Departamento de Ciências e Tecnologias de Informação

Physical Rehabilitation based on Kinect Serious Games-
ThG Therapy Game

Diogo Veiga Ferreira

Supervisor: Professor Octavian Adrian Postolache
ISCTE-IUL

Co-Supervisor(a):

Professor Raúl Oliveira
FMH-UL

Abstract

This thesis presents a serious game platform developed using Unity 3D game Engine and Kinect V2 sensor as a natural user interface. The aim of this work was to provide a tool for objective evaluation of patients' movements during physiotherapy sessions as well as a pleasant way that may increase patient engagement on training motor rehabilitation exercises.

The developed platform based on Kinect V2 sensor detects 3D motion of different body joints and provides data storage capability in a remote database. The platform for patient's data management during physiotherapy process includes biometric data, some data relevant for physiotherapist related to patient's clinical history, obtained scores during serious game based training and values of metrics such as the distance between feet during a game, left and right feet usage frequency and execution time for imposed movement associated with game mechanics. A description of technologies and techniques used for development of the platform and some results related to usability of the platform are presented in this thesis.

Keywords: objective physiotherapy evaluation, Kinect sensor, serious game, limbs rehabilitation, virtual reality.

Resumo

Esta tese apresenta uma plataforma de jogo séria desenvolvida usando o motor de jogo Unity 3D juntamente com o sensor Kinect V2 como uma interface natural de utilizador. O objetivo deste trabalho foi fornecer uma ferramenta para avaliação objetiva dos movimentos dos pacientes durante as sessões de fisioterapia, bem como uma maneira agradável que possa aumentar o envolvimento do paciente nos treinos de reabilitação motora.

A plataforma desenvolvida baseada no sensor Kinect V2 deteta o movimento 3D de diferentes articulações do corpo e fornece capacidade de armazenamento de dados em uma base de dados remota. A plataforma que gere só dados do paciente durante o processo de fisioterapia inclui dados biométricos, alguns dados relevantes para fisioterapeuta relacionados com o historial clínico do paciente, pontuações durante o treino e valores de métricas, como a distância entre os pés durante o jogo, o uso do pé esquerdo e direito, frequência e tempo de execução do movimento associado à mecânica do jogo. A tese apresenta a descrição das tecnologias e técnicas utilizadas para o desenvolvimento da plataforma, e alguns resultados relacionados com o uso da plataforma.

Acknowledgements

I would like to thank Professor Octavian Postolache for all the hard work in providing all the necessary conditions/materials and ideas for the realization of this thesis. Also, I like to thank to Professor Raul Oliveira for showing interest in the project and for providing feedback related to ideas presented.

Contents

Figures

Acronyms and abbreviations

DBMS - Database management system

SDK – Software Development Kit

VR – Virtual Reality

HIT - Health Information Technology

OS – Operating System

3D – three dimensions

eCAALYS - Enhanced Complete Ambient Assisted Living Experiment

IT - Information Technology

GPS – Global Portuguese Scientists

IR- InfraRed

RGB – Red blue green

UI – User Interface

2D – two dimensions

MS-SDK – Microsoft Software Development Kit

QR code – Quick Response code

RDBMS – Relational Database Management System

MySQL – My Structured Query Language

URL – Universal Resource Locator, the address of World Wide Web page

1 Introduction

Physiotherapy (in Europe) or physical therapy (terminology used outside Europe) has an important role in rehabilitation of different types of musculoskeletal or neuromuscular impairments (e.g. those resulted from accidents or stroke). The physiotherapists also are assuming leadership roles on health maintenance, wellness, and fitness of their patients.

To increase the effectiveness of physiotherapy sessions, physiotherapists tend to engage their patients by helping them knowing their physical condition but also providing them feedback on rehabilitation process by using measurements instruments for body functions (e.g. body segments movements). Electronic devices can store data and compare with historical data, for evaluation of the patient motion capabilities [1] or to predict future motor behavior.

The physical rehabilitation process for patients who suffered stroke events, is characterized by a long period of physical training and is associated with high costs for patients and society. The rehabilitation is strongly dependent to the frequency and effectiveness of the physical rehabilitation sessions, and in these conditions the remote physiotherapy by self-training at home represents a complementary and successful solution, that contributes for reduction of the rehabilitation period.

Training based on classical rehabilitation processes always require professional supervision and the use of equipment for therapy that doesn't provide any type of information regarding the rehabilitation process [2]. In the last decade the physiotherapists' community and the users of physiotherapy services have increased interest for novel solutions that apply information technologies in the physical rehabilitation field. Firstly proposed in 2002 [3] has nowadays many definitions, but as different authors refer to this as the usage of computer games for purpose other than pure entertainment, actually represent the common accepted definition for serious games.

Latest developments in the serious games field are expressed by the serious game for therapy (Theragames) that are reported in the literature, including Kinect Theragames refers to the usage of the Kinect sensor [4][5].

This type of game developments is linked to Kinect 3D sensor release in 2010 and also of the Software Development Kit (SDK) in 2011. A boom on serious game based on natural user interface by capturing human motion in a non-intrusive way took place, with many contributions in all kinds of areas. Kinect v2 that was release in 2015 is reported

having high accuracy in human body detection. The use of Kinect v2 as a natural user interface for physical rehabilitation scenarios, provide data related to patient joints position that can be analyzed in correlation with the applied rehabilitation game. The Kinect based games can provide training localization flexibility - the patients may train at home or at the physiotherapy institution - and may contribute to reduce the physiotherapists' efforts, and to extend the capability to follow their patients everywhere by recurring to mobile applications connected to Kinect serious game frameworks [6].
Other benefit of this technology is the capability to increase the user motivation during the rehabilitation process based on more innovative and enjoyable trainings [7].
In this work, the Unity 3D game engine was used to develop different virtual reality scenarios adapted to different types of patients. The interaction with the implemented VR environment is based on the use of the Kinect v2 motion sensing sensor that allows to track the patients' motion, and to use the patient detected motion as game action.
The virtual reality scenarios presented in the literature provides the possibility to the user under rehabilitation to make exercises that work by imitation of the action performed by an avatar, or to play a game that will train specific parts of the body depending on the type of rehabilitation. The data recording functionality implemented on framework level and data analysis of the stored data, the evolution of the motor capabilities of the patients can be easily analyzed by physiotherapists.

1.1 Motivation

Nowadays technology is applied to almost everything to improve or facilitate people's lives. One great impact that it has been showing throughout the years, it's in the medicine [8].
In motor rehabilitation is a great need for improvement and application of new technologies, so the patients can have a better and more documented progress in their sessions. The technology may bring an easier and careful way for the resolution of their problems as well as for the monitoring and attention needed by physiotherapists [9].
Physical rehabilitation is often related to stroke events. Many people are unsatisfied with the level of rehabilitation provided in the years following the stroke. Exercises are often tedious or unpleasant leading to patient's tolerance towards the exercises to decline, with consequences in their process of recovery [10].

The use of Kinect not only provides adherence to treatment by adding fun and entertainment processes to rehabilitation processes but also reduces the high cost associated with traditional rehabilitation making it more accessible and affordable [11].

It is expected that with this work and research of new technologies for the physical therapy area, in this particularly case the development of serious games, will have a positive impact on the life of people that need motor rehabilitation of superior and inferior limbs, but also to provide new and better tools for the physiotherapists, so they can give services for a better and faster evolution of their patient's well-being.

2 Related Works

Health information technology (HIT) it´s an application where the information is retrieved by using technology that can help and improve the patient's treatments (e.g., providing faster rehabilitation). HIT include data storage, data retrieval, data sharing and usage of health information for the best decision making [12].

An information system was designed in this work based on mobile technology – software application for smartphone and tablet for physiotherapy process management, and serious games with natural user interface based on Kinect sensor, for physiotherapy process improvement.

2.1 Serious Games

Through the years video games have shown to be of more use than just for fun and entertainment. Studies have shown that serious games can be applied in different areas such as the military, education, and healthcare systems. Their main advantages that are related to intuitive interface, affordability, environments that cover a wide range of both educational and training applications, capacity to provide faster results on players and also instant gratification and rewards, are very important on creating a feeling of satisfaction on players [13].

In the motor rehabilitation there is a great need for improvement and application of new technologies, so the patients can have a better and more documented progress in their sessions. This will bring an easier and careful way for the resolution of their problems as well as for the monitoring of patients by physiotherapists [14].

In [15] authors test a set of commercial games with a group of therapists to identify the best game design for rehabilitation purposes. They reach the conclusion that meaningful play, where the patients actions reflect on the game results led to increase interest and motivation of patients. By challenging the patient through different sessions, and displaying instant results the patient will notice a skill increase, that lead him to take a higher challenge in the game to continue enjoy playing.

Authors in [16] discuss the impact these games might have in patients by categorizing the benefits by three criteria. The first criteria is related to effectivity and efficiency, the second to social impact, physiological, psychological, and sensory motor factors, and the third is the quality of the results produced. By having these three criteria, they concluded that games can bring substantial benefits to a set of patients, however they suggest researchers to adjust and evaluate the different settings to accommodate every patient with different characteristics, as the ones defined on the second criteria in order to achieve sustainability.

Learning from games can be viewed by two types: motor or affective learning. In [17] we can identify some features that are optimal when creating a serious game play environment. Such as:

- Different experiences and practices that keep challenging the learner and reinforce expertise
- Continuous monitoring of progress, and use the retrieved information to diagnose performance and keep adjusting patients level of mastery
- Clear goals

In [18] have shown that the use of Kinect not only provides adherence to treatment by adding fun and entertainment processes to rehabilitation processes but also reduces the high cost associated with traditional rehabilitation making it more accessible and affordable.

2.2 Wii sensing

Through the advance of technology, interactive computer gaming technology has gained more attention due to the components it provides to help in the rehabilitation process. Monitoring technologies have proven to be successful in bringing a different environment regarding the rehabilitation sessions, relieving the stress, and bringing encouragement around the patients.

The use of gaming technology such as Nintendo Wii, allows therapists to do more targeted treatments by taking advantage of the motion sensing controllers. Also, known as Wii-Hab, this approach helps patients to confront and engage their problems in a more active and enjoying way, making them more inclined to continue to do it. [19]

In [20] the author implements 7 stroke rehabilitation games using both webcams and Wii remotes. They attached the Wii remotes to patients' arms to make use of the accelerometers of the Wii remotes for detection of upper limb motions.

As mentioned in [21], the Wii console and its accessories have been a very practical and helpful tool for physiotherapists, having a wide variety of games for a different set of patients. Although Wii has a great games variety, there are some limitations they bring. Authors mention that not every game is ready to accommodate a patients need, thus they experienced some lack of precision in monitoring while playing certain games, because they were mainly developed to be used by healthy users and do not take in consideration the possibility of a user with some kind of disability. The lack of metrics is also another problem described, which were proven to be insufficient to continuous patient's evolution monitoring.

Although some Wii games give positive reports to patients and their evolution through sessions, there is a problem that still needs to be addressed, that Wii gaming technology can't offer, which is game interaction without the need of game controllers or any type of body accessory. As mentioned in [22] the patients require a controller provided with accelerometers that are sensitive to changes of direction and speed, which allow the communication/interaction with the game.

2.3 Kinect

2.3.1 Kinect tests

Kinect device was firstly released onto the market in 2011, bringing important contributions to all kinds of areas. The device may have great impact in the physiotherapy area, allowing to capture human motion in a non-intrusive way, to adjust physiotherapy intervention to each patient need and to give them more freedom of movement during sessions.

