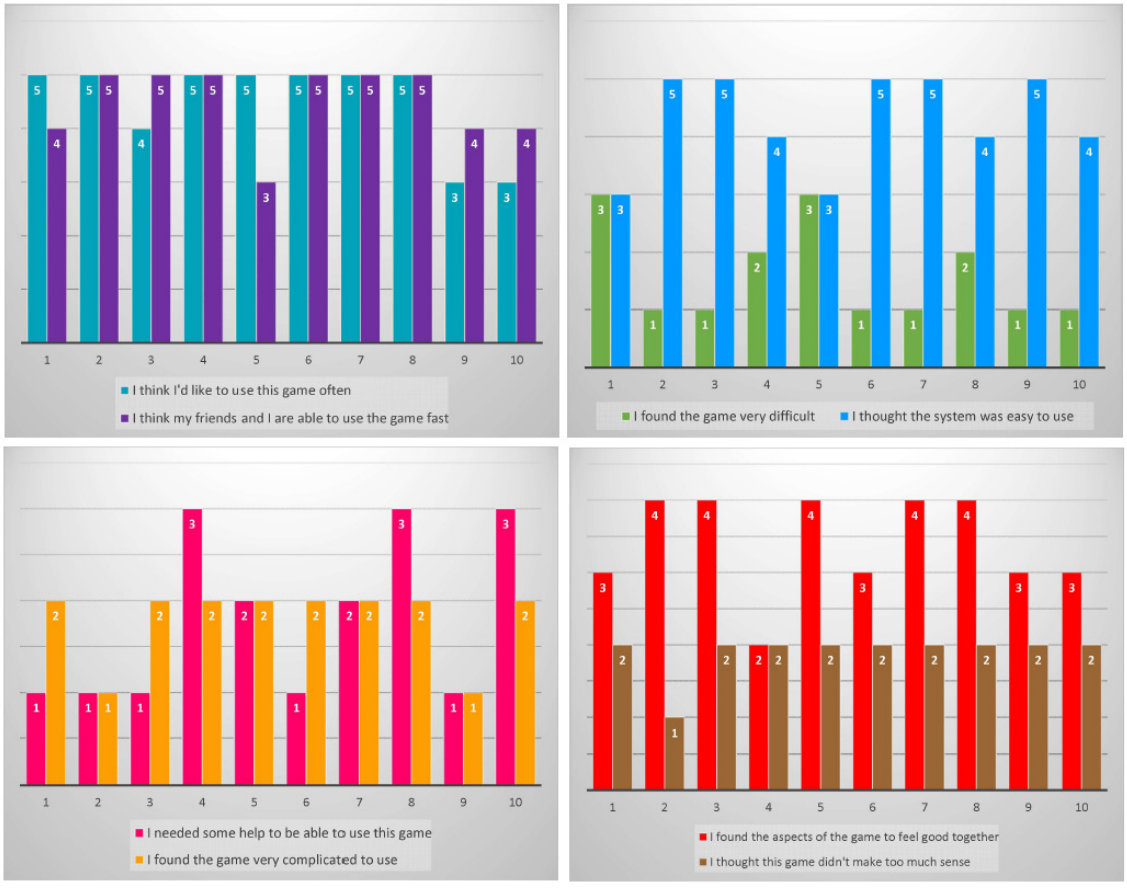


8.67 Graph of children's response rates

As can be seen from the figure, the results were very positive and the children did not have particular difficulties in using the game. The only substantial difficulties were not from the point of view of the difficulty in interacting with the game environment but more to understand its purpose. This is related to the problems of the initial video that have already been described but it can also be related to the fact that during the examination before starting the game the main purpose of the game was not said immediately but only after watching the video. This is so as not to influence the judgment and understanding of an element created specifically to give an explanation of the game. This may have led in some responses of the children in the test to give a higher score in considering the game difficult.

In fact, once they understood the game and what the child had to do, the answers were almost all correct and the children made few mistakes, some due to distraction and others, with a very low percentage, to the difficulty of the trial. Only two out of ten children had a result slightly lower than the usability threshold this is mainly due to the fact that they gave a lower than average score on the first statement. This can be conditioned to monotony during the game. This problem can be solved by varying, for example, environments and animals depending on the data of the trial in question.

In fact, children have no further difficulties during the test since each child was offered a series of trials of increasing difficulty and after finishing and answering all the questions and expressing their thoughts on the usability of the game to the question: “Do you want to try the game again?” 9 out of 10 children answered in the affirmative by trying to do more difficult trials or trying again to do one with the same level of difficulty. The answers were almost all correct and the children made few mistakes, some due to distraction and others, with a very low percentage, to the difficulty of the trials.



