POLITECNICO
DI TORINO

LINKÖPINGS
UNIVERSITET

Polytechnic of Turin & Linköpings university
Computer engineering department

Interval training gamification

a mobile app developed using Unity and mobile sensor data

Author: Giovanni Forcelli

Turin, 20^{th} April 2018

Foreword

Don´t hesitate or allow yourself to make excuses. Just get out and do it. Just get out and do it. You will be very, very glad that you did.

Christopher McCandless

Abstract

This thesis is a report about all the aspects that are involved in the development of a serious videogame. In particular this game has to catch the attention of the users in order to change their lifestyle. Gamification and its technique of challenging and rewarding the users for the result gained, has been the way used to let this happen.

Like all the new projects, the building process started with a deep study of the technology involved. In this case, the modern smartphone's sensors and the way how to exploit those better. The second step is the investigation of the other similar products on the market. The reason is that we can always learn from each other. It is both helpful to see the errors or the weak points, but also to catch the good ideas to stimulate creativity. The final stage is the development of the own product. It first has to go through fixed stages. The drafting, the development and the testing stages, in a sort of endless cycle, only stoppable when it is close enough to the wished goal.

The final result of this work shows that a new kind of product is ready to be sold on the market. People are usually wide open to try new interactions, especially if they concern the healthy prospective. Gamification is the key word for the future to boost the engagement of users and change their behavior or solve various kinds of problems.

Preface

The thesis work is the result of the ERASMUS+ project, an intercultural exchange between the university of Linköping in Sweden and the Polytechnic of Turin in Italy. It started in the summer of 2017 and ended in January 2018.

My personal interest in video games and running is the reason why this topic has been chosen. The binding of these two subjects is the perfect seal to conclude my academic career and to give me a right passport to enter in the working world. I never had any fear for the scope of this work, because since the beginning I have seen the high potential in it. What pushed me to the end has been the aim to helping people, especially kids. My wish is to have given a contribute for future works that will take care of the people's health.

The project has been led by two representatives belonging to the universities involved, Professor Andrea Bottino and professor Erik Berglund. I am glad to have worked with them and my very first thanks goes to them, who have given me the full trust, and they never have missed a commitment.

Contents

List of Figures

Chapter 1

Introduction

The first chapter of the thesis has been written to introduce the reader in the context in which the project has been developed, the motivation and the tools used.
Since the project consists in the production of a video game for mobile devices, there will be dealt with the market of such video games, the types of games available and other similar work.

1.1 Motivation

Today, nearly 1 out of 4 children and teens in developed countries are overweight or obese as reported here [5]. Those extra pounds put kids at risk for developing serious health problems, including diabetes, heart disease, and asthma. Childhood obesity also takes an emotional toll. Overweight children often have trouble keeping up with other kids and joining in sports and activities. Other kids may tease and exclude them, leading to low self-esteem, negative body image, and even depression. Children who sit too much and move too little are at the highest risk for becoming overweight. Kids need an hour of exercise daily for optimum health. This may seem like a lot, but exercise doesn't have to happen in a gym or all at once. It used to be commonplace to find children running around and playing in the streets of their neighborhoods, naturally expending energy and getting exercise. In today's world, that's not always an option, but you still have options for boosting their activity level.
One of the suggestions for keeping kids moving and healthy is to try activity-based video games, such as those from Wii and Kinect which are played standing up and moving around—simulating dancing, skateboarding, soccer, bowling, or tennis. Once the kid gains in confidence should get away from the screen and play the real

thing outside.
Across Europe, about 46% of children 9 to 16 own a smartphone, as said in a study published in the journal New Media & Society in 2015 [3]. The study surveyed about 3,500 children in Belgium, Denmark, Ireland, Italy, Portugal, Romania and the United Kingdom between 2013 and 2014. A possible cross way could be letting kids have fun with phones in outside activity. This is the guideline for the proposed game. The study is focused in the development of a new kind of game genre that is able to change the life habits of kids. One of the most important aspects is that it has been developed to be enjoyable and playable in safe places.

1.2 Background

1.2.1 Mobile game

A mobile game is a video game played, as the name suggests, on a mobile device. The best example of this kind of support is the smartphone, but nowadays tablet, smartwatch, PDA, portable media player or graphing calculator can be included as well.

The first world wide best known mobile game is Snake (1997). It was pre-installed in most mobile devices manufactured by Nokia, a number over 350 million. As all the software of that period, it was limited by the hardware. It was played with monochrome dot matrix graphics and single channel tones. The only way to play was by using the device's keypad buttons. Despite these limitations, the game industry started to put the attention on this market. The products developed and released totally changed due to the evolution of the technology such as the colored screen, touch input, polytone music and the internet support. The evolution was led by the growth of the smartphone spreading. A device able to solve many functions at a time, with on board a lot of sensors such as GPS, Accelerometer, Gyroscope, Proximity and so on. On the other hand, the very important revolution is the mobile connection to the Internet for everyone. It allows an easy seeking for new games on the market, but also to connect players all over the world and so improving the social aspect. At the end of the first decade of the 21th century the assortment for the developers exponentially increased leading to the birth of new genres.

Casual, social, puzzle, strategy, sport, RPG, action are just a bunch of the new genres created and it is even possible to mix up a couple of those to build a new one. The market has to satisfy the people that are heterogeneous and so their tastes. In the top list of the most downloaded games there are titles like Candy Crush (puzzle),

Clash of clans (war/social), Temple run (endless run), Flappy birds (platform), like a logical consequence of human diversity.

On the mobile markets such as *Play store* for android or *App store* for iOS, there are thousands and thousands of games. This is because even independent developers can make their own game with low budget and distribute it easily. There are different ways of marketing for the distribution including a targeting of the end user. The most used are:

1. Buy and play it all: the classic approach, sell on the market and when the user pay he got the all product;
2. Advertisement income: in the game are inserted adds who generate money every time a user play and visualize them;
3. F2P or free to play: the game can be played for free, but for getting special stuff or just speed up the progression, a payment with real money is requested;

Those strategies have built a dynamic market, able to challenge the other big sectors of the game industry. A few years ago, the video game was meant to be played on a big screen at home. The consoles were the status quo of the entire media. The spread of the smartphones and the low cost for getting a mobile game or developing it are the reasons why the gap with the console and PC gaming will be filled up quickly. The figure (1.1) is the forecasting of the income for the mobile game. The study from the newzoo web site [4] points out the great raising of the mobile market. According to them, the 2020 mobile game will take half of the total turnover.

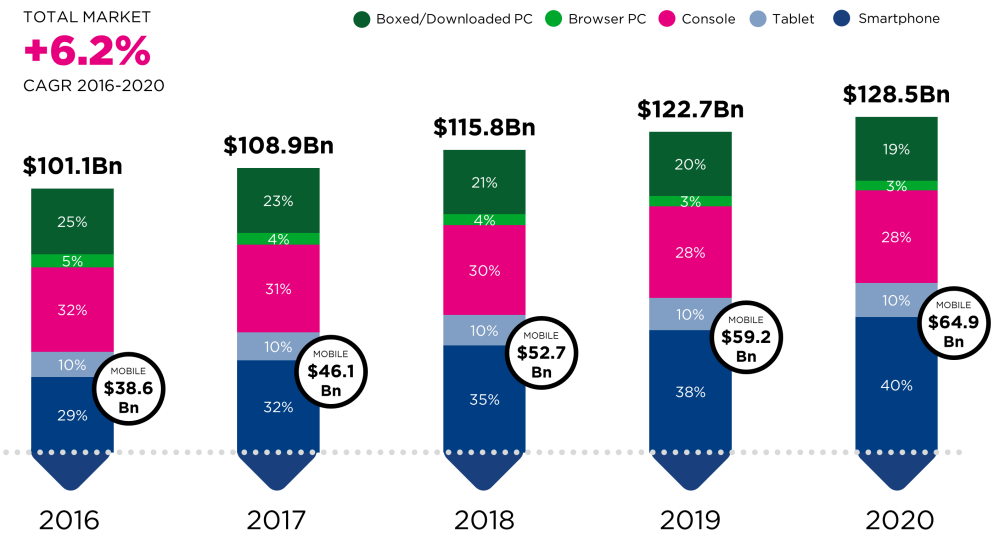

Figure (1.1): *Forecast for video game market.*

1.2.2 Serious game

The birth of this topology can be placed at the XVII century with the war simulation made by the Prussian army. It consists of a balancing of the entertainment and the education. The strength is to build a strong experience able to teach and retrieve the knowledge in a safety environment. The best example is the simulation of the reality in which the user can interact and make the decision. However there is no strict definition of the game genre and even for the technology, the support and the users who will play it. Depends on the goal of the project and the budget invested. The key point of the serious game is the learning by doing. It is the reflection of the fact that when you experience something it will forever stick to your living. It is a faster erudition compared with the facial lesson or the reading, the so-called passive learning. The good aspect of this approach is the internalization of the action by the repetition of the game. The user can get the needed confidence by doing the same action many times and then export it in real life. The major contest of application is the military teaching. By the simulation of a war or a warning situation, the soldier can learn directly how to manage his behavior and get confidence with the equipment. Even for the qualified army stuff is possible a simulation with the working instrument. However the serious game can be useful in many different applications. In commercial training for example the user will face an interview or a meeting in order to get all the tips helpful in a real conversation with a customer. A well defined field of application is the Agile Coaching, where the goal is the team building, creativity and innovation. An example is the Lego Serious play, that is played with LEGO blocks where the users are encouraged to use the mental and practical skills to get more confidence and passion for the work.

Figure (1.2): *A kit of LEGO Serious play.*

1.2.3 Exergame

The exergame is a genre that combines entertainment with the real physical exercise for the player. It is perceived as the reverse of the classic video game dynamic where the user just sits on a chair and plays. The goal is still the satisfaction but also the promotion of an active style of life.

In order to create this kind of game, technology makes the difference. The detection of the user movements is the first key point. All kinds of sensors are required for the detection of user movement. There is no standard one and of course they depend on the detection and the accuracy wanted for the game. Microsoft offers the Kinect camera that, with CV algorithms and IR sensor, is able to recognize the position and the movements of the users. Sony also uses a camera but for the PlayStation move system it also needs a controller with a light to identify the interaction. Nintendo has a motion controller with the Wii, but they don't need a camera and so are less accurate. The smartphone with Gyroscope, accelerometer and GPS is a good device for detection, even though it is difficult to create a good game for the screen size.