According to [23] after some feedback and data collected from patients it was proven that this natural user interface brought an increase in motivation during rehabilitation sessions, by engaging the patients with more innovative and enjoyable sessions.

As mentioned in [24], after testing different Kinect-based applications with a different set of patients in different rehabilitation conditions and situations, the Kinect sensor presented a detection rate of more than 80%, not reaching 100% due to auxiliary accessories used by the patients, like wheelchairs and crutches.

Nowadays there are other kind of technologies that can quantify motion. Some of them are Leap motion and Soft Kinect.

Soft Kinect provided with gesture recognition capacity, can be applied to different areas like consumer electronics, health and fitness and serious game industries. Equipped with two types of sensors, one for near range detection, directed to hand and face tracking, between a range of 0,15 to 1 meter and other for near-far range directed to full body tracking between ranges of 1.5 and 4 meters. The drawback of this equipment comparing, is the cost, not being as affordable as the Kinect v2 sensor. [25]

Developed by a startup called Leap Motion, was designed to be included into laptops and mobile devices. This tracking sensor achieved a high popularity nowadays, mainly because of its size and accurate tracking. As Kinect and Soft Kinect, it allows the user to wireless interact with a computer by moving their segments of body in a 3D space. However, it only works for small ranges (2,54 cm to 50,4 cm) and is only able to track hands and finger joints. Regarding that, it possesses a field of view of 150 degrees. Having a higher accuracy than Kinect sensor, this sensor is more suited to use at desk range, displaying lots of possibilities like handwriting recognition, virtual keyboards and more precise hand gesture controls. [26]

2.4 Smartphones

Nowadays smartphones are increasingly viewed as handheld computers, due to their powerful capability of computing, capacious memories, large screens, and open operating systems that encourage the develop of different mobile applications related to different work areas, one of them being the healthcare. [27]

2.4.1 Android Mobile Applications

Android OS platform is one of three most used operating systems in mobile phones, being one of the most used, because of its virtue of being an open source system. The use of android applications to read and collect data from IT applications has been proven to be a reliable and practical tool. It allows access of data from everywhere. This may permit to the physiotherapists to have a closer following in patient's sessions and evolution by maintaining a constant "eye" through the data collected and display on the mobile app.

The eCAALYS Mobile application is being developed as part of a project named (Enhanced Complete Ambient Assisted Living Experiment) which aims at developing a remote monitoring system for older people with chronic diseases. The main functionality is to act as intermediary between wearable health sensors, used by the older patients to provide information regarding measurements obtained and the geographic location of the user via smartphone GPS. Additionally, the mobile applications were built for detection of anomalies in patients' health such as tachycardia and signs of respiratory infections, based on established medical knowledge [28].

3 Serious game for Physical Rehabilitation

A "serious game" also called "applied game" is a game designed for more than pure entertaining. Serving other purposes like education, scientific exploration, engineering, health care, among others. Nowadays serious games have more than one meaning. Appliances, sensors like Kinect to detect body motion without the use of wearable sensors. Connected to healthcare "Theragames" were created to address therapy games. With these advances, new approaches came up improving and enhancing the classical rehabilitation processes and their approach to patients.

3.1 Kinect sensor

Kinect known as a motion-sensing device, was developed for Microsoft Xbox 360 gaming console. A feature that stands among others, that allows the replacement of the traditional game controller, is the possibility to play the game by the use of the user's body motion, position and voice that allow interaction with the game.

The human joints are captured in unobtrusive way during the physical rehabilitation exercises using Microsoft Kinect sensor that allows the acquisition of IR (Infrared), RGB and depth image with high frame rate (30fps). Today there are currently two versions of the Kinect sensor (V1 and V2). The Infrared emitter and an Infrared depth camera (640x480 pixel for Kinect V1 and 512*424 for K.V2) allow extraction of depth information. The IR laser emitter delivers infrared light beams and the depth sensor reads back the IR beams reflected back to the sensor. An RGB camera that stores three channel data in 1280x1024 pixels resolution for Kinect V1 and 1920x1080 for K. v2.

The most recent version of the Kinect was used for this project and its sensors are represented in Figure 1.

Figure 1. Kinect V2 Sensor characteristics

The values of depth distance change from V1 to V2 while V1 as a maximum and minimum distance of 4 and 0.8 meters, the V2 as 4.5 meters of maximum and 0.5 of minimum depth distance (Figure 2). Kinect V2 sensors have a wider angle of view comparing to Kinect V1 making possible to come closer to the sensor and thus decreasing overall space requirements [29][30].

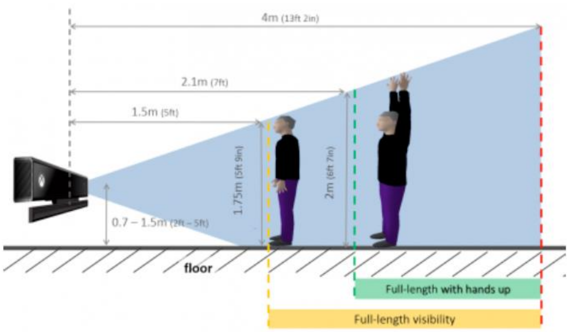

Figure 3. Kinect V2 Field View Representation

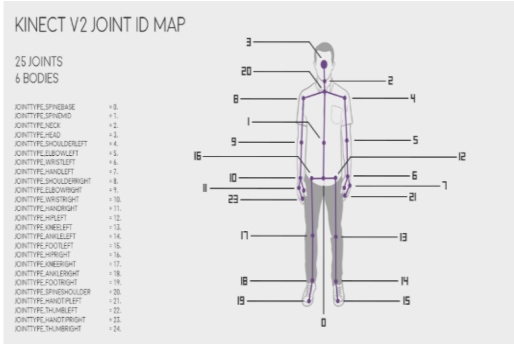

Figure 2. Kinect V2 joint map

The number of joints that can be tracked is also bigger, Kinect V2 allows 25 skeleton joints (Figure 3) and Kinect V1 only 20, thus contributing to a better gesture recognition [31].

3.2 Game Engine

A game engine consists in a cross-platform game engine, mainly used to develop video games and simulations either for computers, consoles or mobile devices. There are a wide variety of game engines able to support 2D and 3D makes graphics, each one provided with different features. Some examples are, Unreal Engine 4, Unity3D and CryEngine. We opted to choose the Unity3d game engine, because of his intuitive work interface, a complete set of tools and a rapid workflow. It is also able to support multiple languages like C# and JavaScript what makes the developers work easier given the semantic similarity between the two. As mentioned before it as a very intuitive editing interface with all kinds of tools, and very powerful documentation that supports and helps the developers to have a better integration with the engine. Also, it is provided with third party solutions for Audio and Physics systems.

Figure 5.Unity Logo

In the asset store, there are many types of assets like objects, scripts, textures, UIs and more that can be accessed and integrated into the project, some free, some paid. Also, there is a very good community where developers can expose, and answer questions related to the projects they are developing. It also makes publishing into different platforms like mobile, PC, PS4 easy. Those were the leading factors that made us choose Unity3D [32].

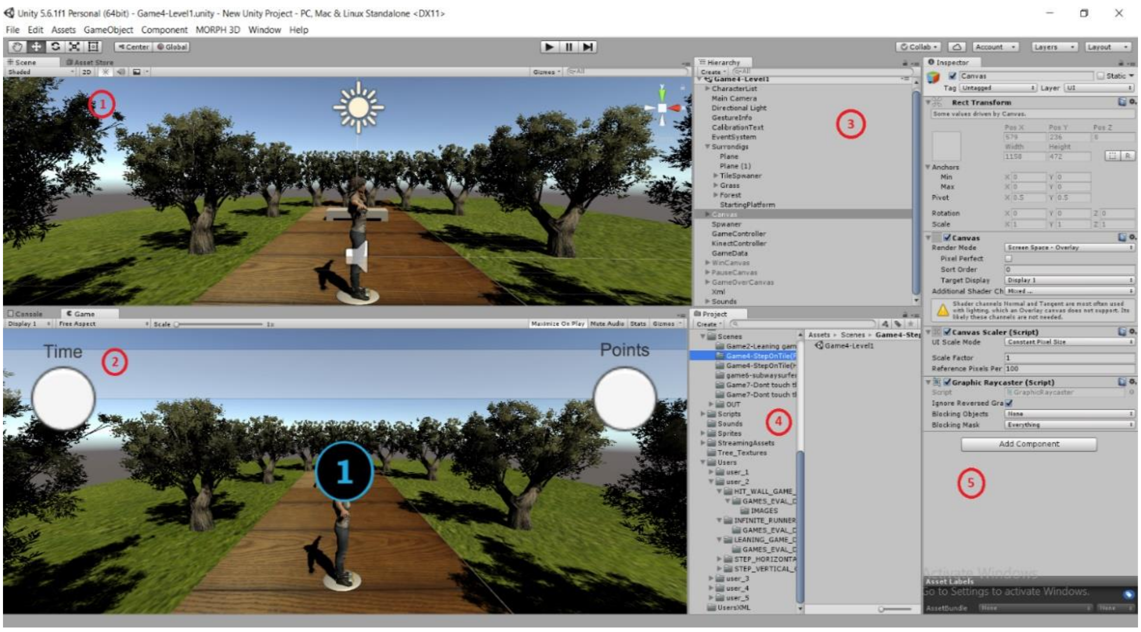

Figure 6.Unity work interface

The user game interface of Unity 3D can be presented in various ways, being most of the tabs represented, movable. In Figure 5 is represented a possible one. Unity Dinterface is composed by various working tabs. Represented with number one is the scene tab where the developer can deal with all sort of objects, by drag and drop them into the scene, scaling, rotating, changing textures and format (1). It also as an area where it's possible to see the changes made in real time preview represent by the tab "Game" (2). Next to the game tab there is a Console tab where the user developer can see the errors detected in the game code and print Debub messages. A hierarchy tab is also provided, where all the scene elements are represented and can be dealt with (3).

When selecting a game element in the hierarchy or directly in the game scene, the details of that element will be presented in the Inspector tab (5), which provides information of what is attached to that object. It can vary from scripts that add certain functions, to a rigid body element that gives the object attached physic properties.

The user can also see and manage its project in the project tab (4) that gives the user a tree vision of the resources, scenes, sounds, scripts and other elements needed for the game development.

3.3 Tests

When starting with Unity 3D and the Microsoft Kinect sensor, the first thing to be done was to understand how to connect and interact with the Kinect sensor within the game engine. We made use of an asset available in the Unity 3D asset store named "Kinect v2 Examples with MS-SDK" [33] that facilitated the learning and integration between Kinect and Unity3D.