8.68 Scores given by children tested to Brooke’s claims

In the previous figure, the answers of all the children to the relative statements are shown, given thanks to the expedient of the smileys, the only statements not included in the image are the last two since in sentence 9 the children gave practically all the same answer and the last question was not understood by any child and talking with the teacher it turned out that the children practiced on mathematical games that introduced concepts of basic mathematics, during school hours, and therefore the agreed score was three.

The average, the variance and the standard deviation have been calculated to be able to better analyze the final result of the sus test:

- Average: 76.75
- Variance: 53.8125
- Standard deviation: 7.335700376

Although the average is higher than the usability limit, there is a fairly high and not too satisfactory variance. This is mainly due to the fact that the number of children was low to be considered a particularly significant sample and because there are some percentages that greatly lower the value of the overall average this is due to the difficulties and considerations had in the face of the children's responses. Despite some low scores the basic structure of the game holds up but must be better adapted to all children who could play it.

In addition to these statements, some questions were asked aimed at knowing how the children had found themselves within the videogame environment. In addition to the classic questions such as "Did you like the game?" or "Did you have fun?", questions to which the answer was positive, other things were asked to understand if feedback and the main graphics were within the reach of a child of 4/6 years. Once the game started, the children understood the visual and sound feedback, such as James' UIs with a given answer or the timer that indicated the time available to them. Having overcome shyness and initial distrust, they had fun and were able to finish and understand most, if not all, of the dynamics and architecture of the game.

The last question asked to the children was mainly designed for possible future development of the application. In fact, ideas for new levels were asked. Some children

remaining on the theme of the farm have begun to list and propose many other animals suggesting to change depending on the numerical size the type of animal that enters the given enclosure. In addition to this, some children have proposed different scenarios that can be combined with or replace the setting of the farm such as the sea or the garden with related animals attached. This question and the ideas received from the children will serve the future developers of the client part to expand and make the game less monotonous for future players.

CONCLUSIONS AND FUTURE WORK

This project was born with the aim of preventing a learning disorder still a little unknown and in some cases difficult to diagnose. With this application it is hoped first of all to train the child on the basis of the filtering hypothesis, then focus on a new way of treating the symptoms of dyscalculia and focus on the number in spite of the non-numerical variables that distract the child.

The project is still in development and in the subsequent design phases it is hoped to be able to continue with further tests to understand the effectiveness more mainly on the long duration on the child's mind.
This application has been designed to be available on all types of devices so that the child can learn and have fun even at home with the parents and not in specific places that could compromise and stress the child but allow him/her to use the game wherever he/she wants so as to be inclined and not obliged to play. That is why before continuing with further tests in the project there will have to be a refinement and the server-side programming and artificial intelligence will have to be completed.

Once the server part is completed by new thesis students who have recently joined the server-side development group, it is hoped to start with new testers so as to see the effectiveness of this application. For now, on the server side there is a lack of an optimized artificial intelligence that can analyze and implement new levels for the child.

This is a game designed to follow the child during his/hers developmental period by helping him/her and sharpening cognitive and mathematical skills. The long-term idea for the gaming app is to turn this already made part into a minigame. It will be implemented

as a smaller game within a larger game to be created chased.

Playing continuously in the way the game was created and structured can become monotonous and boring for the child; therefore, to make it more interesting it will have to be integrated into other types of minigames and levels. One strategy can be to include it at different intervals as a step to reach another level of the larger game, so as to have a dozen or twenty series of tests always modeled on the child's knowledge. IIn addition to this, more animals could be used in the game to avoid the same repetition of chickens depending on the difficulty and level of the child. Or always referring to the starting model, the animals could be replaced with plants, fruits or flowers to make the videogame process even wider and non-homogeneous. Another hypothesis to make the game more and more captivating and engaging is to change the game background; in fact, the peasant and rural theme that the farm has could allow an idea to expand and characterize the various game environments. Changing the background and fences, for example, with objects and situations typical of the peasant world could guarantee less monotony and make the game less boring and more engaging.

All these tactics serve to make the game more exciting for children since it is designed to be used for a long time. In addition to this, in the future it would be possible to create new scenarios and minigames that not only examine the sense of the number itself but also could expand mathematical knowledge such as numbers and their representation in figures or with the first calculation operations so as to make the game complete from many points of view. In order for this gaming application to reach a wide range of recipients it would be necessary to be able to change the language and therefore it would be necessary to create a wide range of languages so as to reach different parts of the world and even to help enlighten people about the presence of this disability. In addition, this game could be expanded later to suit both prevention and cure tactics. This means that instead of focusing only on the prevention of the diagnosis of dyscalculic, it could also go further to help people who already suffer from this disability and have passed the prevention phase so as to expand the target audience that can take advantage of the benefits that this game wants to have. This could be a strong point for this application in the future and be used by doctors and experts in the field.