1.3 State of art

Creating an exergame is always a challenge because it must be suitable for as many as many kinds of users. So the development is a tricky process and the successful games are just few, like:

1. Pokemon GO: the augmented reality game for mobile of the well known Pokemon's trade. It is the most downloaded app in the first week of all time. Free to play with in-app purchases, it is also the most played game at the same time. It exploits the GPS data to show the Pokemon around the players, and the points of interest. The goal is to catch all the Pokemons and to train them. They are spawn all over the world, so it is necessary to go in a certain place to get the chance to take them. Other features are the eggs that will break only after an amount of steps taken.

2. Zombie, Run!: the world's best selling smartphone fitness app with four million players. The player has to run, driven by voice from the earphones, to avoid zombies. They are not real but they can be perceived by their own breathing that comes close if the user doesn't move. There is a background story that keeps the engagement alive and let the people follow real long term training. The version "Zombie, Run! 5K " has a story telling for 8 weeks, that covers a

training for a beginner to run a 5 kilometers race.

3. FS Ingress: is a massively multiplayer online game that let people move around the city. There is a fanta-scientific background in which the world is split in 2 groups: the resistance and the enlightened. They fight to get control of the portals that are built near the monuments of the cities. So the player must go near the point of interest to get the points for the game to win the final battle.

1.4 Game Engine

A game engine is an essential tool for a rapid application development (RAD) of a videogame. The basic functionality provided are the renderer, physic engine, collision detector, networking, sound, scripting, animation, IA and scene graph. For this reason they can be included in the category of most complex software. A good game engine is usually platform independent, it means that the game can be easily exported for many consoles with few or no changes at all in the source code. Usually a GPU abstraction is needed to work properly, and it is done by using graphic API such as Direct3D or OpenGL. Instead of having direct access to a hardware component, low level libraries like DirectX, SDL or OpenAL are used.

1.4.1 Unity

It is one of the most used game engines for independent developers, due to the fact that it has a free release able to cover many aspects related to the creation of a video game or other visual applications. It was created in 2004 by David Helgason (CEO), Nicholas Francis (CCO) and Joachim Ante (CTO) in Copenaghen, with the aim of "democratize game development". For this reason an active community supports it by testing new features, reporting bugs, proposing new tools and replying others' developer answers. As Kevin Murphy writes in his review [6] another important help are the tutorials often uploaded. There's also a live web training roughly once a month where everyone can follow along with a new topic and interact with the tutor. The Unity stuff knows that an asset store can improve the creation of games for beginners so they have created a well integrated search engine for their store. The Unity Asset Store is an advantage to any small teams or inexperienced developers who either can't afford or don't know how to implement certain features. Here can be found anything from 3D models to sound libraries, custom scripting solutions, level or AI editors, or even complete projects to poke around in and see how everything slots together in a "real" game. Most contents are made by the community and

will cost money, but there are often free demo versions of some assets, and Unity themselves often upload completely free content. The Unity player can currently export builds for over 20 different platforms – more than most of its competitors. Including the predictable Windows, Mac, Linux, Xbox, Playstation, Android and iOS, but also the Windows phone, WiiU, PS Vita, Android TV, and of course, all of the current VR platforms. In particular for this thesis will be used the support for android. The official guide advises that the Java development kit (JDK) correctly installed on the computer, the Android SDK and in some cases also the Android native development kit (NDK). Unity provides scripting APIs to access various input data and settings of Android devices. But for a better control and direct access to the native functionality it is mandatory to use the plug-ins written in Java. Those scripts can be written in any other IDE but must be always linked with the manifest file. An important aspect for mobile development is the texture compression. By default, Unity uses ETC1 for compressed RGB textures and ETC2 for compressed RGBA textures. If ETC2 is not supported by an Android device, the texture is decompressed at run time. This has an impact on memory usage, and also affects rendering speed.
There are many ways to test the project during the development phase.

- The integrated game preview, very fast and allows to switch different screen size and the orientation. Unfortunately it doesn't work if the project uses the native plug-ins. In fact in this case it will throw an exception and the code will not be executed. Another limitation is due to the fact that android devices have a touch interaction that can not be performed on the computer.

- Unity remote application, is an app that can be downloaded. It is a tool for helping mobile development. The app connects with Unity while you are running your project in Play Mode from the Unity Editor. The visual output from the Editor is sent to the device's screen, and the live inputs are sent back to the running project in Unity. This allows you to get a good impression of how your game really looks and handles on the target device, without the hassle of a full build for each test.

- The build release mode is the classic method that allows to create the SDK version that will run on the device. Before the building it is necessary to choose the scenes to be included and the version for the target device. It is the slowest approach but reproduces the real final version.

Chapter 2

Theory

The theory chapter has an important role because it deeply analyzes the subjects that are behind the project. The crucial meaning of the game for kids and adults in their life is treated as the first topic. There is space for a description of the gamification phenomenon, a new way for leading people.
The second part is an abstract on the interval training technique, the advantage of the practice and the possible way to play it.
Last session is dedicated to the tools available in the android devices for the developers, provided by the Google company itself.

2.1 Game & gamification

What is a game? Answering this question is not easy at all. The reason is that a standard definition of a game doesn't exist. Generally speaking a game is an activity striven for enjoyment. In this way also dancing, drawing, reading or everything that brings pleasure must be regarded as a game. However is not how people perceive the game and so the definition must be improved.
Something that a game always presents are components and rules, a sort of hardware/software approach. Both define the game and can exist independently from each other, but separately are not a game. For example a card game where the hardware part is the cards deck and the software part is represented by a set of rules, needs both two units. Otherwise a deck of cards is just a collection of figures with no value, and the rules are just a list of sentences.
Nonetheless the presence of rules and components doesn't guarantee all the features that describe a game. At least it must assure the following requirements:

- *Activity*
- *Freedom*
- *Equality*

Activity refers to the possibility of giving to the player to acting, not like most leisure activities that seduce people into passivity. Acting can concern different areas like spiritual, emotional or motor. Examples of spiritual acting are the planning, combining, concentrating and training of your mind. The emotional one involves the learning of the team work, accepting laws, learning how to lose and use fantasy and creativity. Motor area instead works with skillfulness and the practice of reaction. Equality is the aspect that makes the game an enjoyable experience for all the gamers because all of them start with the same possibility to win. This breaks the social boundaries and children can play with adults as equal partners. Freedom refers to the nature of the game to be a voluntary choice. The player is not forced or coerced to play. The player must agree to start and he has the faculty to retreat in every moment safely. This is because the real world and the game world are distinct, even though they share the same aspects. Both worlds are ruled by law, they give chances, build a competition and the destiny is unpredictable. Despite these similarities, the game time and space are limited so the play ends and the players can just carry back to reality the emotions. By crossing this small border between the real world and the game world while playing a game one is relaxed and can escape from the real world and then return relaxed and happy.

Games have a therapeutic power and can be used to heal people or to improve their involvement in some contexts. Gamification is born with this goal and so it takes the game design elements and uses them in real case scenarios. Gamification can have different intentions such as fidelity programs, boost positive actions and problem solving. In order to achieve these goals it exploits the human instincts of competition, social status, rewards and success. The gamification can be applied in different fields, in fact it is not limited only to social-business or web-engagement but can be used in all those occasions where the final goal is to communicate and spread a message. Below there is a list of possible fields of application:

- Healthy
- Education
- E-commerce
- E-learning

- Fidelity program and engagement
- Social learning
- Training

The game mechanics are continuing to change due to the new technology. Gamification follows the studies conducted on game design in order to exploit those techniques. However, there are fairly simple basic mechanics that are used to provide a stimulating and satisfying experience to the user through commitments, interest and participation. The basic mechanics of a game or gamified product are points, levels, prizes, virtual goods and rankings.

- Points / Credits - Reward: Points collection is a very powerful tool for increasing user participation. Even without a real value associated with the points, users are urged to take certain actions to increase their "treasure". By dividing these points into categories, it is possible to push the user to perform various actions and to have different attitudes. The points will be used to win prizes, so that the user will have the feeling that the time spent to earn points is time well spent.
- Levels - Status: The level is a means to classify the entire audience based on the score obtained, and very often also reflects real contexts such as work, school and social environments, where people are divided by hierarchically ordered classes. The level always introduces a new goal to reach, often through the accumulation of points. Each level reached results in access to certain unpublished privileges, which can be highlighted and reported in the user's profile. This process stimulates the user to commit himself more and more because he has a real visual return of his hierarchical status.
- Virtual goods - Self expressions: The accumulation of points must allow the user to win, exchange or buy virtual goods. The virtual goods are a fundamental element, in addition to encouraging the user to commit to buy the object, the user can use these objects to create their own identity within the community, personalizing their avatar and showing it to others users. The use of virtual goods can also be a valid method for the creators of the game to generate profit by proposing the purchase of virtual items with real money.
- Rankings - Competition: Rankings are the most effective way to sort and divide the use of a game. This subdivision can be based on various characteristics such as time spent, level, points and performance. The user will be

pushed to spend more and more time in the game to climb the rankings, monitoring step by step their progress and others. Competition is a very powerful mechanic because it is closely linked to the desire to become the best within a community.

The word Gamification was first used by Jesse Shell, a well known American game designer, in 2010 during the "Dice Conference" in Las Vegas. In the last 10 years, an incremental interest in this subject has been registered, due to the fact that the game devices have increased. The stereotype of the gamer is also changed, not anymore mainly a male between 13-33 years old, but also a lot of women. They now represent the 40% of the players and are becoming the new target for the revenue. This aspect must be taken into account for the future works.