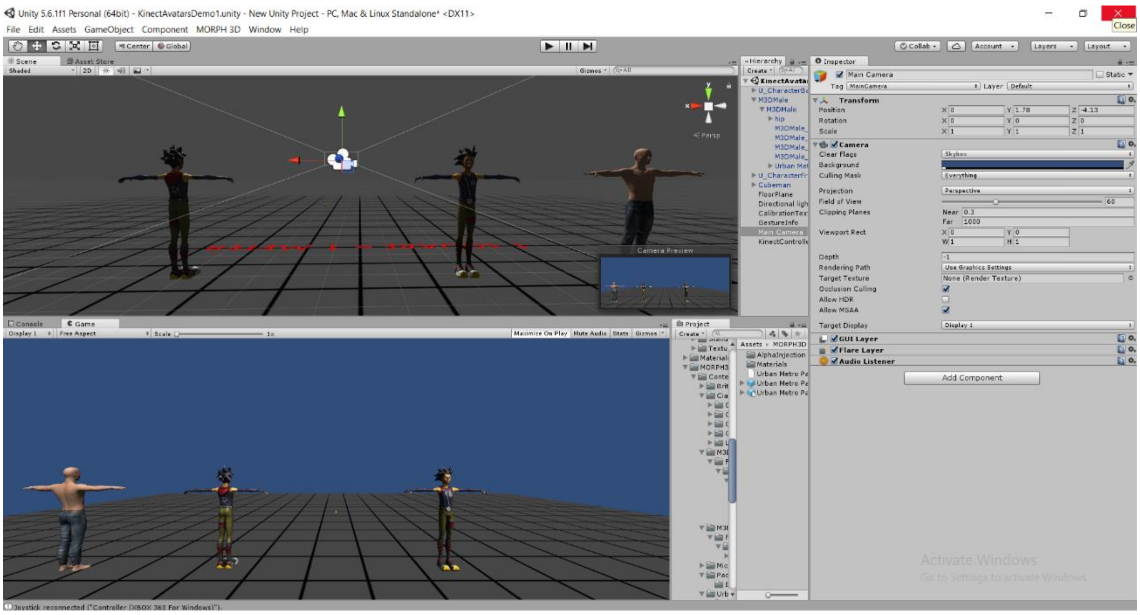

Figure 7.Kinect V2 avatar test control

To really understand how the Kinect and unity function together and how it get to track the human body, some examples provided by the asset acquired before were run and some avatars were added to see how accurate and reliable the user body movement was compared to the avatar controlled.

After running other examples regarding other Kinect aspects like the use of gestures to control objects, some gaming tests tutorials were made to familiarize with the unity interface and its components. These tutorials are in the unity web site under tutorials [33]

3.4 The Game Environment

3.4.1 First Scene – Initializer

After some tests made and some problems found regarding the Kinect manager, the scene was created only containing an empty Game Object named "KinectController" with a script attached named "Kinect Manager" that serves to initialize the game and turn the Kinect sensor On. This is the only game object of the game that as the Kinect Manager script attached.

3.4.2 Second Scene – Menu (UI)

This second scene is automatically opened when the first scene is Initialized, presenting the user with the *Login* panel as shown in the Figure 7.
The *Login* could be done by three types of users: *Patient*, *Physiotherapist*, and *Admin*. The *Patient Login* gives the user access to the games and his stats. *Physiotherapist Login* gives access to patient's stats and games. The Admin can add new physiotherapists. Their permissions and functions within the game, are explained in figures 8 and 9.
Patient register is possible through the *Login* panel and should be done in presence of a physiotherapist. Physiotherapist register is only allowed upon Admin *Login* [Appendix B – figure 75]

Figure 8. Game application login menu

After the information required is provided in the register panel (figure 11) a QR code is generated for each user and saved on the local host of this game, which can be used to login instead of typing the username and password credentials.

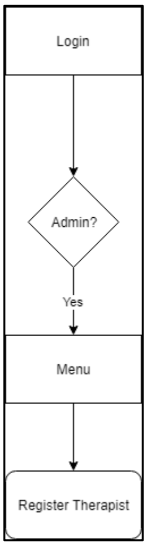

Figure 9.Admin permissions flow chart

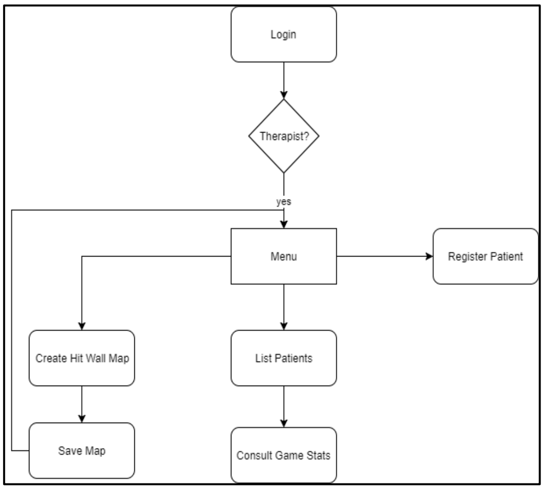

Figure 11. Therapist permissions flow chart

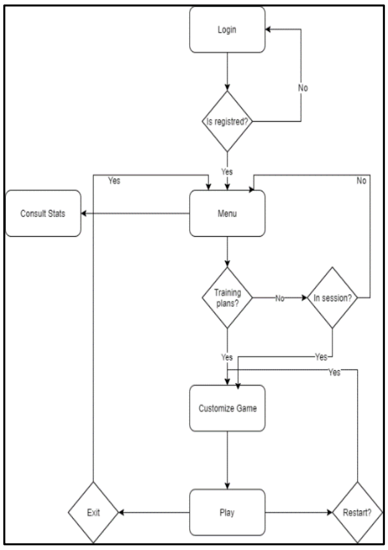

Figure 10. Patient permissions flow chart

Another set of panels also composes the Menu scene [Appendix B] regarding the UI where the user can navigate and play the different games within. Before the scene games, all games must be customizable in the respective interface. Both the customization and game play will be addressed in the next section.

3.4.3 Gaming Scenes

Figure 12.Game application register interface

Each game is represent by a different scene, a different customization panel where input need to be provided by a physiotherapist or by a companion, provided with a therapist prescription/plan, in order to configure the game. After every game is played, data is collected both for the patients and therapist observation. There are two types of data the gameplay provides: Front data and back data both will be discussed in the sections following.

Before the *Play* button is pressed it is imperative that the patient must be in Kinect field of view, to be detected and assume control of the player.

Data

Related angle data, a first approach was trying to obtain the angle between joints by using the dot product properties formula, represented below, but after some research, simplified unity was realized by creating a function that calculates the angle between two three dimensional vectors represent by the bullet point sequence below.

$$\overrightarrow{AB}.\overrightarrow{BC} = \|\overrightarrow{AB}\|\|\overrightarrow{BC}\|\cos\theta \Leftrightarrow \theta = \cos^{-1}\left(\frac{\overrightarrow{AB}.\overrightarrow{BC}}{\|\overrightarrow{AB}\|\|\overrightarrow{BC}\|}\right)$$

- Vector3 1 = A – B
- Vector3 2 = B - C
- Angle = Vector3.Angle(Vector 3(1), Vector 3(2)

3.4.3.1 Leaning Game

This scene regarding the Leaning game, is where the player will train their lower limbs by applying some pressure on the leg and leaning to the respective side he wants the player to move, to avoid /catch the different set of tokens.

Before playing, as mentioned before, the game needs to be customized, providing input to configure the different set of options of the game - as the ones represented on the figure 12 where (1) represents game information that may be helpful for the user and (2) the different game variables to configure the game.

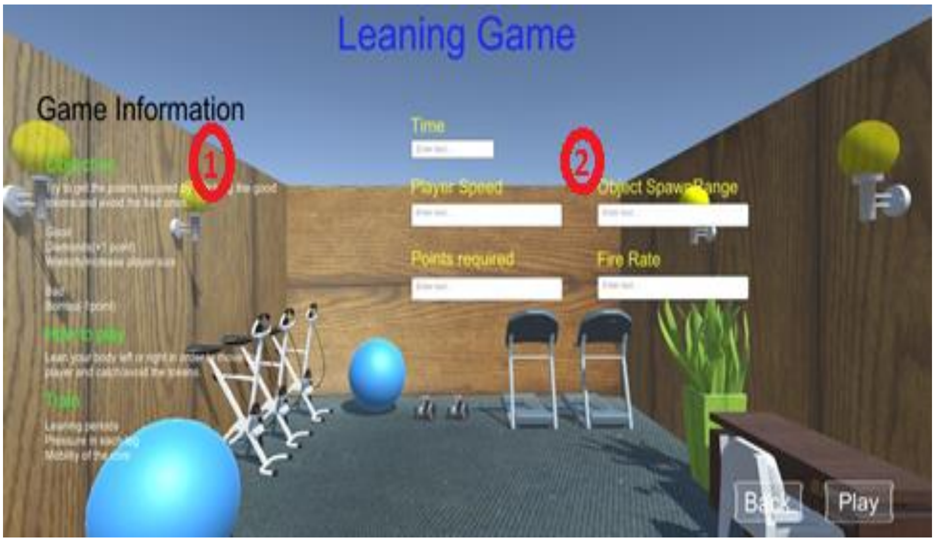

Figure 13. Leaning game Customization interface

The player of this game is represented by a “stretched cube” (3)(parallelepiped) where the patient takes control of it by leaning to the side he wants the player to move. With this, the main objective is to collect objects that give points and avoid the ones that take, to reach the established score upon customization. There are other types of objects within the game that impose challenge or help the patient during gameplay. By adding different objects, the physiotherapist may understand some cognitive capacities of the patient, observing which ones he prefers to avoid and to catch.

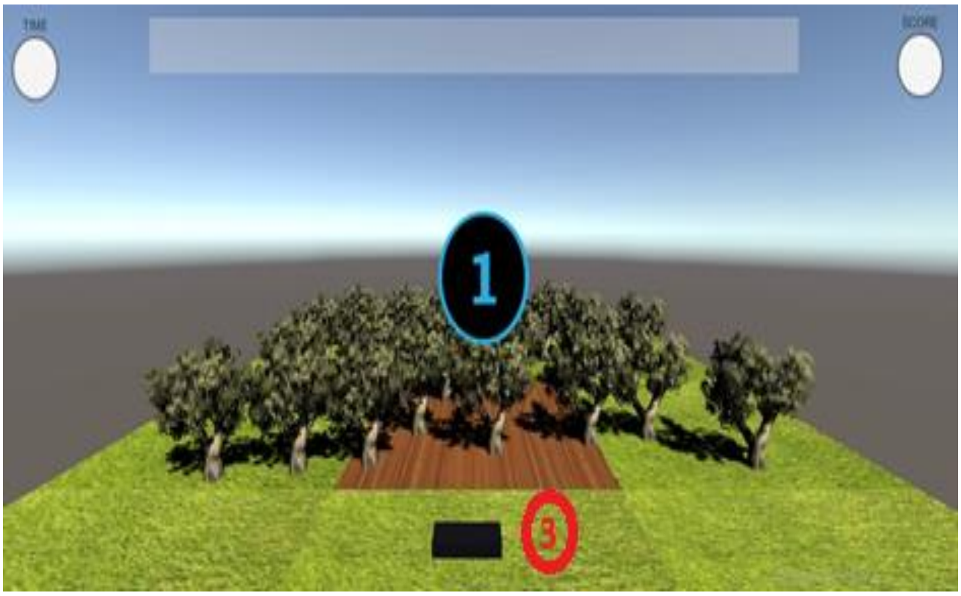

Figure 14. Leaning game Play scene

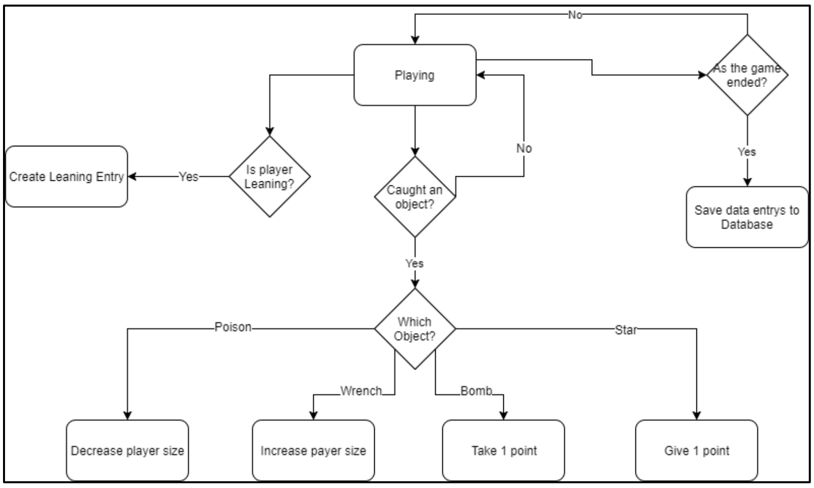

Figure 15. Leaning game flow chart

3.4.3.1.1 Data collection

As mentioned before there are two to types of data. The front data that consists in data that can be seen by all three types of users, displayed for the patient and companions. These are the final score, input fields provided upon customization, number and name of the objects caught and win or loss of the game. These can be display on the game in stats panel or in the mobile application. The second type of data is directed only for the physiotherapist analysis and consulting. It is gathered during game, while the patient controls the player.