REFERENCES

1. I. Berteletti, D. Lucangeli, M. Piazza, S. Dehaene, M. Zorzi (2010) “Numerical Estimation in Preschoolers“, Developmental Psychology, Vol. 46, No. 2, 545–551
2. Price, Gavin R. and Ansari, Daniel (2013) “Dyscalculia: Characteristics, Causes, and Treatments,” Numeracy: Vol. 6: Iss. 1, Article 2.
3. Dr. T. Nagavalli, Mrs. P. Fidelis Pauline Juliet, Sri Sarada “Technology for Dyscalculic Children” Sri Sarada College of Education, Salem-16.
4. M. Piazza, A. Facoetti, A. Trussardi, I. Berteletti, S. Conte, D. Lucangeli, S. Dehaene, M. Zorzi (2010) “Developmental trajectory of number acuity reveals a severe impairment in developmental dyscalculia” Cognition, Volume 116, Pages 33–41
5. L. Outhwaite, M. Faulder, A. Gulliford and N. Pitchford (2019) “Raising Early Achievement in Math With Interactive Apps:A Randomized Control Trial”, Vol. 111, No. 2, 284–298
6. M. Piazza, V. De Feo, S. Panzerid and S. Dehaene (2018) “Learning to focus on number” Cognition, Volume 181, 2018, Pages 35-45
7. Shannon, C. E. (1948) “A mathematical theory of communication”. Bell System Technical Journal, 27(3), 379–423
8. A.Wilson, S. Dehaene (2004) “The Number Race” INSERM-CEA Cognitive Neuroimaging Unit
9. D. Dotan, S. Dehaene, C. Huron and M. Piazza (2011) “The Number Catcher” INSERM-CEA Cognitive Neuroimaging Unit
10. K. Kucian, U. Grond, S. Rotzer, B. Henzi, C. Schönmann, F. Plangger, M. Gälli, E. Martin,

M. von Aster (2011) “Mental number line training in children with developmental dyscalculia”, NeuroImage 57, Pages 782–795

11. A. Wilson, S. Revkin, D. Cohen, L. Cohen and S. Dehaene (2006) “An open trial assessment of “The Number Race”, an adaptive computer game for remediation of dyscalculia”

12. D. Eddeland, “LPC style farm animals,” OpenGameArt, 6 August 2012.

13. Turbosquid: Farm Package 3D Model by Mitylernal (2019), Product code: 1469993

14. John Brooke (1995) “SUS: A quick and dirty usability scale”

15. A. Bangor, P.T. Kortum, and J.T. Miller (2009) “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale,” Journal of Usability Studies, 4(3), 114-123

ACKNOWLEDGEMENTS

First of all I would like to thank my coordinators, Professor Andrea Sanna and Professor Vito De Feo for having proposed this project to me and having followed me in these months. Then I would like to thank two professors from the University of Essex: Professor Katerina Bourazeri for the advice and feedback received during the design of the application and Professor Anthony Vickers for lending his voice to farmer James. I would also like to thank all the staff and all the children of “Asilo Zuccone” in Quarona for trying my project and helping me understand its usability. There are many people who have been close to me in these years that I would like to thank in reality. First of all my parents and my family. Thank you for always believing in me and supporting me. Without them I would never have come this far and I hope that whatever the future holds for me they will be as proud of me as I am of them. Then, not in order of importance, I would like to thank all my childhood friends who have been close to me despite the kilometers away and the nights out that with some have become sporadic. Despite everything I want to thank you a lot for having supported me and, above all, endured me in all our years of friendship. Then I would like to thank all the people I met in these years of university with whom I spent some beautiful years that have contributed a lot to make me feel Turin as a second home. Thank you very much my friends for being at my side in the pre-exam anxieties, during the lessons at the Polytechnic or remotely, for having shared the days to study, but above all to laugh and play, in the classrooms I and in all the support received during this thesis and over the years. There are too many names to mention so I prefer to remain vague so as not to forget and offend anyone, but I would like to thank anyone who is reading these lines for having been next to me in recent years and having helped to make me who I am today. And finally I would like to thank my thesis partner with whom I spent days working on this project with a lot of complicity, Ivana Orefice thanks for being the perfect server to my client.