Figure (2.1): *The gamification cycle.*

2.2 Interval training

Interval training indicates all the kinds of workout that are based on different heart beat zone intervals. That is translated in a training that alternates different workloads in the same session. The line guide suggests a heart rate around 60% of the maximum rate during the so-called rest phase, instead values around 80-90% are requested for the high intensity phase. Because of the high effort, the intense session can be performed for a limited time, usually no more than one minute. The repetition of the 2 stages is the key point of this technique. It activates different energetics systems, the aerobic one during the rest and the anaerobic for the other part. The constant variation helps to burn fat and gives benefit for the cardiovascular system. The high heart beat burn fat quickly and increments the metabolic process after the

work, the so called EPOC (Excess Post-exercise Oxygen Consumption).
The interval training can be used during the preparation of many sports such as running or rowing, because it is a concept not a fixed schedule. The only thing that matters is the alternation of intensity and not how it is stimulated. Other advantages are the time safety, due to the fact that on average the duration is even 50% less, or the pleasure, because it provides a mix of different exercises. On the Internet and among personal trainers a lot of variants are born like Fartlek, a Sweden workout with fixed distance run, walk-back sprinting, run and come back always from the same point, or High Intensity Interval Training (HIIT). As the words suggest, the HIIT is a very high effort training in fact during the practice the athlete has to work with his one hundred percent. The intention is to push yourself and try to raise the bar of your upper limit. Due to the expansive use of resources, this kind of training is adapted for big muscle stimulation, such as legs or chest.
An important article on the HIIT [7] points out that high-intensity exercise is brain healthy. Just as the heart adapts to high-intensity exercise by becoming stronger and capable of delivering more blood with each heartbeat, the neurons, or nerve cells, that lie in the brain adapt to the stress of exercise. Scientists discovered that when a person exercises, his muscle cells pump out a protein called FNDC5. This protein, by fragmenting into a second component called irisin, ramps up production of brain-derived neurotrophic factor, or BDNF. It's BDNF that stimulates the production of new neurons or nerve cells. It actually helps the brain build new architecture. Plus, BDNF protects nerve cells against damage, including damage related to aging. It's exciting that exercise can actually rebuild the brain.

2.3 Activity recognition API & step counter

The end of the first decade of the new millennium has been characterized by the fast spread of mobile devices and those are becoming part of daily life. This means that people always keep their smartphone with them throughout the day as they drive, walk, exercise, work and play. Recognizing what the user is doing can be exploited to better adapt the applications to the circumstances. For example could be smart if the device automatically starts to count the speed or save the path when a run activity is detected. Going in this direction, Google has developed an integrated system that is able to detect the activity that the user is playing among these one: Still, Running, Walking, Cycling, Tilting, Driving. The way it is done is by merging data belonging around multiple sensors, embedded in the device, and processing them using machine learning models. The API is made to help developers and save their time, in this case for example all the processing part is hidden and is retrieved just meaningful information. The notification when an activity is started

or finished, is gotten throughout a callback, implemented as an IntentService. Of course the detection could be wrong due to unpredicted data, so the statement is always correlated with a certain grade of accuracy.
Analyzing all the data in order to get derived information could cost a lot in terms of line of coding needed and consuming battery. For example, detect steps involve the data collected from the gyroscope, accelerometer and maybe GPS sensor. To have a low latency for the new input information required a fast computation of data and so high CPU. To solve part of this problem, from android Kit Kat 4.4 Android has introduced the low power step detector and counter sensors feature in the platform itself. They are similar but radically different. The step detector triggers an event each time a step is taken by the user, instead the step counter provides the total number of steps detected from the reboot. The first one has a very low latency but is affected by the false positive values. On the other hand the counter is more accurate because it uses extra time after each step to remove the false positive working with a model.

Chapter 3

Methodology

This is the chapter where a complete description of the game play will be done. Starting from the main idea of the setting, going through the artistic choices and the definition of the characters. A brief review on each screen view created will clarify all the study behind the scene. A full collection of screenshots is included just to better show what is mentioned in the text.

3.1 The concept idea

The game is a part of an educational video game series for kids that have to be developed further on. Right now there is another motion game that basically tries to let the player move his own body and training. The idea is to build a fantasy environment where the main character lives in a village and have to defend it and the inhabitants. In the motion game for example he has the duty to save the tree from bugs. The other games will lead the story on, and the main character will assume the role of hero. All the games have to be related, and progression is shared among them. Keeping the village safe is a hard task that must be pursued strictly. For this game the idea is that a disaster in the village has escaped all the animals. The community is lost without them, so the player has the task to go out and catch them. When this is done and the player has shown his skills, the inhabitants will ask him to take care of the creatures. These creatures are not animals as reality, but kind of fantasy pets able to evolve and mix each other. In this way the game can go further on forever.
For the thesis work the focus was on the core mechanics for the kids training that are the running stage and shaking action.

Figure (3.1): *The game's brand.*

3.2 The game

3.2.1 Art style

As shown in the figure 3.1 the style chosen for making the game is the pixel art. In particular a very low granularity pattern in a non-isometric projection. So the creation of the content can be sped up and keep the building process of the project fast. On-line there are a lot of tools that help in the definition and creation stages. For that usually independent game developers prefer this art in consequence of the faster creation of the textures. But in this case there are two other reasons that bring this to use. Due to the fact that the game will be played on the mobile, and there is not a standard device resolution and CPU specification, the pixel art is the preferred one. It is fast to load, because the image resolution is not that high but still provides good feedback. The non-isometric projection leads to an easier representation of the world. Instead the 3D view is a tricky issue due to the fact that all the parts of the object must be created, even though this game is not needed. The second, more trivial is that all the other games of the series share the same graphic. Changing the art style may lose the sense of belonging useful for the creation of a believable world.

3.2.2 Characters

The player during the game acts like an invented character whose objective is to catch as much animal as he can and complete the mission. His appearance is not well defined, but during the training stage he can be seen, and he appears like a black skin kid with yellow trousers and white the shirt. However the main role is played by the twenty different kinds of animals. These animals have been purchased from stockunlimited.com under a royalty-free license. They included pets like cats, dogs or bunnies but also large animals such as lions, tigers or elephants. All of them

are represented in a cartoon style and even the carnivorous one are nice and not scary at all. In order to make the set even bigger, every animal can have different shape colors. The five colors blue, green, red, yellow, and white scale up the set to one hundred combinations. A list of all the animals already met can be found and it shows the original shape, keeping secret the shape of the other one. This kind of visualization has been adopted to track the progress of the player and let him continue to play by intriguing him.

Figure (3.2): *An example of an animal and all its shape.*

3.2.3 Menu

The menu is the main screen of the game where all the sections can be reached. For the realization of this screen a lot of attempts were needed. The reason is that it is not only a place for redirecting the game, but also a concept art to spur the player into playing. Rhythmic music has been selected to give power and stimulation. The background color is light blue like the sky and on the bottom there is grass. This choice is done in order to transmit a sense of freedom and try to incentivize the player to go outside. The top part is the place for functional information. There are the buttons for access to the settings (3.2.4), the messages (3.2.8) and the animals gallery (3.2.2) pages. Instead on the top right there is space for an endless progress bar. It indicates the status of the GPS sensor and it will be visible till the accuracy of the sensor is 4% because the training can be performed only with fine precision. The button placed in the middle with the word "run" will be available only when that progress bar is deactivated.
In the middle part the run button already quoted is not always active, but depends on the GPS coverage, instead the other three smaller buttons are available all the time. Respectively they link to the animal gallery (3.2.7), the list of missions (3.2.9) and the shaking training (3.2.6). The last information readable in this screen is the actual level of the player and it can be found in the bottom part.

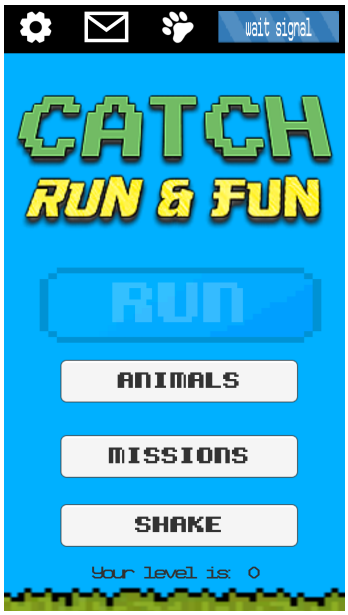

Figure (3.3): *Menu view.*

3.2.4 Settings

The screen for settings has had two aspects because they have been used differently during the development and for the final version. At the beginning, tuning the value for the algorithm was mandatory and having a controller in game was a simplification needed to speed up the tests. The parameters involved in the testing stage are: the total distance of the training, the time for running and the time far walking, all of them controlled with a own scrollbar. The first one, the total amount to be covered during the running phase, is really meaningful because it strongly impacts the heaviness of the training. In fact, covering a long distance of course required more strength. This distance is exactly the lower length that the player has to run to get the animal. The timing parameter instead influenced the kind of training wanted. Shortest walking period means less time to recover and so more anaerobic training. The same for the running period but exactly the inverse process. After the trial stage for finely calibrate the value and see how the algorithm is fast and precise in the detection, these parameters were no longer useful to be narrowly tuned.
The second life of the settings becomes a place for choosing the difficulty of the game directly by the player. Four kinds of steps are available, as shown in the figure 3.4. The tuned values are shown in the table 3.1 and are been chosen according to the interval training guide line.

Figure (3.4): *Settings view.*

DIFFICULTY	RUNNING TIME	WALKING TIME	TOTAL DISTANCE
easy	10s	90s	100m
medium	12s	60s	150m
hard	15s	60s	200m
extreme	20s	30s	350m

Table (3.1): *Details of each training mode*

3.2.5 Training section

This section is the most significant one, because it has to provide good feedback and try to stimulate the player. For the realization it has adopted a state machine approach in order to easily define a right events change. Can be counted seven different screens that cover the entire execution, from the beginning through the running and the walking stage, in a timed loop. The state machine used is strictly related to the settings chosen at the beginning, because they will influence the switch from one view to the next one. Now the seven states will be illustrated and analyzed.

The preview

As soon as the training session this stage is shown. It is a kind of recap because there is the total number of attempts available and already done. This is also a crossroad in the sense that only from here a loop of training can be started with the button "start" or the entire session concluded with the back arrow on the top left. The main point of this stage is the egg placed in the middle. It hides the animal that the player is catching. It is an integral egg in pixel art shape that is standing on the grass. It indicates the phases of the run due to its shell. In fact it has a different state, the closest to the end the user is the most broken it is. This ploy is thought to reveal the inside gift little by little. Keeping the animal hidden is a way to make the player curious and let him play until the secret is revealed. There are seven different textures for the external part of the egg, gradually more and more broken. The broken holes will show the animal inside but it is still covered by a colored texture that just underlines the border. The egg can be rotated with the finger and every time is touched it is squeezed and so the holes elongated.