In this case, a script named “gesture listener” (Appendix – C) is constantly “listening” to the leaning gesture (Figure 15), every time a leaning is detected the time period of the leaning movement is saved as well as the leaning side and angle progression during that period of time.

```
if ((gesture == KinectGestures.Gestures.LeanLeft && progress > 0.5f)) {

    leanLeft = true;
    if (gestureInfo != null) {
        //Screen Debug///////////////////////////////
        string sGestureText = string.Format ("{0} : {1:F0}", gesture, screenPos.z);
        gestureInfo.text = sGestureText;
        ///////////////////////////////
        progressGestureTime = Time.realtimeSinceStartup;

        LeaningLeftList.Add (sGestureText + " ," + progressGestureTime);

        progressDisplayed = true;
```

Figure 16. Code part representing Leaning Left Gesture

The angles are calculated recurring to two vectors depending on the leaning side. One vector common to both sides is the z axis vector that remains static and other that goes from the hip to the shoulder as shown in Figure 16.

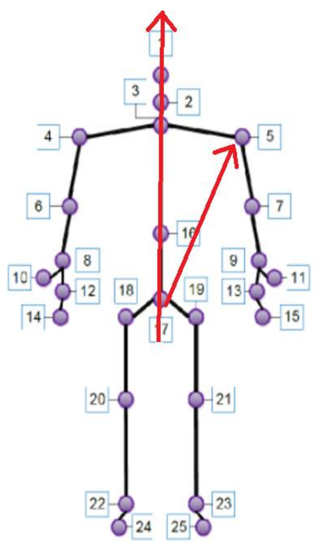

Figure 17 .Representation of vectors through where lean angles are calculated

3.4.3.2 Step the Tile

This game was made with the aim of providing a tool for working patient's reaction and flexibility on the lower limbs. The game is customizable to the point of challenging the patient even more, by providing different game inputs if the game does not pose challenge to the patient.

Before playing the game, the physiotherapist must evaluate the patients' movements and start with easy game inputs. While customizing the game the physiotherapist is given two options, if the patient is to perform side steps or front and back ones by selecting "Horizontal" or "Vertical" [Appendix B- Figure 83] .

On *Start* the patient will have control of the avatar placed in game. The avatar must be on the platform like represented in Figure 17 (1). Two types of tiles will be spawned, the green ones (2) that gives one point and the red ones that take points if stepped.

Figure 18. Step the tile play interface

3.4.3.2.1 Data Collection

Regarding the data collected in this game, the inputs provided before gameplay and number of steps taken with each foot are also stored at the end of each game as well as the number of tiles stepped.

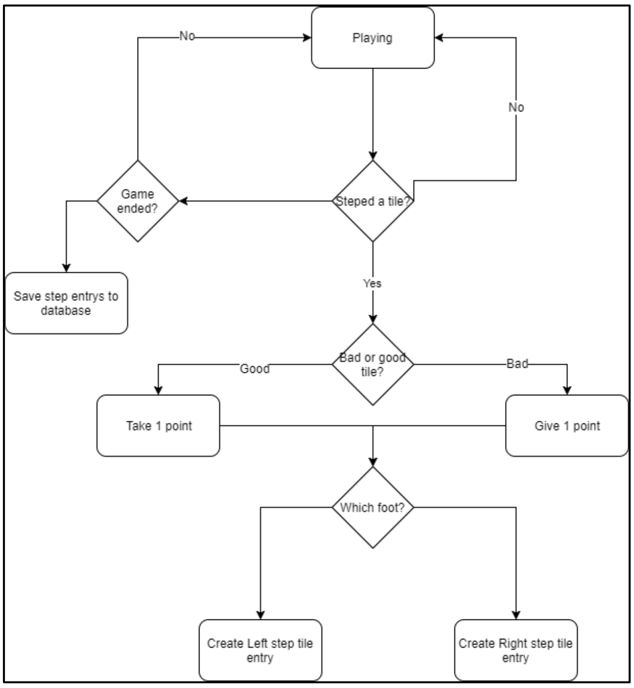

Figure 19. Step the tile flow chart

Movement can be understood by starting on the initial position with both feet inside the platform (1) doing a step and come back to the original position.
For the physiotherapist analysis the angle between legs is calculated after stepping a tile. Step distance is also calculated every time each tile is stepped, attached to movement execution time and movement velocity.

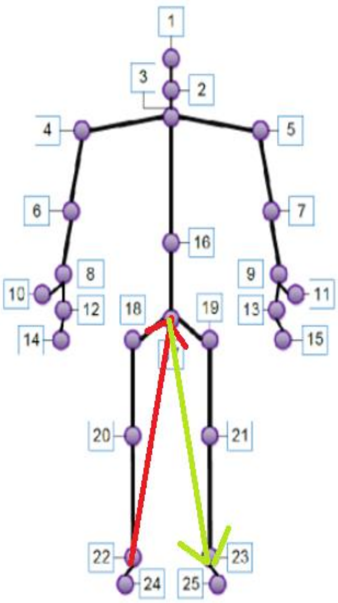

Figure 20.Vectors used the leg angles are calculated

The formulas used to calculate equation referred before are represented as follows.

Angle scripting calculations:

```
public float calculateAngle(){
    Vector3 leftFootPel = LFoot.position - Pelvis.position;
    Vector3 rightFootPel = RFoot.position - Pelvis.position;

    angle = Vector3.Angle (leftFootPel, rightFootPel);

    return angle;
}
```

Figure 21. Code part representing angle calculation

Distance:

```
public float calculateDistance(){
    distance = Vector3.Distance (RFoot.position, LFoot.position);
    return distance;
}
```

Figure 22. Code part representing distance calculation

Time taken to performed the movement is calculate by starting a time variable when the movement starts (when the avatar starts moving out of the platform) and finishes the movement by getting back to position.

The movement velocity is calculated by first multiplying the step distance by two in order to have the full movement distance, and divide it by time of execution as can be seen in figure 22.

Velocity calculation:

```
public float MediumVelocity(float distance, float movTime){
    float mediumvelocity = (distance*2) / movTime;
    return mediumvelocity;

}
```

Figure 23.Code part representing velocity calculation

3.4.3.3 Infinite Runner

Referring to the upper body members, the game was manly made for the patient to gain flexibility and dexterity on the upper members by collecting the objects in the scene with his hands.

On start, the avatar controlled by the patient will move forward in the map at a certain speed defined by the therapist. The patient will have completely control of the avatar from the waist up. While moving forward the patient will have to avoid the bombs (1) objects placed on the game ground, meant to take points, and try to hit the medic Packs (2) in order to gain lives (3), to do that he will have to control the avatar by leaning to the desired side and stand straight to make the avatar move straight.

The game objective is also to reach a certain score stipulated by the physiotherapist, by gaining points, hitting the objects placed on hands reach. Score increases by hitting a diamond object (4) (gains points) that are placed on both avatar sides and top when the game starts. The player also must be aware of poison objects (5) that take lives and spawns on the same areas as the diamonds.

With this the physiotherapist can have a better notion how the cognitive part of the patient is responding according to the objects he catches during the game.

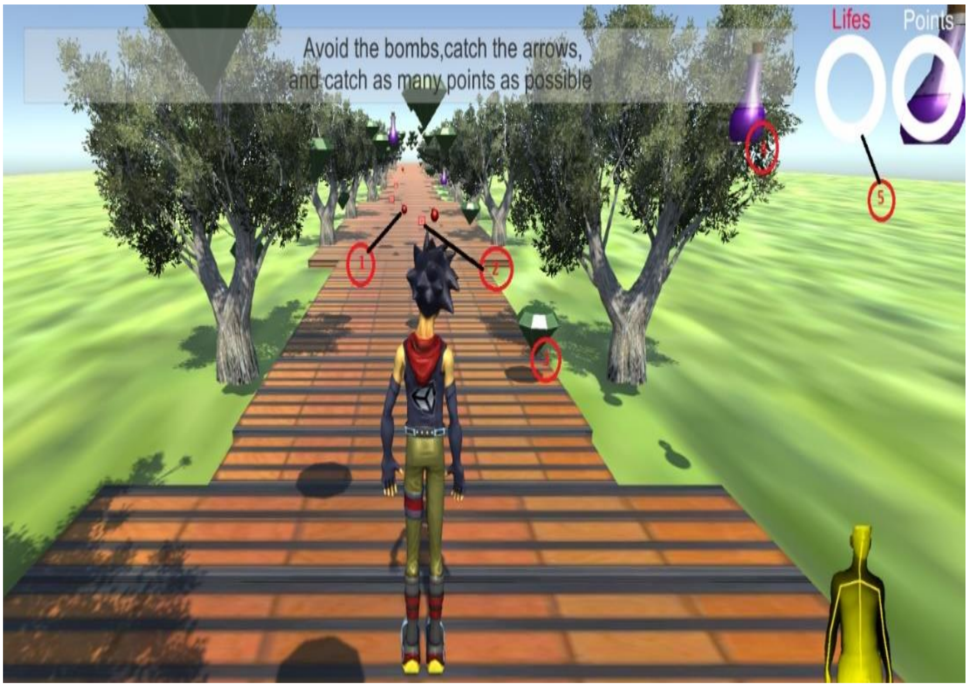

Figure 24. Infinite Runner play interface

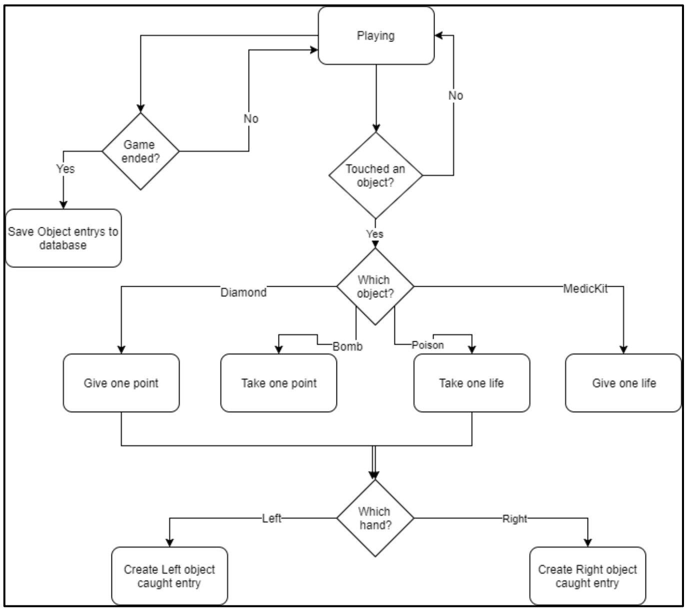

Figure 25. Infinite Runner flow chart

3.4.3.3.1 Data collection

The game has the same type of data for patient view as the leaning game, counting the number of objects caught.