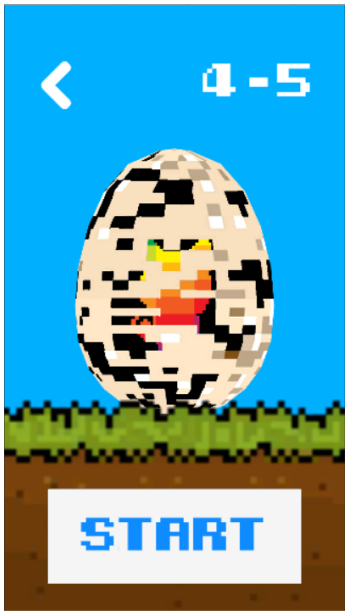

Figure (3.5): *Preview view.*

The countdown

It is a fast and intermediate view, in fact the main use of this scene is to give time to the player to get ready. It lasts a few seconds, in which a countdown is shown starting from three to go. At the same time the step counter is started, but nothing is visible. This counter will be used later to check the veracity of the run. A whistle announces the beginning of the next state that is the running workout.

Figure (3.6): *Countdown view.*

The workout

When the screen is displayed the game starts to count the distance covered by the player. This is the core view of the section even if it will be hardly seen. During this time gap the player is supposed to run as fast as possible to reach the goal established and so it is focused on running safely. What is still needed is something that is quickly perceived and doesn't need a long time to be read. Many tricks are used to pursue the cause such as the use of music. In particular a tune has been chosen from a track posted on youtube. This track [8] has been specifically created for HIIT training where thirty seconds of training are followed by thirty seconds of rest. The fragments used belong to the training phase and it also furnishes some hints like a rhythm change ten to go and a voice who counts the last three seconds. Nevertheless a visual stimulation is also desired. In the middle of the screen is present a round progress bar like a clock, that is filled radially. The round fill area is the simplest way to be recognized as a timer in order to provide a sense of speed up. It also changes color over the time, the transition from green to red indicates the nearing to the end. Last but not the least the background movement. As a backdrop is a gif image that shows a slider full of color. It is like a conveyor belt, with a color scale printed on it that alternates in an infinite loop. It has an incremental speed in order to motivate the player to run faster.
All of these technical measures are part of the study made in this thesis and have been chosen after a scrupulous analysis and different attempts.

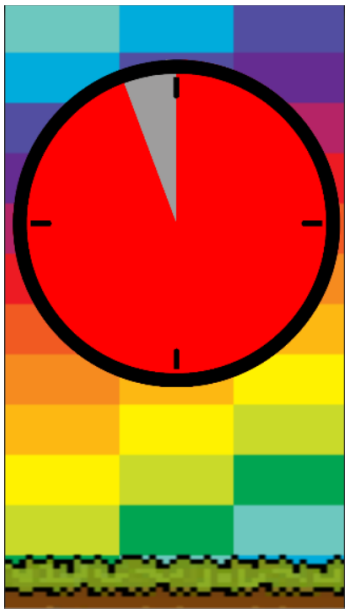

Figure (3.7): *Workout view.*

The result.

This is a crucial stage because the player has just ended to run and is waiting for the result of his efforts. Anyway the player wants to be rewarded and only by giving him positive feedback will he continue to train. Keeping this in mind, the outcome has been chosen after many trials. Tee end choice is a double bar to retrieve both informations about the actual run and the total distance covered from the beginning. The first bar is a vertical progress band composed by three stars one upon the other. It is filled by the last distance run and it is scaled by the total distance to be covered divided by the number of attempts available. So to fill it completely the player has to run at least one fifth of the length requested. The progression of the bar level is incremental so as the excitement linked to it. When the end is reached a yellow dots explosion is followed. It wants to simulate the breaking of the bar and it has a powerful effect on the attention according to the confetti style. All the dots, with a fairy music, fall down and are conveyed in a kind of big funnel. The funnel just told, is the second bar. It stores all the previous sessions and is filled gradually. Every run session gives his contribution to fill the bar and it can be easily detected by the different color used for each one. On the top there is like a finish line and when it is reached means that the player was successful.

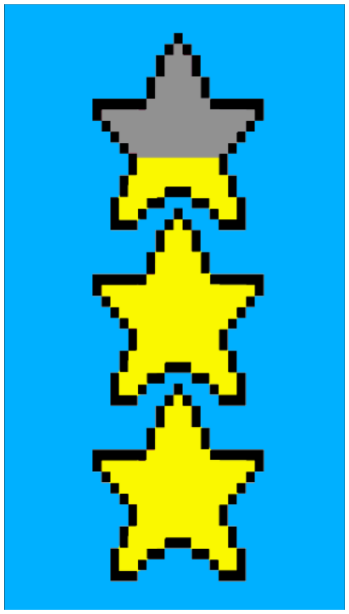

Figure (3.8): *Run result view*

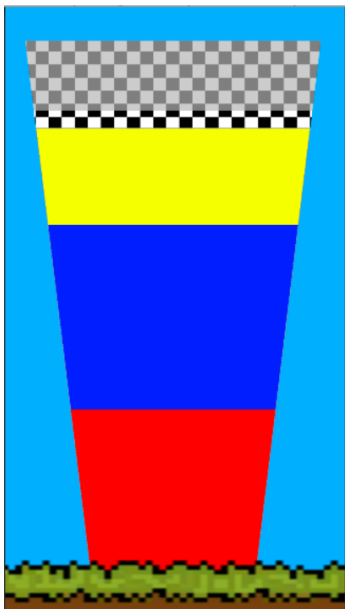

Figure (3.9): *Overall result view*

The price.

Here the egg is displayed again but is bigger and moves on its own. There is a spinning session in which it does several self turning with incremental speed. When the top speed is reached the egg changes its texture if a considerable distance has been run in the previous session. Instead if the total distance is covered, after the spinning part, the egg will completely break and finally show the animal contained inside. Now not only the nature of the creature is revealed but also the color and is possible to give it a name. After that it will be sent to the animal gallery and it will be part of the collection until its needs are respected.

The walking.

For a interval training the rest part is important such as the rest part. For an high intensity workout the recovery time must be spent not in a totally relaxed status. Keeping the body motion in order to avoid muscle strains and accustom the body to action. So for the game is important rewards the player who keep walking. First of all is displayed an animated guy who unceasingly walks to give an example of what to do. Second a time based progress bar is shown with a countdown on it, that is gratifying because the player can see the time expired. Third a complex system of rewarding is been adopted. If the smartphone detects the steps and the speed of movement is appropriate, on the screen are spread out fruits. The idea is to keep the player busy during the, let's said boring, phase of walking. The fruits are embedded in a sort of big soap bouncing bubble, and spread all over the screen. Each of them

has a time life, randomly chosen between ten seconds to fifteen. Their life cycle is punctuated by expansion and restrictions that alternate. Close to the end of their life these alternations are so fast that they seem to punch out of the screen. The player task is to collect them, and it is possible by tapping on it and because of the algorithm, the more he walks the more he has the fruits on the display. When a bubble is popped a vibration feedback is performed.

Figure (3.10): *Walking view.*

The outcome.

The final stage of the machine is used to a recap of the collected fruit. It is drawn as a panel table and is fulfilled one by one with all the detail numbers for each fruit. The total score in the bottom part gives a sense of charging, because of its decremental speed in counting the total. If all the fruits have been caught a big stamp and a great vibration are performed. The stamp is drawn on the panel and bears the inscription "perfect".

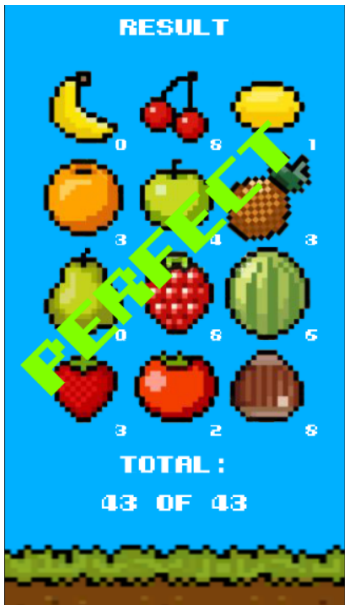

Figure (3.11): *Outcome view.*

3.2.6 Shaking section

This part has been created to extend the training for the kids in order to include the arms. Shaking as exercise warms up the muscles, organs and joints, and gives to the cells a buzz. Qu Gong, an old Chinese martial art, points out the good results of body shaking as a detoxified process in which all the fluids in your body flow quicker and more easily, as written in this web page [2]. The whole body will benefit from this activity, and the energy channels will be flowing smoothly and the arms will be warm and tingly.

The idea is to simulate a tree agitation in order to get the fruits fallen down from it, as shown in the picture 3.12 from the game. In order to perform the operation, the players must have earned tickets. This kind of item can be obtained during the training session, one for each repetition not used to catch the animal. Spending two tickets let access in a ten seconds long shaking operation. Three badges enable a twenty seconds performance, instead, the best price/duration ratio purchase is done by spending 5 for 30 seconds.

The process consists in a shaking session in which the player has to move the arm who holds the smartphone as fast as possible. The movement can be performed in all the 3D axis, because the reckoning is made by applying the square root on the tree spatial contributions, as shown below:

$$x, y, z = \text{acceleration on the respective axis}$$

$$\text{gravity} = \text{a parameter that simulates the gravity acceleration}$$

$$\text{gX} = \text{x} \;/\; \text{gravity} \quad \text{gY} = \text{y} \;/\; \text{gravity} \quad \text{gZ} = \text{z} \;/\; \text{gravity}$$

$$gForce = \sqrt[2]{gX * gX + gY * gY + gZ * gZ}$$

The *gForce* so obtained is the ratio between a given acceleration and the acceleration due to gravity. If this value is greater than the threshold chosen it means that a shaking is being recorded.

The algorithm works by counting the number of shaking detected after one second, for all the duration of the session. The value read is converted in a counter that will be used to spawn the fruits obtained. There are basically three ranges and each of those correspond on a different amount:

- $0 < \text{gap} < 3 \rightarrow 1$ fruit;
- $3 < \text{gap} < 5 \rightarrow 2$ fruits;
- $5 < \text{gap} = 1 \rightarrow 3$ fruits;

Of course if no shake is detected the game will generate no fruit, instead on the other hand the best performance is done when ninety fruits are gotten in the thirty seconds. During the shaking time it is impossible to watch the screen and realize what is happening, so another way to give feedback is needed. The vibration component has been adopted to advise for good execution. Every time a fruit falls down a short quiver is made by the smartphone and the user is quickly noticed.

Figure (3.12): *Shaking stage.*

3.2.7 Animal gallery

This part can be also called the exhibition room or the pet house because here all the caught animals can be found. They are shown one by one in a kind of slide list that starts from the first animal taken to the last one in time order. Basically on the screen one at a time the animals are represented with all the informations linked.
The given name of the animal is widely written on the top of the screen and just below the animal is drawn with the same color shown when captured. Here it is retrieved and additional information related to the hungry status. That is shown by a progress bar, put just under the animal, with a value between zero and one hundred. An animal can be good if it is well fed or it is sad and weak if is not fed enough.So three kind of situation are allowed:

- good: the score is over than ninety;
- starving: the score is lower than fifty;
- stable: all the other possible value;

The animals build a kind of dialogue with the player, in fact they greet him using welcome messages picked from a preset and after that they display their health condition. Even for this situation a preset of speech is stored, and it also includes thankful words. Those are used when a fruit is given to the pet.
The most important features in this part is the feeding system. Here the player has the important duty to take care of his creatures, trying to keep a good level of satisfaction among them. All the animals, even the carnivorous one, eat fruits but indeed they are not all the same. Actually, each of them has a list of preferred foods or detested one. The effect of feeding them with those fruits is evident, a right fruit will increase the score by ten points instead a wrong one will decrease by five units. The animal is not passive, but will notify if it likes the food provided or completely detest it. Not only the progress bar will change but a message will be shown and the color of the eating effect will have different color. Green is for a loved food, red for wrong one and white for an unconcerned. The third category of food is the non category, in fact a not loved or not hated food, is a neutral one. It will increase the score just by one and so the impact is not really effectiveness.
A drag and drop interaction let the player feed his animals. The hard task for him is to finely manage the fruits, collected during the walking stage between two run sessions or by shaking the tree and keeping the score always over zero otherwise the animal will run away.

Figure (3.13): *Animal gallery.*

3.2.8 Messages

To make the game more realistic and try to improve the engagement, the animals' hungry points always decrease by the time. It is set that every half hour a unit of that score is inexorably lost. For example an animal with the full score takes two days to decrease till four. If the player doesn't care and doesn't give it food, after two hours it will get away. If an animal escapes it is lost forever and so it is impossible to get it back. This section is useful to have a quick review of the farm. This part can be considered as a mailbox where all the animals can leave a meaningful message among these kind:

- happy: if the animal is really well fed, a value greater than ninety;
- angry: if the animal is starving and needs help, score lower than thirty;
- left: when the animal notice its forsake;

Figure (3.14): *Message screen.*

3.2.9 Missions

Missions are intended to be the background challenge that encourage the player to keep catching. A mission is a request of animals made by an urban zoo from all over the world. This request is composed by the name of the city where the zoo is supposed to be, an image which shows that place, the animal type wanted, the color requested and the amount. These two last fields can be filled with null and it means that all kinds and colors are accepted in that mission. It is also possible to just fill one of parameters to limit the request for a selected kind of animal not caring the color or the exact opposite.
The user interface (figure 3.15) present the list of all missions available in that moment on the top and below the list of all animals who fit the description. When an animal is picked the background changes color in yellow and the partial amount in the mission label is updated. When the correct number of animals is selected the bottom button will enable to send those animals to the zoo and both the mission and the animals will disappear forever. Every mission completed raises the level of the player by one.

Figure (3.15): *Missions view.*

Chapter 4

Result

In this chapter are presented the results gained and the way how they have been reached. A rich analysis of the techniques used and the procedure as they have been used in the project, underlining the technical choices. It is also possible to find in this chapter the attempts that have been done but are not a part of the end outcome. All the ways and the efforts done are relevant to the pursuing of the final goal.

4.1 The first attempt: activity recognition

One of the things that give the input for the creation of this game, is doubtless the awesome result claimed by Google with the activity recognition API. The ability to detect the user activity with low consumption of battery seams like magic, as also they write in the documentation of their application FitBit that use those API.
On the Internet are shared a lot of android projects based on the usage of this new support launched by the company of Mountain View. The integration of the library in a personal project is quite easy and a lot of tutorials are retrievable on-line. However for a Unity game it is not possible to be made directly, but is mandatory to use a plug-in compiled in Java or something similar. Nevertheless the potential claimed was really big and the impact on the game play as a consequence was high too. The initial idea was to create a kind of forest simulator in which the user can walk freely. During the walk of course he will encounter some wild animals and with a notification he will be informed of that. As a first reaction he would like to run and try to catch it. So the game must be able to quickly detect the actual user activity.
So moving in this way, the first experiment was to test the Android library in order

to see the field of application. For this reason a project from scratch has been created following the tutorial on code.tutsplus.com [9], that in just three steps points out how to use the activity recognition. A brief overview:

1. First thing to do is create an Android project, import Play Services 8.4 at least in the gradle file and include the permission for the API in the manifest. After that it is mandatory to create a class that extends the IntentService because it will allow you to perform the application logic in the background.

2. In order to use the Google Play service it is necessary to connect with them. This can be done in the main activity of the application and implement the ConnectionCallBacks and OnConnectionFailedListener interfaces. By using the OnCreate() function and asking for the required API a listener will be instanced. The communication will use the pending intent created before and it will retrieve information after a certain gap time indicated by the developer in the apposite method.

3. Once received the intent it has to be analyzed. The data available show the confidentiality expressed with a percentage on each possible activity detectable. If a confidence is 75 or higher, then it's safe to assume that the user is performing that activity.

Following all these points, the Android application was ready to be tested. The first checks were done inside the building where the office is placed. As contour informations must to be count that the GPS signal was low, because the data retrieved were affected by a 48% grade of accuracy.
The test consists in the evaluation of the detecting time for a new activity and the accuracy of the type activity detected. Unfortunately the first test was far away from the result expected. Inside the building the activity detected with more confidentiality was on_vehicle. The highest value was 79%, instead the others accuracy were always lower than the activity on the wheels. Those results may be related to the GPS jump effect. In fact with an other application running on the smartphone at the same time, the user position were monitored and this impression was partially confirmed.
As first result it was not very encouraging, and so the game drastically changed the target area for play and it is been re-evaluated only for the outside.
The second test has been taken in the university courtyard, where the geolocation works better. In this new settlement, the detection of the activity improved a lot. Onfoot and run activity were finally recognized, with pretty high accuracy. The top value obtained was 92% for the walking activity detection, linked to the run confidentiality higher than 75% when the user actually ran. Despite the good outcome,

there was an aspect that ruined all the hopes. The time to detect the activity was too high, on average 6-7 seconds, but the data at first were requested each second. In order to enhance the results, a lower latency in the request was set and the test redone. The new time gap was fixed to a hundredth of a second. The accuracy of the result decreased and the time for the detection didn't change that much. A value with lower accuracy is fated to be less reliable.
After many tests and tuning operations, the decision was not to use those libraries. The game must be accurate and fast in the detection at the same time. The activity recognition doesn't fit this purpose, even if they are really a powerful tool for other, maybe less responsive, applications.

4.2 The second attempt: GPS sensor

After the failure of the first way, another path has to be chosen. The GPS sensor comes out as the easy and the only practicable direction to follow. This sensor is intended to add rich location and motion capability to the application.
The Global Position System explained in a few words is an infrastructure made by satellite and sensors that can identify the exact location on Earth. It is a system build in the 60s by US Department of Defense to get an all-weather, accurate, positioning and navigation definite procedure.
It works with the trilateration principle based on the time. The sensor collects the data that travel on the radio waves sent by the satellites. For a right result it has to combine four different data in order to get the latitude, longitude, altitude and time.

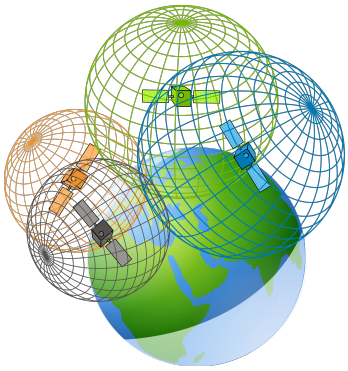

Figure (4.1): *GPS trilateration procedure.*

The data used from the user sensors are not the most accurate one, because they are reserved for military purposes. It is worth bearing in mind that 38 microseconds are enough to have errors in the order of kilometers. The errors are due to the relativity, the obstacle and the condition of the transmission medium. Knowing the limits and

the boundaries of the system, a possible way to use it is the next effort taken.
There are two ways to make it work, both start with the method Start() from the object input.location, but it first checks the status of the service. As parameters it is possible to insert the desired accuracy in meters and the update distance in meters. If an accuracy less than 500 is required, Unity will automatically add the ACCESS_FINE_LOCATION permit in the manifest, instead in the other case the ACCESS_COARSE_LOCATION. Another way to make it works is the use of a coroutine, in an infinite loop asking for the new data.
Unluckily both of them present an issue and for the purpose of the thesis is unacceptable. The problem is the not direct control of the sensor data and setting. This leads to other related disadvantages.

- **Rate.** The updating of the data is not constant, it means that all depends on the accuracy.
- **Accuracy.** There is not a direct way to check the accuracy, and it makes it difficult to make decisions based on it.
- **Network.** The data retrieved are not mandatory from the GPS sensor, but they can be taken from the network. Usually are from the closest WI-FI router available. In that case the position is not fine and takes time to be updated. There is no way to enable this feature.
- **Speed.** Derived measures such as speed, must be calculated, increasing the computation time, the code and the data that must be stored.
- **Battery consumption.** It is not possible to set a time gap directly for the request of the data. The use of a coroutine will impact the game speed, because it runs in parallel with the game code and steals power.