One vector is parting from the hip joint [18,19] to shoulder joint [4,5] and the other from the shoulder joint [4,5] to the hand joint [14,15] respectively (Figure 25).

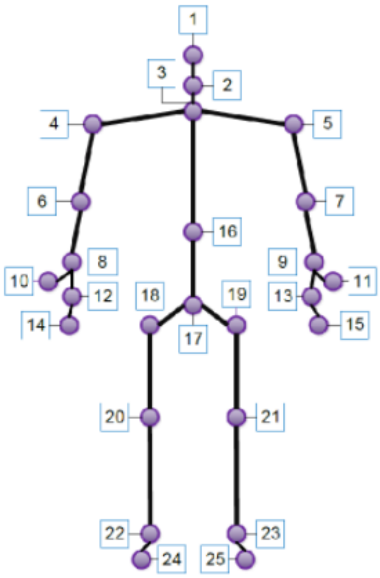

Figure 26.Person Kinect v2 joint display

Moreover, as the patient collects an object with his hand, the angle when the object caught is calculated. The scripting game calculation can be seen in Figures 26 and 27, making use of two vectors depending on which side the object is caught.

```
void calculateLeftVectors(){
    float angleL;
    Vector3 hipToShoulder = Lhip.position - LShoulder.position;
    Vector3 handToShoulder = LHand.position - LShoulder.position;

    angleL = Vector3.Angle (hipToShoulder, handToShoulder);
```

Figure 27. Code part representing left arm angle calculation

```
void calculateRightVectors(){
    float angleR;
    Vector3 hipToShoulder = Rhip.position - RShoulder.position;
    Vector3 shoulderToHand = RHand.position - RShoulder.position;


    angleR = Vector3.Angle (hipToShoulder, shoulderToHand);
```

Figure 28. Code part representing right arm angle calculation

3.4.3.4 Hit Wall

This game focused also on the upper limbs. It, having a set of different game scenes which correspond to four different levels for each hand, levels which were created manually. A different approach was taken where it only exists one scene, and the different maps/levels are loaded on game start from a *xml* file.

In this game we wanted to turn the game even more flexible, than just have premade levels, so in customization game window, before starting a new game, there is a possibility for the therapist to create his own map, leading to an editing window represented by Figure 28, where is possible to instantiate wall game objects to define the path. Player and finish point game objects must be placed on the desired positions to have a better space perspective. When the game map is saved, a screenshot of the map is taken, so when the map created is selected in the customization menu the map is displayed so the therapist can know which one is selecting.

With this, the therapist can try a wider variety of maps with different formats and definitions in order to test the patient and challenge him as he advances in his sessions.
During gameplay the patient should try and simulate grabbing (1) that is represented on the canvas by controlling the pointer hand format that is connected to the patient's hand tracked joint, and reach the finish game object (2) without touching the map walls. A life is lost each time the player touches the wall.
For better performance within the game, it's better for the patient to have only the hand is using in Kinect field of view to avoid conflict detections of the opposite hand while controlling the object.

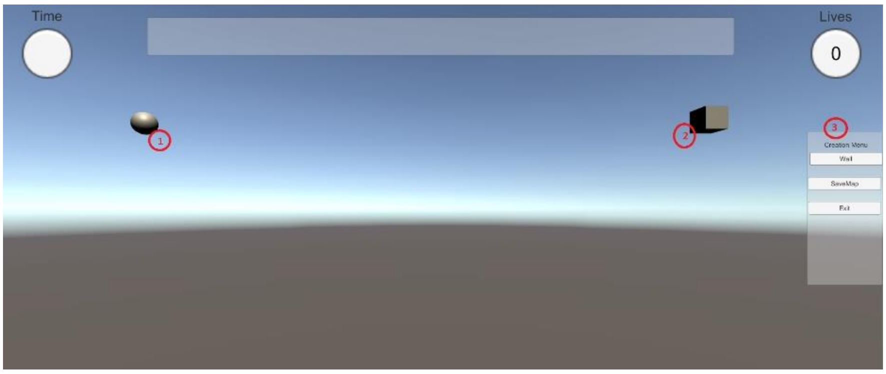

Figure 29. Create hit map interface

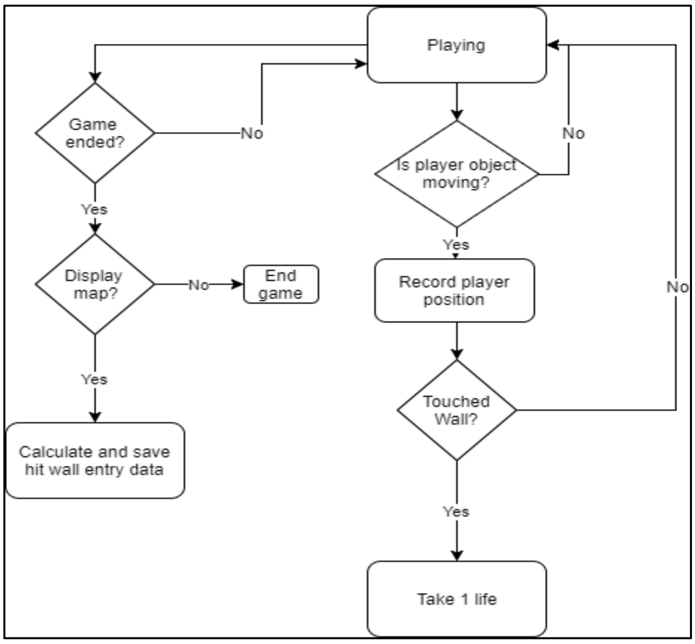

Figure 30. Hit game flow chart

3.4.3.4.1 Data collection

In this game, the collected data the patient might consult is the time it took to complete the course and the input fields on the custom menu before play.

After the end of each game, an option appears for the physiotherapists that is named “Display map”. By clicking it, a set of dots tracing the way the patient took to complete the course are set on the screen, each dot represents each player position every frame during gameplay as seen in Figure 30.

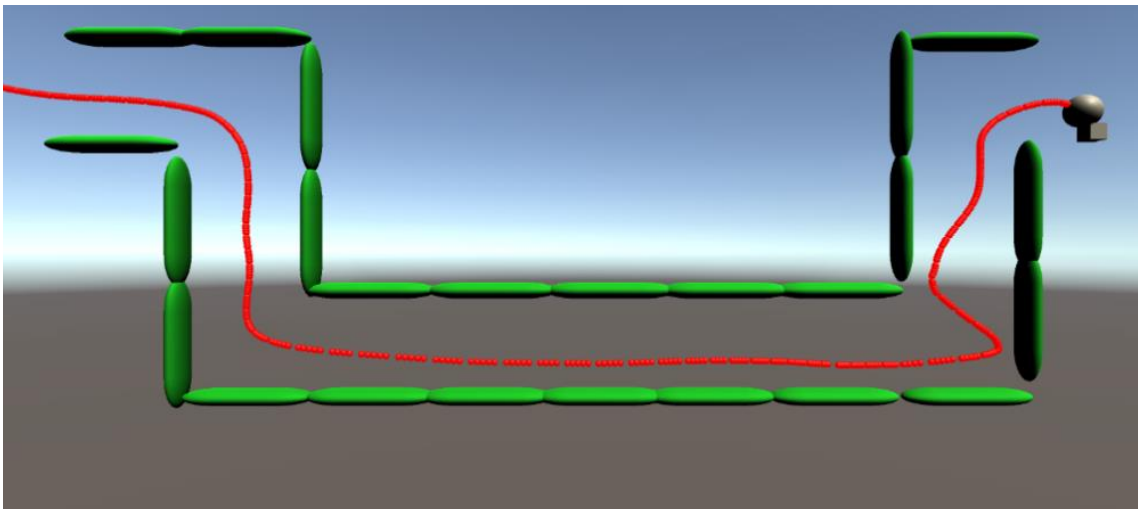

Figure 31.Map path display

By having that, the path distance was calculated by measuring each dot to dot distance, represented in Figure 33, and sum it up. The medium velocity during the course was calculated by dividing the distance to the time taken from one dot to another.

```
void FixedUpdate(){

    playerPositions.Add (playerPos.position);
    playerPositionsTimeStamps.Add (gameTime);
}
```

Figure 32. Code part regarding player positions

```
public void DisplayMap(){

    foreach (Vector3 t in playerPositions) {
        Instantiate (dot, t, Quaternion.identity);
    }
}
```

Figure 33. Code part for path display

```
for (int i = 0; i < (pos.Count -1); i++) {

    previousDistance = Vector3.Distance (pos [i], pos [i + 1]);
    distance = distance + previousDistance;

}
```

Figure 34. Code part for path distance calculation

3.4.4 Game Feedback

For keeping the player motivated and interested during training based on serious game, feedback actions were considered.

```
for (int i = 0; i < (pos.Count - 1); i++) {


    previousDistance = Vector3.Distance (playerPositions [i], playerPositions [i + 1]);
    float timeInterval = playerPositionsTimeStamps [i] - playerPositionsTimeStamps [i + 1];

    float mVel = (previousDistance)/timeInterval;
```

Figure 35. Code part for player velocity

Thus, different elements were added, like audio and text message feedback. For each game the audio feedback varies, but the idea was to play comfortable sounds whenever, the player catches an object that gives them points, when enhances the game player or when a game is won/lost. Also, feedback is provided when the player interacts with objects that take lives or points like in the case when the user through his avatar steps a red tile in the "Step Tile" game. The message feedback, works in the same way as the sounds. Motivating messages pop up whenever the user interacts with a good object.

Figure 38 on Appendix-B is represented an example of message that might appear in the message panel.

3.5 Conclusions

The serious games may have an important role for future rehabilitation sessions. Increasing patient's motivation and engagement in their training sessions. To be able to achieve future expectations this game must be flexible to accommodate all kinds of users and collect all type of relevant data given the patient's condition. This data will be helpful for therapists to analyze and to give closer attention to patient's progression. A sensing device like Kinect combined with powerful game engine tool Unity 3d give more flexibility for acquiring patient motion data.

4 Database

A database is a group of accumulated data that is typically organized and is understood as a way of modeling aspects of reality aiming to support processes requiring information. When a system that deals with large amounts of data is developed a well-defined and structured database is essential for a good functioning of the system.

4.1 Database Management

For this project we used a Database Management System (DBMS)[35], known as a special software designed for user interaction and managing databases. Its purpose is to allow users to create, retrieve, update and manage data, but essentially it serves as an interface between users or between the applications and the databases. Between the different types of DBMS, we opted to choose the relational model (RDBMS) named MySQL which is a popular choice for web-based applications.

MySQL uses a programming language designed specially to manage data held in databases.