These disadvantages can be avoided by using the native function to retrieve the data sensor. The only way to integrate a Java piece of code is with the plug-in. In order to bind the two scripts it is mandatory the use of the right library and class. For the android side, written and compiled with Android Studio, there is a library to be imported in the project. This is unity3d.player, and lets the developer define a class that can be called from outside, such as Unity, that receives a string as only a parameter. For the reverse way it is possible to use the method UnitySendMessage that needs three parameters. The first one is the name of the class intended to receive the call, the function belonging to that class that will be called and the string to be passed as input. For the thesis the Java plug-in has been used as the main interaction with the GPS sensor, because of the benefit introduced. In particular three specific instances of GPS interaction have been created at different

rates. This is because the highest rate will consume the battery easily, so it is not important to keep it high. For example, before and during the training session there is the alternation of the all different rate. In order to unlock the start button for entering the stage, a certain level of accuracy is required. The purpose is called the GPSbackground service. It has a rate fixed to two seconds and retrieves the accuracy of the sensor in that moment. When the training session is started the background service is no longer requested and so it is stopped. The GPSrun service is activated during the countdown before the running view and it is characterized by a location interval equal to zero. Having as interval zero, let the device send all the data as soon as it is detected. In this way all the changes are detected and the rate depends on the sensor detections. Potentially it is really costly and for this reason is just used in the more sensitive part and nothing more. In this situation the plug-in sends the accuracy value plus the distance from the previous point. Once they are received by the game they will be processed and used for the counting of the distance. For the last part of the workout cycle is necessary to detect the walking activity, and so the precision of the distance covered is not needed. What is useful is the speed of the user, he should move with a speed between 1 and 1,5 meters per a second. The rate set for getting the information is one second and half, that is a compromise for the accuracy of the detection and the frequency.

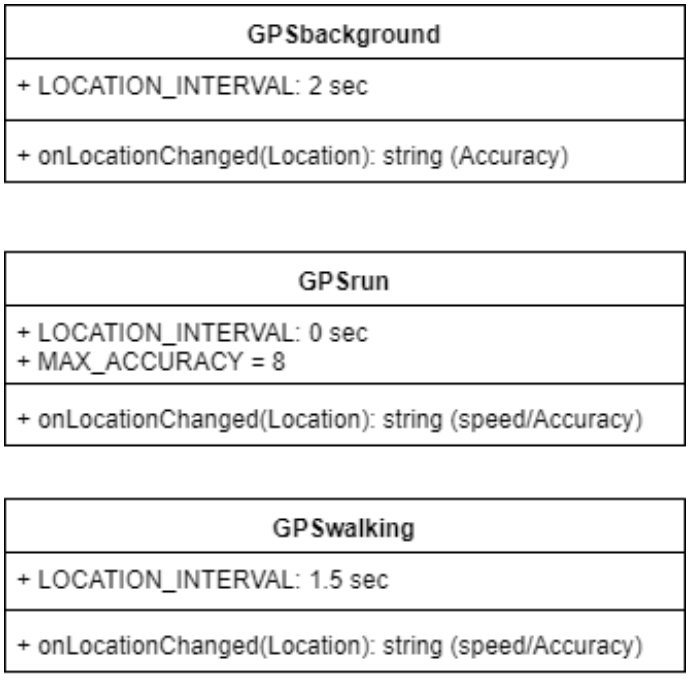

Figure (4.2): *An UML view of the services.*

The accuracy for the retrieved data is an important factor. It influences the measurement and can help to better analyze the information. The value blinded to the detection is like a radius for a circumference that is drawn from the GPS location given. The higher it is the more uncertain is the actual position. For example, if the

accuracy is 10 it means that the user could be located in an area spread over 314 square meter. It is too much for the game purpose because, as saw in the section 3.2.4, the maximum distance to be covered is 350 meters. Taking that value as minimum accuracy required will fail all the game, due to the big mistake given by the sensor. A good estimation for the sensor precision is 2. This accuracy is easy to get, in fact outdoor usually is the standard. Precision like 1 is quite impossible to be obtained because the phone uses the non military one and so less accurate. Instead 0 accuracy means that the sensor has some problem or is not ready at all. After all these considerations a test is necessary to verify if they work as expected. The first check has been done inside the building, with a low accuracy, more or less 48. This kind of situation is the worst scenario possible, because, as said before, there is the jumping phenomenon. This event leads to a wrong estimation of the overall distance to be covered. The inside detection still remains a critical point and there is no way to improve the effectiveness, because the solution could be the WI-FI spot, but it retrieves the static position of the device and not the user one. The second session of the test has been made outside, in the University courtyard where it is easy to get a high level of accuracy. Here a series of analyses have been conducted on different footages. Since there is not an athletics track with marked distances or something else, the only known distance gap is the interval between two lampposts of light, the measurement has only fixed size. That is 15 meters, and the tests have been done on three different sizes: 30m, 45m and 60m. The results are shown in the table below.

DISTANCE (m)	**TEST 1 (m)**	**TEST 2 (m)**	**TEST 3 (m)**
30	30.1	30.2	30.0
45	44.7	44.9	44.8
60	59.4	58.8	61.1
90	88.9	90.3	90.7

Table (4.1): *Details of each test session*

As the table (4.1) shows, the results are successful, the maximum gap from counted is 2%, but the average is less. This fact leads the project to next stage.
The float value for the distance retrieved by the sensor must be dealt in some way, the function used to treat it is the Mathf.Round(). It will cut the decimal part by rounding the value depending on the size, and will return an Int.

4.3 Anti-cheating system

It is intrinsic in each of us to search for the easiest way to achieve goals. Often it is misunderstood with the wish of cheating and taking advantage of the shortcomings. Saving time, getting higher results, getting special items or, more generally, winning the game are the reasons why this practice is followed. The role of the developer, mainly the game designer, is to put the player in the condition that cheating is not profitable. It can be done by decreasing the score or giving some penalties to the player that will influence the end result, but of course the primary way is to avoid points where it is easy to pursue the wrong path. Usually the gamer is aware of the limit for a device and so they try to exploit the points of possible failure to get the advantage. The problem is even bigger because the players are young, smart and more competitive. Their ability to quickly analyze the game and compare it with the others let them be highly responsive to the detection of the point of failure. The desire for victory often is tempting and make the player blind to the true spirit of competition.
For the game developed in this thesis, the wishing to cheat could be justified, because it would save the physical effort of the player. Running or just walking are activities that need real fatigue that is not equal for everyone. It depends on the grade of athletic training, age, body shape and so on. Balancing all these variables is impossible, but the introduction of the different levels of challenge let the player think about the kind of challenge that fits. Except for the scenario that a player chooses a lower difficulty level for his need, the other ways to make the game easier are just a few. The vital points are still the run and walking stage. The new game Pokemon GO! has shown that a way of lying is by using a car and driving it slowly. In this way the device considers the moderate movement like a slow-paced activity. In order to give less chances to fake, the game has been developed with a cross checking for both the crucial parts.
During the walking session it is not a big deal if sometimes the player can fake the system, because the reward is not valuable. However if the whole phase is subject to cheating the purpose of the game is not pursued. In fact the player has to walk, because in this way he can recover and slow down the heart beats. The player speed could not be that high but he can't even stand still. The control set for avoiding most of the issues is on the speed. The system provides the speed for each detection. In this scenario it is possible to choose the relevant information and the one out of range or faked. The speed required must fit the standard one, that is between one and, one and half meters per second. The report on the fidelity of the data is still sent to the game and it is taken in count for the analysis. The running phase, the key one, has a high control for acquiring data. Every time a new location is estimated by the device, the plug-in code checks if the accuracy is less than a

maximum hard coded, in this case it is 8. Once controlled, a distance is calculated and it is sent to the game related to the accuracy grade. This information could be used for further computation inside the game, as different weights are given to the distance estimations for the total summation. At the end of the running interval it is done for the last verification. It is made upon the number of steps caught from the start of the workout. It is necessary that at least one is done, because the step counter loses sometime. Of course if no steps are identified, but a certain distance has been covered, it means that with high possibility there is a wangle attempt. In this case the running session will be lost and a toast message is shown to warn about the problem encountered.

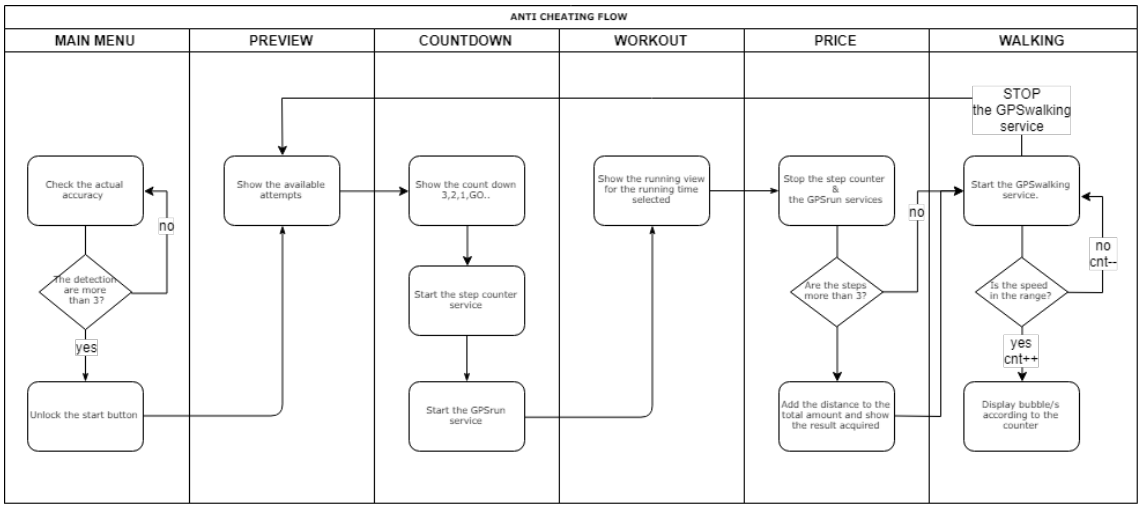

Figure (4.3): *The anti cheating flow chart.*

4.3.1 Step counter

Boosting the control with the data from the step counter is a way for a more accurate and truthful detection. Android provides an integrated sensor that is able to provide an indication of the movement done by the user. The reaction time is slow, it takes between 4 or 5 seconds in average to update. The delay is due to the computational time and the interval necessary to have discovered a pattern. Google, the developer of the feature, doesn't provide a full picture of the components that play the role of the detection and neither how it is done. For sure the accelerometer plays the main role, because it is able to retrieve the information about the force over the three axes. By computing those force it is possible to get the movement over the time and the space for the smartphone, that is supposed to be handled to the user.
One of the intermediate goals of the thesis was to develop a self made step counter. Usually the studies of step detection based on accelerometers use a certain threshold, peak detection, auto-correlation or spectral analysis. For the standardization of the problem it is easy to consider a pre-fixed position for the sensors and so for the

device. Another simplification is to consider just a range of use such as the running or the walking. Both of those restrictions made the algorithm too restricted and limited. For a better result it is necessary to have an adaptive threshold able to change over time. It has to dynamically analyze the time based evolution of the acceleration magnitude and decoupled in peak and valley the evolution over the time.
The implementation for the purpose has been inspired by the research work named *"Step Detection Robust against the Dynamics of Smartphones"* [1]. The proposed algorithm extracts the data candidates, and after a magnitude and temporal filtering it takes the decision.

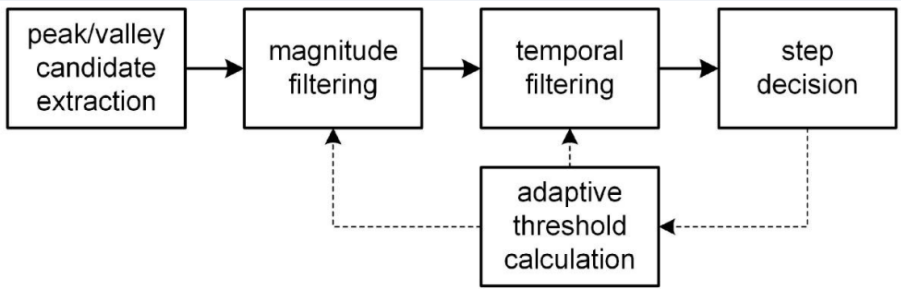

Figure (4.4): *Logic flow for the step counter.*

Peak and valley candidate detection

Set a fixed threshold is reductive, because the movement can stand in others values, depending on the orientation of the device or the actual action done by the user. A possible way, to easily change the boundaries, is a constant computation on the latest values, as show below:

$$\mu_a = \frac{|a_p| + |a_v|}{2}$$

Where a_p and a_v are the accelerations in the most recently detected peak and valley. Then the μ_a is the most updated average magnitude. The restricted data used retrieve a time based variation of the average. It shouldn't change that much for the following data. Based on the observations of the step average, the type of n-th acceleration sample is determined using adaptive magnitude thresholds, as follows.

$$peakcandidate, |a_n| > max|a_{n-1}|, |a_{n+1}|, \mu_a + \frac{\sigma_a}{\alpha}$$
$$valleycandidate, |a_n| < min|a_{n-1}|, |a_{n+1}|, \mu_a - \frac{\sigma_a}{\alpha}$$
$$intermediatesample, otherwise$$

Validation of peak and valley candidates

The selection of a right candidate could be a matter of time. Usually the other algorithms use a certain gap to validate a value. This way presupposes that a fixed time has been chosen and it is independent of the actual user activity. Here the goal is to detach the fixed parameters. The solution suggested takes in count the history, in particular for the peak validation, the time distance to the recent peak using the following threshold:

$$Th_p = \mu_p - \frac{\sigma_p}{\beta}$$

Where the μ_p, σ_p and β represent the average and the standard deviation of the time interval between adjacent peaks in the magnitude of acceleration and a time scale constant, respectively.
The same mechanism is proposed for the evaluation of the valley. Of course, to get a less restricted boundary, it is possible to calculate the standard deviation for the recent M peaks or valleys.

Proposed Step Detection Algorithm Based

The proposed step detection algorithm is described using the notations in Table that follows as in Algorithm 1, which utilizes the functions in Algorithms 2, 3 and 4. Algorithm 2 describes the detection of peak and valley candidates for incoming acceleration samples. The type of the acceleration sample an (Sc) is determined using the adaptive magnitude thresholds in Equation (2) and returned. Algorithm 1 is executed for every incoming acceleration sample. It determines the state of an (Sn) and current step count from the inputs consisting of the state of an-1 (Sn-1), recent step count (count) and an+1. During the initial startup of the algorithm execution, only a peak candidate can be determined to be a new peak as in Line (1), and any valley candidate is discarded. If a peak candidate found in the valley state (Svalley) satisfies the adaptive threshold for the peak interval, it is determined to be a new peak, and the parameters related to peaks are updated as in Line (2). If a peak candidate in the peak state (Speak) is close in time to the recent peak and its magnitude is larger than that of the recent peak, it replaces the recent peak as in Line (3). If a valley candidate in the peak state (Speak) satisfies the adaptive threshold for the valley interval, it is determined to be a new valley, and the step counter is increased by one as in Line (4). If a valley candidate in the valley state (Svalley) is close in time to the recent valley and its magnitude is smaller than that of the recent valley, it replaces the recent valley as in Line (5). Algorithms 3 and 4 describe the processes after finding peaks or valleys. The statistics of the time interval between adjacent peaks (or valleys), the occurrence time and the magnitude

of acceleration are updated for every peak or valley found. The step average μ_a is updated for the first peak or valley right after the recent valley or peak as in Line (2) or (4) of Algorithm 1, respectively.

Algorithm 1: StepDetection$(\mathbf{a}_{n+1}, n+1, S_{n-1}, count)$

$S_c \leftarrow$ DetectCandidate$(\mathbf{a}_{n-1}, \mathbf{a}_n, \mathbf{a}_{n+1}, \mu_a, \sigma_a, \alpha)$
$S_n \leftarrow S_{intmd}$

if $S_c = S_{peak}$
- **if** $S_{n-1} = S_{init}$ **then** { $S_n \leftarrow S_{peak}$; UpdatePeak$(\mathbf{a}_n, n)$ } (1)
- **else if** $S_{n-1} = S_{valley}$ **and** $n - n_p > Th_p$ **then** { $S_n \leftarrow S_{peak}$; UpdatePeak$(\mathbf{a}_n, n)$; $\mu_a \leftarrow \frac{|\mathbf{a}_p| + |\mathbf{a}_v|}{2}$ } (2)
- **else if** $S_{n-1} = S_{peak}$ **and** $n - n_p \leq Th_p$ **and** $|\mathbf{a}_n| > |\mathbf{a}_p|$ **then** UpdatePeak$(\mathbf{a}_n, n)$ (3)

else if $S_c = S_{valley}$
- **if** $S_{n-1} = S_{peak}$ **and** $n - n_v > Th_v$ **then** { $S_n \leftarrow S_{valley}$; UpdateValley$(\mathbf{a}_n, n)$; $count \leftarrow count + 1$; $\mu_a \leftarrow \frac{|\mathbf{a}_p| + |\mathbf{a}_v|}{2}$ } (4)
- **else if** $S_{n-1} = S_{valley}$ **and** $n - n_v \leq Th_v$ **and** $|\mathbf{a}_n| < |\mathbf{a}_v|$ **then** UpdateValley$(\mathbf{a}_n, n)$ (5)

update σ_a
return $(S_n, count)$

Algorithm 2: DetectCandidate$(\mathbf{a}_{n-1}, \mathbf{a}_n, \mathbf{a}_{n+1}, \mu_a, \sigma_a, \alpha)$

$S_c \leftarrow S_{intmd}$
if $|\mathbf{a}_n| > \max(|\mathbf{a}_{n-1}|, |\mathbf{a}_{n+1}|, \mu_a + \frac{\sigma_a}{\alpha})$
  then $S_c \leftarrow S_{peak}$
  else if $|\mathbf{a}_n| < \min(|\mathbf{a}_{n-1}|, |\mathbf{a}_{n+1}|, \mu_a - \frac{\sigma_a}{\alpha})$
  then $S_c \leftarrow S_{valley}$
return (S_c)

Algorithm 3: UpdatePeak$(\mathbf{a}_n, n)$

update μ_p, σ_p with $|\mathbf{a}_n|$
$n_p \leftarrow n, |\mathbf{a}_p| \leftarrow |\mathbf{a}_n|$
return

Algorithm 4: UpdateValley$(\mathbf{a}_n, n)$

update μ_v, σ_v with $|\mathbf{a}_n|$
$n_v \leftarrow n, |\mathbf{a}_v| \leftarrow |\mathbf{a}_n|$
return

Figure (4.5): *Pseudo code for the algorithm.*

The K value, that influences the sampling rate, does not change the accuracy for value larger than 10 Hz. In order to get a good detection, with a percentage over 90% it is enough a rate of 8%. This level of accuracy is comparable with the embedded stem counter. Both have a high precision for the detection, but the one proposed is faster in showing the value. On average for a 30 seconds walking session the deviation is just 2 steps. For the thesis purpose, it is important to detect if at least one step is detected and get the trustability of the action. The only problem encountered is the battery consumption. Even decreasing the samplings, there is an expending of energy. The sampling power consumption contributes to the overall power consumption greatly. The power consumption for the algorithm execution is less than 11% of the overall power consumption.
Due to the fact that when the step counter is needed, the application already asks for a high number of samplings from the GPS sensors, introducing an huge loss of battery, the algorithm has been shelved. The great benefit of the accuracy doesn't overcome the limitation on the battery life. However The step counter by Google runs always in the background, even if no applications are waiting for the information. The last decision is to use that one and let the smartphone stand longer. More battery means more time spendable outside for fun.

4.4 Shaker

The goal for this thesis is to make the players move and do exercise. The main interaction, as saw from the beginning, is the interval training. The kind of sport related to that training is running. It permits to improve the resistance to stress with a lot of benefits to all the body parts. Unfortunately it is not always possible to go outside and run. There are hundreds of reasons to stay home, and in those situations the game will be useless. A new kind of game feature was needed to face this limitation. The introduction of the animal gallery, where it is possible to feed the animals, is an expansion to the game and can entertain the player for a while. However that doesn't pursue the aim of the thesis. Using the accelerometer it is possible to analyze the forces on the device. This could be exploited to create a shaker.
The meaning of the shaker is to let the user move his arms in order to increase his heart rate. The fast movements requested easily speed up the heart, and they can be performed everywhere. The only precaution is to stand aside obstacle or people, because it is dangerous due to the quite uncontrollable arms shake.
Using the accelerometer data on the three axis is the way to build a system able to detect the shaking. The implementation has been described in the section 3.2.6. Actually the first algorithm tested was the so-called balloon. It is requested to move

the smartphone keeping a certain speed, but the information on the right action is obtained by the time. The first part is the filling followed by the maintenance. If the user stops the reached level starts to decrease to zero. There are 3 areas with different thresholds. The flow is shown in the figure 4.6.