Initially, a free integrated front-end tool named MySQL Workbench was used, which gave us the possibility of visually design, manage and create databases. It is also provided with a feature that enables developers to forward and reverse engineering on database allowing to export a graphically designed database to a SQL file that can be used to replicate the database everywhere.

For the visually design part MySQL was very helpful, but due to some restraining we switched to phpMyAdmin that is a free software tool written in PHP, that as the same purpose of MySQL, which is to handle the administration of MySQL but over the web. This tool is also provided with and intuitive web interface that enable to export and import data.

4.2 Database Structure

For better understanding of the created database, the same was divided in two parts. The first type related to users namely patients and therapists, and the second type related to data collected by the Kinect sensor during game play. The database allows storing and management of data on users – patients and therapists, and data collected by the Kinect sensor during game play.

4.2.1 Users Data

Different type of users were considered thus, the first one is the *admin*, that is defined within the game and his only function is to register therapists in the database. The second type with the highest permissions, is the therapist, that as the right to register the patients. The patients will be associated to the therapist that registers them. Therapist can see every type of data patients can see, and are also able to interact with the system regarding certain events, like to establish training plans via android application, register patients, and delete patients profile in case they are no longer needed. The second profile is the *patients*, which have the lower permissions. Patients can have a different set of training plans created by their therapists. Every time the game is logged by each patient, a different session is associated to that same patient. To each session, a set of different games can be associated (Figure 35).

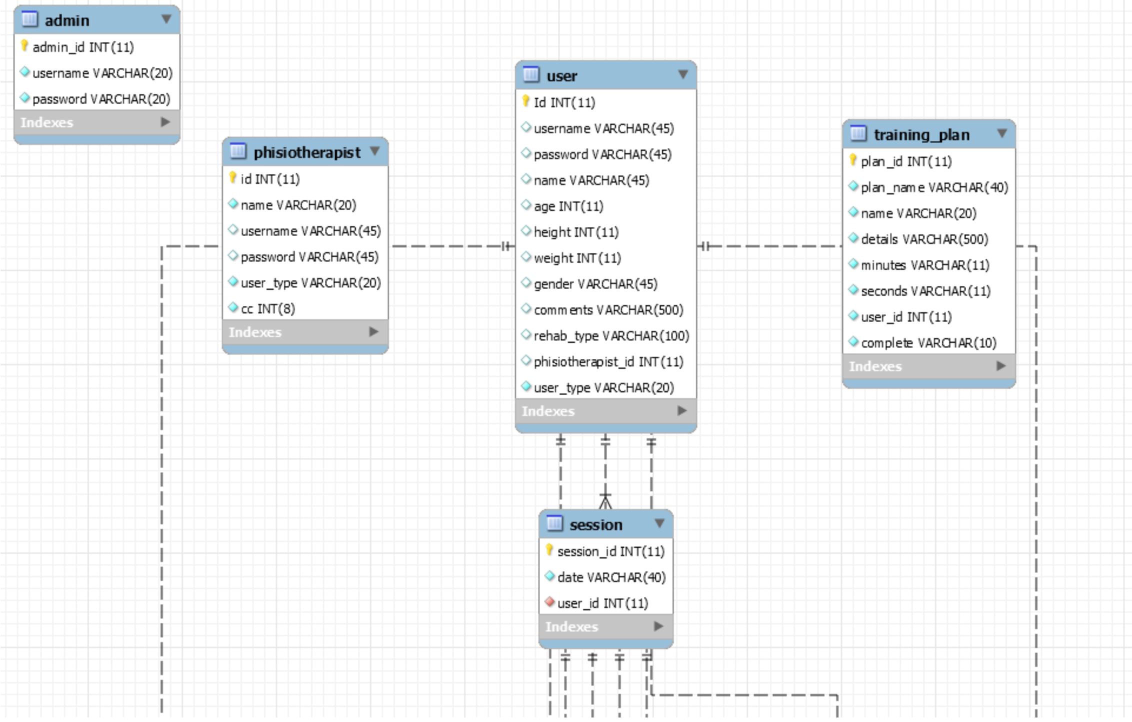

Figure 36.User data model

4.2.2 Game Data

During each session, that is provided with a time stamp when it's created, all games the patient plays without logging out will be associated to the session in task. Thus, the therapist is able to access all patient's data in an organized way. All sessions are also provided with an identifier. As represented in the figure below there are four possible games, which collect two types of data. The first type is related to game customization inputs before each game, objects caught, steps taken. The second type is data collected from the Kinect sensor and is used to produce more technical results, that will be collected, stored and presented in the android application for therapist viewing and analysis.

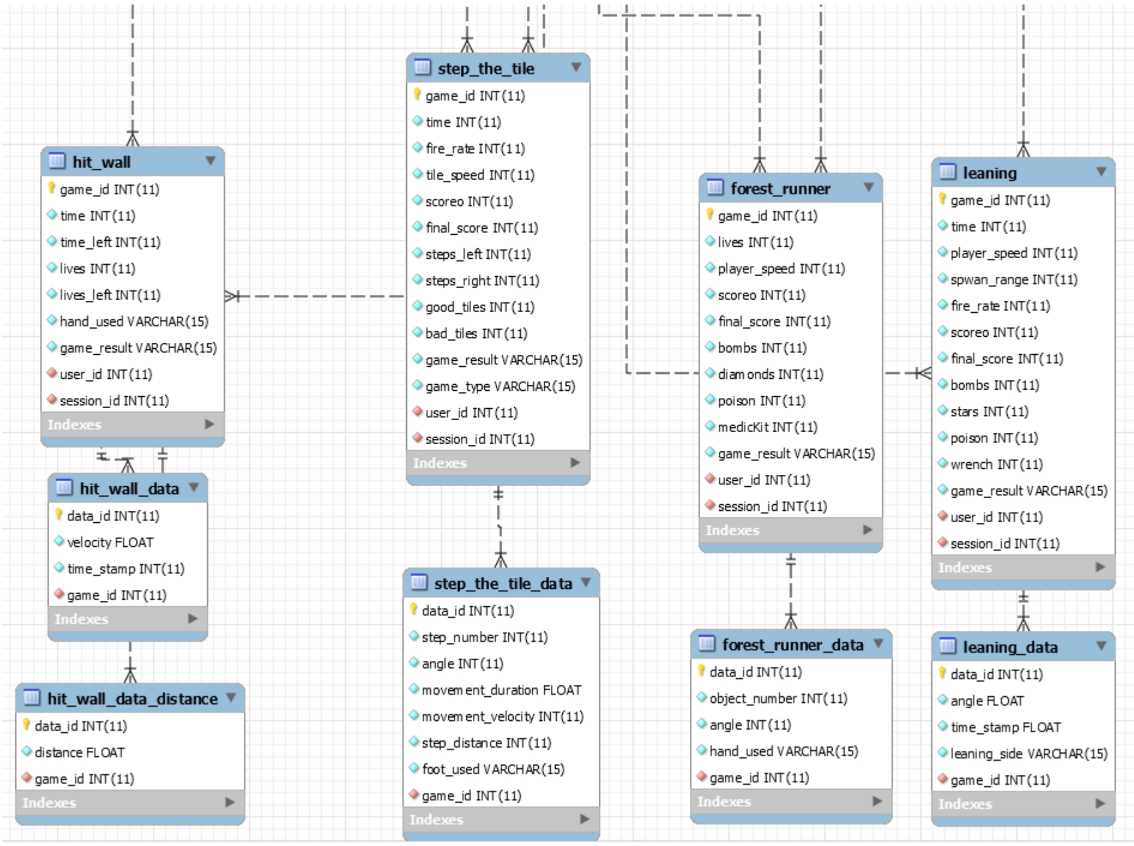

Figure 37.Game data model

4.3 Conclusions

To construct an organizational database system able to store all kinds of information, regarding the data collected directly, either from users input or indirectly from the Kinect sensor, an approach to accomplish that is presented. The advantages of tools like MySQL and phpMyAdmin where explained Description of data organization to facilitate future analysis and consulting is also presented.

5 Scripting and Connections

To establish connection to the server where the developed mobile application and the unity game were included and to manage the data collected, some connections and scripting had to be made both on the game side and mobile application, as is represented on the flow diagram below.

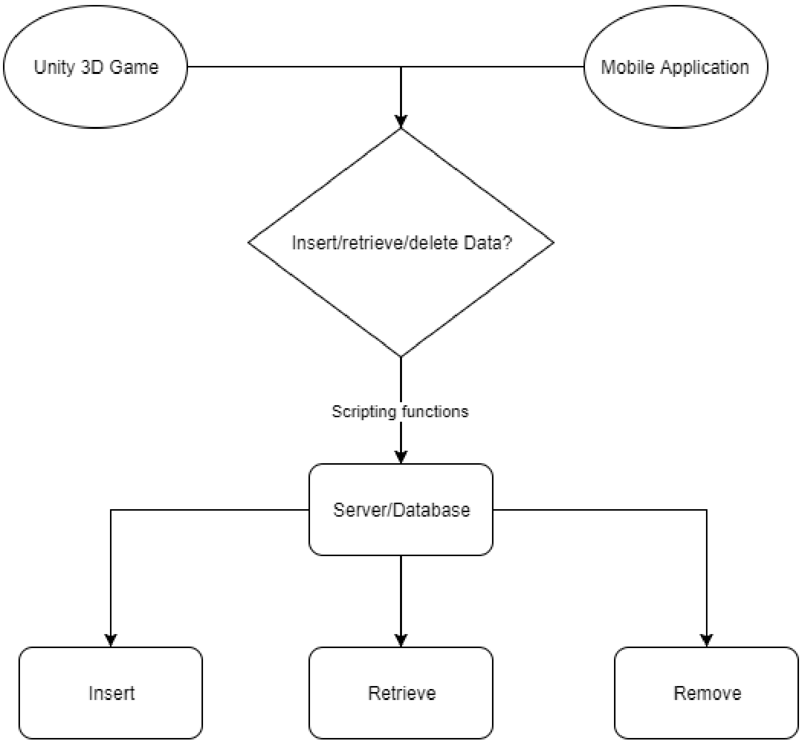

Figure 38.Game and mobile application connection flow chart

A WampServer was used to establish the connections needed, in order to apply the different actions (represented in the flow chart above), by using a large set of PHP scripts, every script with their specific function.

WampServer is a Windows web development environment that makes possible to create web applications with Apache2, PHP and MySQL database, alongside we can also Using this tool also is easy to manage databases in PhpMyAdmin, and to receive support on using local PHP scripts on Windows system [36].

WampServer provides support on using PHP scripts locally. For that a virtual machine was accessed via remote desktop were the WampServer was installed and the PHP scripts allocated.

From the game application view, were the main goal was to retrieve data from the Kinect sensor and allocate the same in the database, a set of scripts were created. In Figure 38 and Figure 39 are represent two of them regarding the algorithms for inserting two types of data in the database, the first figure regarding user registration inputs and the second the *Step the Tile* joint treated data.

```php
<?php
    //variables for connection
    $servername = "localhost";
    $server_username = "root";
    $server_password = "Phisio123";
    $dbName = "therapy_database";


    $username = $_POST["usernamePost"];
    $password = $_POST["passwordPost"];
    $name = $_POST["namePost"];
    $age = $_POST["agePost"];
    $height = $_POST["heightPost"];
    $weight = $_POST["weightPost"];
    $gender = $_POST["genderPost"];
    $comments = $_POST["commentsPost"];
    $rehab_type = $_POST["rehab_typePost"];
    $phisiotherapist_id = $_POST["phisioIdPost"];
    $user_type = $_POST["userTypePost"];

    //Make connection
    $conn = new mysqli($servername,$server_username,$server_password,$dbName);

    //Check connection
        if(!$conn){
            die("Connection Failed.". mysqli_connect_error());
        }

        else echo("Connection success");

        $sql = "INSERT INTO user(username,password,name,age,height,weight,gender,comments,rehab_type,phisiotherapist_id,user_type)
                VALUES ('".$username."','".$password."','".$name."','".$age."','".$height."','".$weight."',
                '".$gender."','".$comments."','".$rehab_type."','".$phisiotherapist_id."','".$user_type."')";
        $result = mysqli_query($conn,$sql);


        ?>
```

Figure 39. PHP script to Insert new patient

A large set of scripts with less instructions were created instead of fewer ones with more commands to have a better code organization.

A set of common points between the different scripts are commented below, related to the variables needed to establish connection to the database. The connection to database and the SQL query allow to insert, retrieve, or delete information from the database.

```php
<?php
	//variables for connection
	$servername = "localhost";
	$server_username = "root";
	$server_password = "Phisio123";
	$dbName = "therapy_database";

	$step_number = $_POST["stepNumberPost"];
	$angle = $_POST["anglePost"];
	$movement_duration = $_POST["movementDurationPost"];
	$movement_velocity = $_POST["movementVelocityPost"];
	$step_distance = $_POST["stepDistancePost"];
	$foot_used = $_POST["footUsedPost"];



	//Make connection
	$conn = new mysqli($servername,$server_username,$server_password,$dbName);

	//Check connection
		if(!$conn){
			die("Connection Failed.". mysqli_connect_error());
		}

		else echo("Connection success");


		$qry = "SELECT MAX(game_id) FROM step_the_tile";
		$result = mysqli_query($conn,$qry);

		if($result){
			while($row = mysqli_fetch_array($result)){
			echo $row[0];

			mysqli_query($conn, "INSERT INTO step_the_tile_data(step_number,angle,movement_duration,movement_velocity,step_distance,foot_used,game_id) VALUES
			('".$step_number."','".$angle."','".$movement_duration."','".$movement_velocity."'
			,'".$step_distance."','".$foot_used."','".$row[0]."')");
			}
		}else{
			echo ("i didnt make it");
		}
```

Figure 40. PHP script to insert step the tile game data

The two scripts described above represent actions regarding inclusion of data in the database. The one represented in Figure 40 represents a retrieval of information, in this case regarding the user. This type of information is then treated and displayed, as in the bellow example, on the profile activity of the mobile application.

```php
<?php
	//variables for connection
	$servername = "localhost";
	$server_username = "root";
	$server_password = "Phisio123";
	$dbName = "therapy_database";



	//Make connection
	$conn = new mysqli($servername,$server_username,$server_password,$dbName);

	//Check connection
		if(!$conn){
			die("Connection Failed.". mysqli_connect_error());
		}

		$sql = "SELECT id,username,password,name,age,height,weight,gender,comments,rehab_type,phisiotherapist_id,user_type FROM user ";
		$result = mysqli_query($conn,$sql);

		if(mysqli_num_rows($result)>0){
			//show data for each row
			while($row = mysqli_fetch_assoc($result)){
				echo "Id:".$row['id'] ."|Username:".$row['username']."|Password:".$row['password']."|Name:".$row['name'].
				"|Age:".$row['age']."|Height:".$row['height']."|Weight:".$row['weight']."|Gender:".$row['gender'].
				"|Comments:".$row['comments']. "|RehabType:".$row['rehab_type']."|PhisioID:".$row['phisiotherapist_id']."|UserType:".$row['user_type']. ";";

			}


		}


		?>
```

Figure 41. PHP script to retrieve patient information

Figure 41 illustrates a script where data is removed from the database, in this case all the user related data.

```php
<?php
    //variables for connection
    $servername = "localhost";
    $server_username = "root";
    $server_password = "Phisio123";
    $dbName = "therapy_database";

    $Id = $_POST["IdPost"];

    //Make connection
    $conn = new mysqli($servername,$server_username,$server_password,$dbName);

    //Check connection
        if(!$conn){
            die("Connection Failed.". mysqli_connect_error());
        }

        $sql = "DELETE FROM user WHERE Id = '".$Id."' ";
        $result = mysqli_query($conn,$sql);


        if(($result)){
            //show data for each row

                echo "Patient Deleted";

        }

        ?>
```

Figure 42. PHP script to delete user from database

While creating all the scripts, the same were called from both game and mobile application when needed. To call the different scripts the URL of each script had to be presented in the following way either on the mobile application or the Unity 3D game

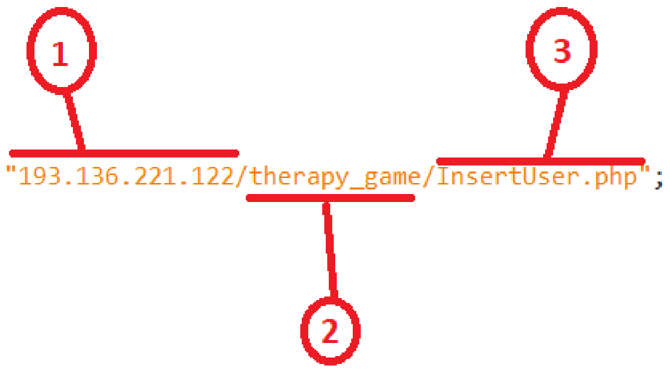

Figure 43.String address URL for Insert User PHP script

- 1 - Server Ip address
- 2 - Folder located in the WampServer install folder containing all the PHP scripts created
- 3 - Php script

In the Unity game, after the script was provided, a representation of the function described in Figure 43, is called.

```
public void SaveUserToDatabase(){
    WWWForm form = new WWWForm ();

    form.AddField ("usernamePost", Username.text);
    form.AddField ("passwordPost", Password.text);
    form.AddField ("namePost", Name.text);
    form.AddField ("agePost", Age.text);
    form.AddField ("heightPost", Height.text);
    form.AddField ("weightPost", Weight.text);
    if (Male.isOn) {
        form.AddField ("genderPost", "Male");
    }

    if (Female.isOn) {
        form.AddField ("genderPost", "Female");
    }

    if (Superior.isOn) {
        form.AddField ("rehab_typePost", "Superior");
    }
    if (Inferior.isOn) {
        form.AddField("rehab_typePost" , "Inferior");
        }
    if(Inferior.isOn && Superior.isOn){
        form.AddField("rehab_typePost" , "Both");
    }

    int phisioid = int.Parse(PlayerPrefs.GetString ("PhisioId"));

    form.AddField ("commentsPost", Comments.text);
    form.AddField ("phisioIdPost", phisioid);
    form.AddField ("userTypePost", "Patient");


    WWW www = new WWW (createUserURL,form);

}
```

Figure 44.Unity code part for register patient in the database

In Figure 44 is represented a class called from the android application side, when some interaction with database is needed. This type of class also extends an Async Task. An AsyncTask enables proper and easy use of the User Interface thread.

```
private class DonwloaderProfile extends AsyncTask<String,Void,String> {
    Context context;
    AlertDialog alertDialog;
    private  String Username;
    DonwloaderProfile(Context ctx) { context = ctx; }

    @Override
    protected String doInBackground(String... params) {
        String typeOfmethod = params[0];
        String patientFetchDataURL = "http://193.136.221.122/therapy_game/UserData.php";
        String patientDeleteURL = "http://193.136.221.122/therapy_game/DeletePatient.php";
        if(typeOfmethod.equals("fetchPatientData")){
            try {
                String username = params[1];
                Username = params[1];

                URL urlLogin = new URL(patientFetchDataURL);
                HttpURLConnection con = (HttpURLConnection) urlLogin.openConnection();
                con.setRequestMethod("POST");
                con.setDoInput(true);
                con.setDoInput(true);
                OutputStream outputStream = con.getOutputStream();
                BufferedWriter bufferedWriter =new BufferedWriter(new OutputStreamWriter(outputStream,"UTF-8"));

                String post_data = URLEncoder.encode("usernamePost","UTF-8")+"="+URLEncoder.encode(username,"UTF-8");
                bufferedWriter.write(post_data);
                bufferedWriter.flush();
                bufferedWriter.close();
                outputStream.close();
                //to get the response from php
                InputStream stream = con.getInputStream();
                BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(stream,"iso-8859-1"));

                //reah response line by line;
                String result ="";
                String line = "";
                while((line = bufferedReader.readLine()) != null){
                    result+=line;
                }
                bufferedReader.close();
                stream.close();
                con.disconnect();
                return result;
```

Figure 45. Code part from android studio for patient data retrieval

This class allows to perform background operations and publish the results on the UI thread without having to manipulate threads or/and handlers.

5.1 Conclusions

In this chapter we discussed the different tools required to have access to the remote therapy database. It was also approached the different scripting interactions with the database and explained how they were called both in the application and Unity Game. The structure of the PHP scripts was explained as well.

6 Mobile Application

A mobile application is important in this system, serving as an interface between the data collected during sessions and the therapist. The importance of a mobile application is to allow users to access data from everywhere. For the therapist it's a very important tool, allowing him to keep track of each patient by recurring to a practical and simple system. Through this the therapists will be able to give more attention on patients and possibly provide better feedback on how the patients' sessions are going. For the patient, can also be a motivating tool, by allowing to keep track of their game results, and see their constant evolution during each game.

6.1 Structure

This mobile application was developed to embody the interface between the users and the game system. Figure 45 emphasizes the structure of the mobile application for easier comprehension of it. In mobile application, the therapists and patients, as was mentioned before, each one has different permissions. Upon *Therapist Login*, a list of associated patients is presented allowing him to take a set of actions towards each one. Some actions like create training plans for the patient where detailed information can be inserted regarding each game and its game play. Consult patients profile may contain helpful information for the rehabilitation process. Deleting patients from the database, erasing all their records within the database when therapists see their records are no longer needed. These are secondary tasks. The main task is actually the visualization of patient's data sessions, where data will be presented in graphs. It is intended that with the sessions' progression, physiotherapists can observe a clearer evolution on patient's medical state.
If the logged user is a patient, their permissions regarding certain features are revoked, as is represented in Figure 47. The patients might access their plans to see the therapist inputs and comments and if the training plan is completed. Physiotherapist can also check that option, updating the database. The patients can also have access to their own profile and game stats from each individual session they took. However, they don't have permission to consult the graphs, but only the scores, game configuration inputs and so forth.