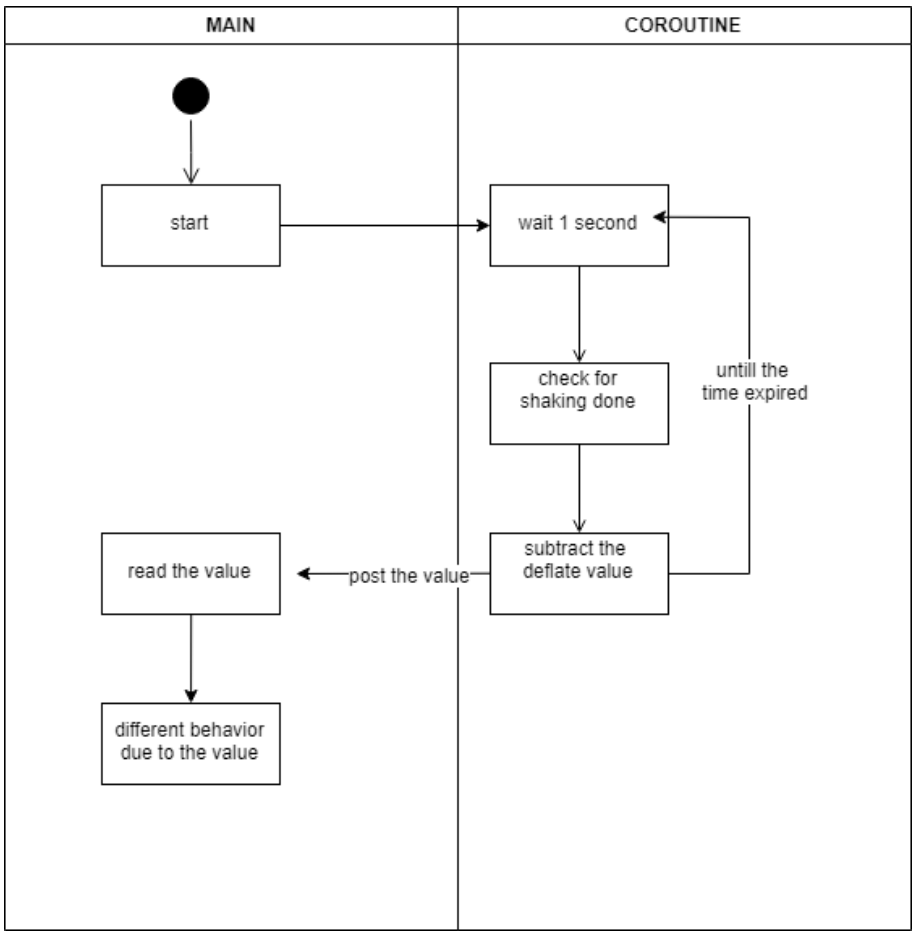

Figure (4.6): *The balloon algorithm.*

For the feedback has been used the color shown on the display. it changes from red to green when the right area is reached. Furthermore a different vibration pattern is played in each area from the slow pace to a continuous one .

Chapter 5

Conclusion

In the last chapter is the part where the overhead view took place. There is space for some analysis about the game developed and the possible influence on the global market and new trends. As a thesis work the game still needs a lot of improvement to make it a full product. So an entire paragraph is spent in subjection for coming students who will work on it.
The final part is a recap of the work made by me and the ability I got during the entire experience.

5.1 The possible impact

Serious games can be fun, training has to be rewarded and success must be shared. This must be the starting point of an educational game. Serious game is the kind of game that the family has to prefer for their kids. Despite the definition that sounds boring, the game, if well constructed, can fulfill the enjoyment for the player. In fact potentially everything can be called a game, but not all is fun. A game must be fun and help to escape from the real world. Even a serious game can pursue that, all depends on the idea and the realization. For the realization of a game, a cycle of tests and changes is necessary, because not everything works as planned. When a good game play is found and it is easy to be learned and performed, the most significant concern must be the polish.

Exergame is the new frontier for mobile games. The high number of sensors embedded in the devices have to be exploited to enhance the experience. If this kind of game will start to spread, more and more players around the world will benefit from it. Playing video games is usually seen as a static activity just concerning the

brain function, instead now it can change. Combining the muscular activity of the whole body will lead to a more immersive reality. Being part of a fantasy, horror or alternative reality stimulates creativity and can be used as a learning tool. In fact the idea is to bring the game to the schools. Teachers have to follow the trend and be updated with the new technology. The game is the fastest way to capture the attention of children, and learning by doing helps to assimilate the concepts. So it is desirable that gamification can "inundate" us bringing the right motivations to go on.
The future games must keep in the first place the brain and the muscular health of each player, by concerning the human differences. The world is big and full of differences, but as human beings we are social animals. This belief results in putting more efforts in the social perspective also due to the more connected world. Sharing results among friends and compete must be the driving engine to the new serious games.

In conclusion, the game industry has to take care of its appearance to the world. The media always try to dig it down, but if the trend changes and new innovative serious games will be born they have to reconsider the viewpoint. Video games should not be linked anymore to the obesity problem or the massacres in America, but as a valve for venting and personal growth.

5.2 Future works

The game developed can be counted as a demo version for a future work. The features implemented are promising and can represent a solid pillar for the next effort. The GPS recognition works really finely and allows a well defined measurement. To improve the effectiveness of the interval training the introduction of a heart beat sensor will perfectly fit the cause. In this way the training can be really adapted to each single user. The customization can boost the final result and make the challenge more enjoyable because it respects the personal thresholds. It can affect the duration on the both stage, the rest and the workout and let the user train well respecting the right heart rate intervals. In this way it can be played more efficiently even by adults. An optimal device that can be used is the new kind of product called smart band. It is a generic name to indicate all the tech bracelets able to detect something. They can be used to count the sleeping hours, the time spent in doing sports, how many steps the user has tokens and someone even the heart beat. They are easy to wear and doesn't disturb the user and at the same time are pretty good in the detection.

Another aspect that must be cured in the next accomplishment is the social one. All the games now share information online and players like to brag and challenge each other. Being a part of a group can give the feeling of belonging in something bigger and all the efforts done can be registered in a superior plan. However, not only the competition moves players, but also cooperation. Playing with one common goal let people join and help each others, something really important for the grown of kids. For sure a new aspect must be designed, in fact the players are really affected by the graphic impact. More confetti effects can help to the engaging factor, because they provide powerful feedback. Colors, explosion, firework, falling candies or stars, these are the stuff that must be included for the polish. Regarding the game play, a suitable improvement is the petting. With petting are included all the actions toward the embellishment and the cure of the pet. Changing clothes, adding gadgets, washing,feeding, etc., are enjoyable activities that move the player's feeling. An inner connection between the animals and the owner will be established, as a tamagotchi, and the player will feel more responsible. A challenge for the cutest animal is just a logic consequence, but an added purpose of the game.
The last possible enhancement could interest the game play. Animals can offer different challenges such as different speeds, longer distance and mini game. The introduction of classes among animals can diversify the way how to catch them and it requires tactics to succeed. Strategy and logical thinking help to increase the brain skills useful in all the aspects of the life.

As Pokemon GO! tried but failed, the new frontier could be the augmented reality. A camera on the device could be easy to integrate this feature. In that case maybe the art style has to change into something more real because of the three dimensions. The animals have to become cuter and must have animations. To make them live is important and believable feedback. Jumping across the screen or moving around the room will totally change the way to play and to interact. Learning from applications such as Snapchat, gives the possibility to integrate a fictitious object as part of the real world and make people scream for happiness. In that way the pages of all the social media will be full of photos picturing the players with their most favorite creatures.

5.3 Skills acquired

Working on the development of a video game from scratch requires a lot of competences. Pragmatic and creative thinking for example are indispensable skills. The creative side leads to discovering a new field of application, new play mechanics and an esthetic sense in the realization. Those qualities have to be balanced to the practical sense because there are some limits to the technology that must be taken into account. When the two sides are well balanced it is necessary to make a plan with deadlines and priorities to be respected. Being able to assign those milestones is a management skill usually characteristics of the leaders, who decide how to organize the workflow. Another ability wished is the understanding of user needs and being willing in changing everything to meet them. Closely related to this aspect is the ability to test and retrieve valuable information from the analysis.

For this thesis the expectations were really high due to the serious aim. I have been immediately delighted in the pursuit of the fixed goal. My wished work is the work to help others to enjoy their life by the simplification of procedures or making educational games. In particular, i am a runner, and the prospection to bring this sport to the children was an idea too exciting.
In the time used for the development, I have improved the knowledge in Unity. I deepened the study of some aspects in the development of mobile games, spendable in the seeking for a job.
The environment settled for me was stimulating due to the presence of other students and the continued presence of the supervisor. The experience of communicating in another language has been difficult at first, but motivating. I have acquired the ability to show my work and point out my ideas. At the same time I also learn by doing, the agile methodology due to the continuous switch to the developing and testing phase.

Reference

Articles

[1] Suji Choi Hwan-hee Lee and Myeong-jin Lee1. "Step Detection Robust against the Dynamics of Smartphones". In: *Xue Wang,Academic Editor* (2015).

Web Pages

[2] Dr. Robert B. Bates. *Shaking the Body*. URL: http://www.funwithqigong.com/five-flows-qigong-set/first-flow-exercises/exercise-1-shaking-the-body/.

[3] Jacqueline Howard. *When kids get their first cell phones around the world*. URL: https://edition.cnn.com/2017/12/11/health/cell-phones-for-kids-parenting-without-borders-explainer-intl/index.html.

[4] Emma McDonald. *The Global Games Market*. URL: https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/.

[5] M.A. Melinda Smith and Lawrence Robinson. *Childhood Obesity and Weight Problems*. URL: https://www.helpguide.org/articles/diets/childhood-obesity-and-weight-problems.htm.

[6] Kevin Murphy. *Unity Game Engine Review*. URL: https://www.gamesparks.com/blog/unity-game-engine-review/.

[7] Kevin Murphy. *Unity Game Engine Review*. URL: https://cathe.com/supercharge-your-brain-with-hiit-training/.

[8] NCS. *HIIT Music Track – Level 3 – 30/30, 19mins – PLUS VOICE PROMPTS*. URL: https://www.youtube.com/watch?v=iHJbnIB7lRg&t=11s.

[9] Paul Trebilcox-Ruiz. *How to Recognize User Activity With Activity Recognition.* URL: `https://code.tutsplus.com/tutorials/how-to-recognize-user-activity-with-activity-recognition--cms-25851`.