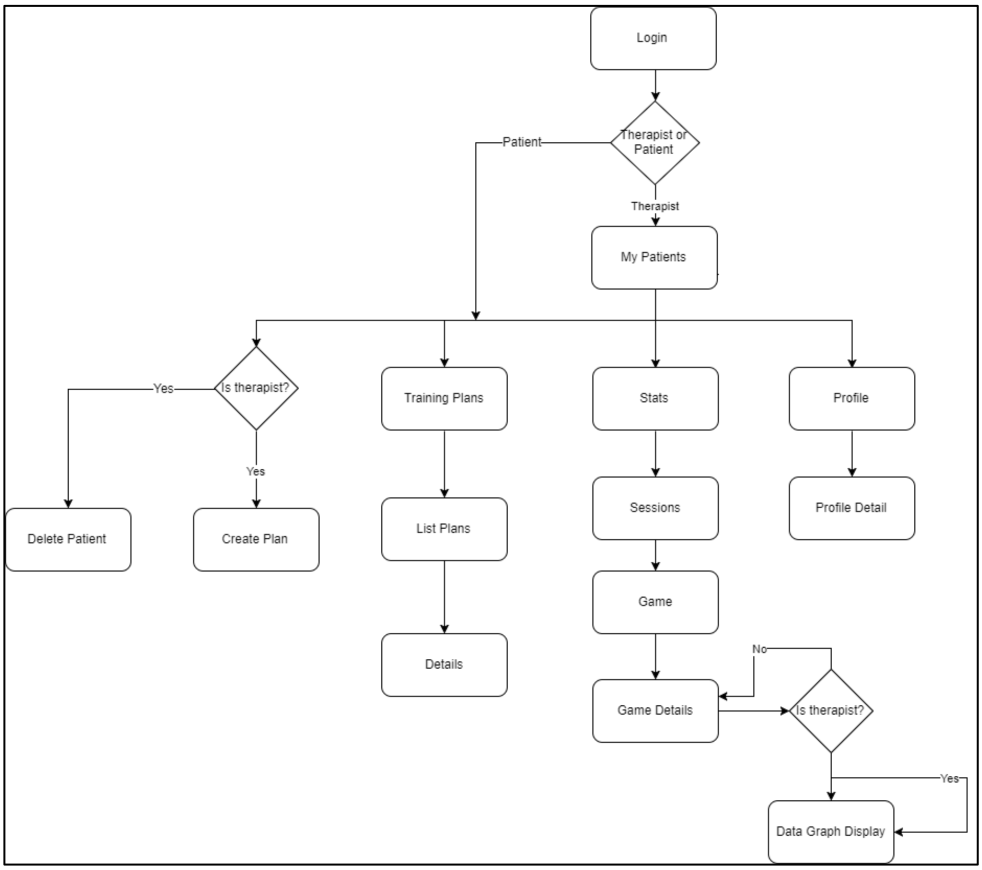

Figure 46. Mobile application flow chart

6.2 Graphical User Interface

The user interface is one of the most important aspects of a mobile application [37]. So, a clear and easy interface was thought and developed in order to achieve the best possible interaction between user and displayed data. Displaying simple and direct data for users to interact was also a careful viewpoint while building this app.

In Figure 46 is illustrated the *Login* display, where each user can enter their credentials in order to prepare the app considering the user type.

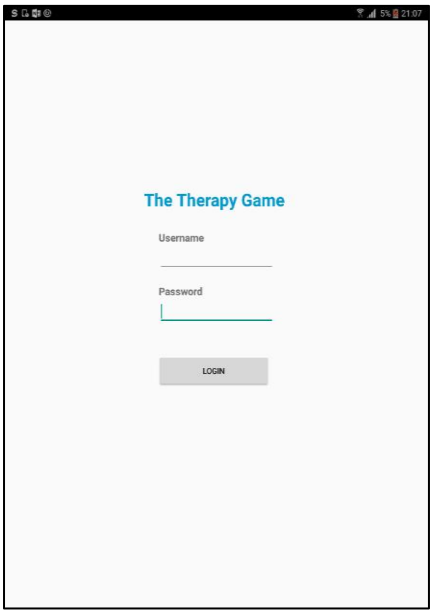

Figure 47.Mobile application login interface

Figure 47 illustrates the user area after successful login. The user area displayed is a therapist area, where features are activated after *Login in*, such as number two, three and four represented in the figure below and mentioned in the previous section. Marked with numbers 1 and 5 are the common features for both patient and therapist in this activity, being number 1 the placeholder for logged user and number 5 the three different card views leading to other mobile activity regarding patient's data.

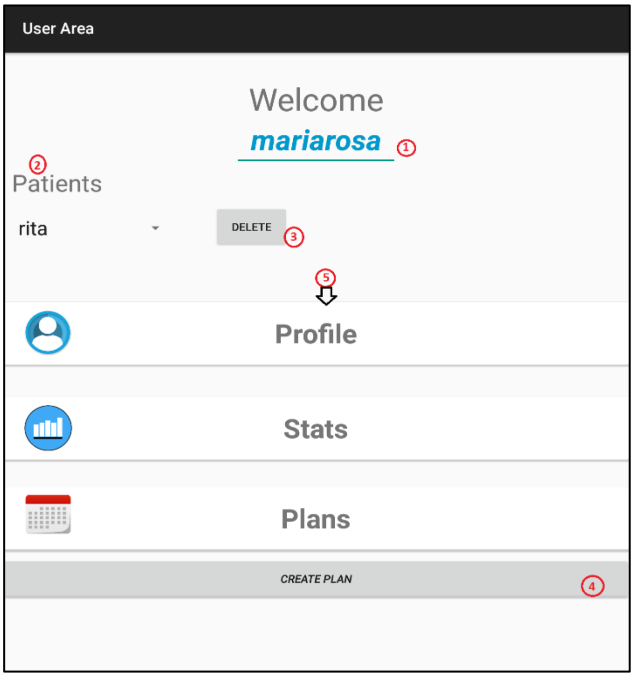

Figure 48. User area interface

The therapist has the option of creating a training plan that will be consulted by the patient or his responsible. Figures 48 illustrates the create plan activity where the therapist is free to configure the patients plan wherever he likes. The use of free fields will allow therapist to insert a plan info for patients, depending on the patient and using the type of language one would understand.

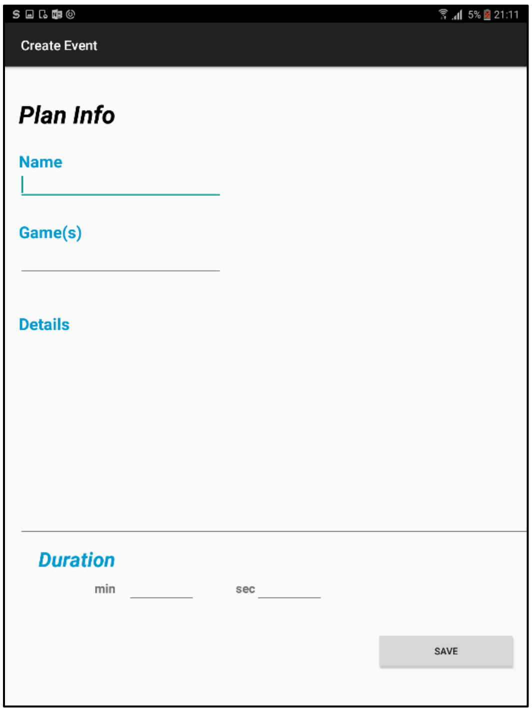

Figure 49. Plan info interface

Regarding to game stats, this activity is illustrated in Figure 49. After changing from the user area activity to the one represented below, the patient's correspondent sessions will be automatically displayed on the first spinner, passing that, the therapist/patient chooses witch game wants to consult. After selecting a game, the last spinner will be filled with the game *ids* regarding the game choice. Finally, the displayed data could be seen as in Figure 50. Figure 50 illustrates, an example of game data displayed after a game is selected. As can been seen at the bottom of the figure is a button called *graphs*, that is one of the features only displayed to therapists. By clicking on this button are presented graphs, relating to different data collected by the Kinect sensor while playing the game in question. Depending on the type of data collected there are different types of graphs that can be displayed as could be seen in the section that follows.

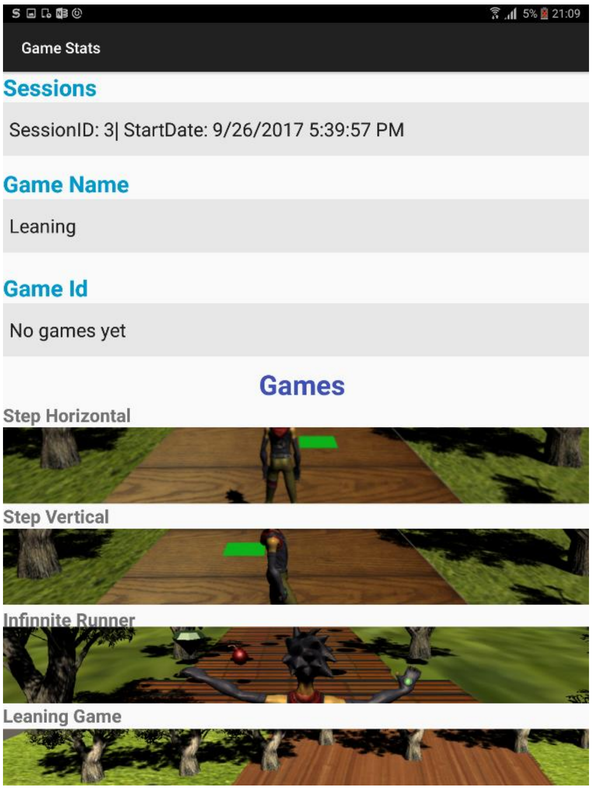

Figure 50.Games stats interface

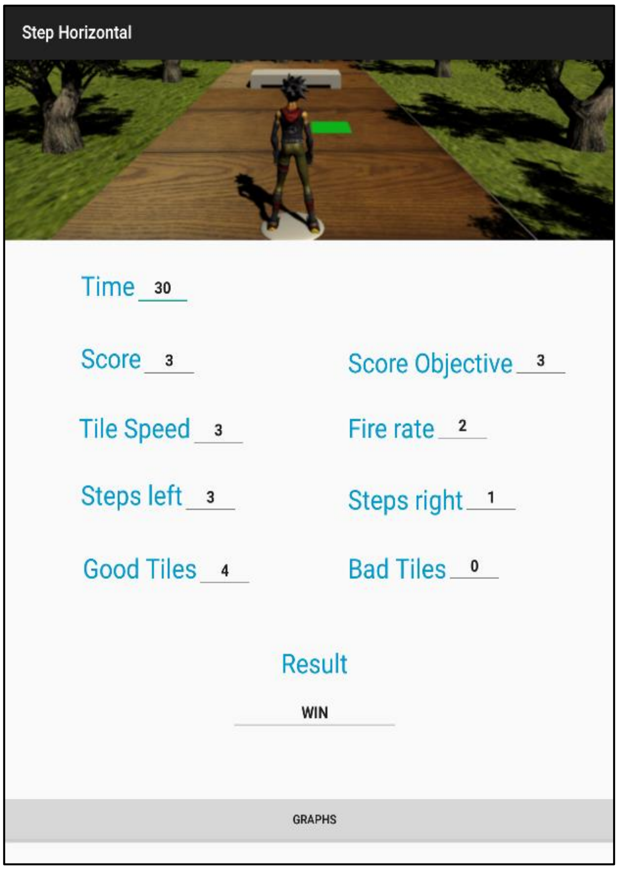

Figure 51.Step the tile game data interface

6.3 Graphs

The data that should be displayed it is important to be clear, synthesized and objective, for the therapist and patient understanding, and have access to it. Regarding the graphs, for the therapist consulting a graph view library was used that provide two types of graphs, *Line graphs* and *Point graphs*. *Line graphs* mostly used to display data during certain periods of time, like player velocity during hit wall game – as is represented in Figure 52 and *Point graphs* to illustrate fixed points events, like catching an object or stepping a tile as is shown in Figure 51. By touching each data point, this information is displayed dynamically through a TOAST message, displaying, in this case, step number, distance reached, and the foot used to do so regarding the activity, where a line graph is represented. It also presents the traveled movement distance registered during the play time.

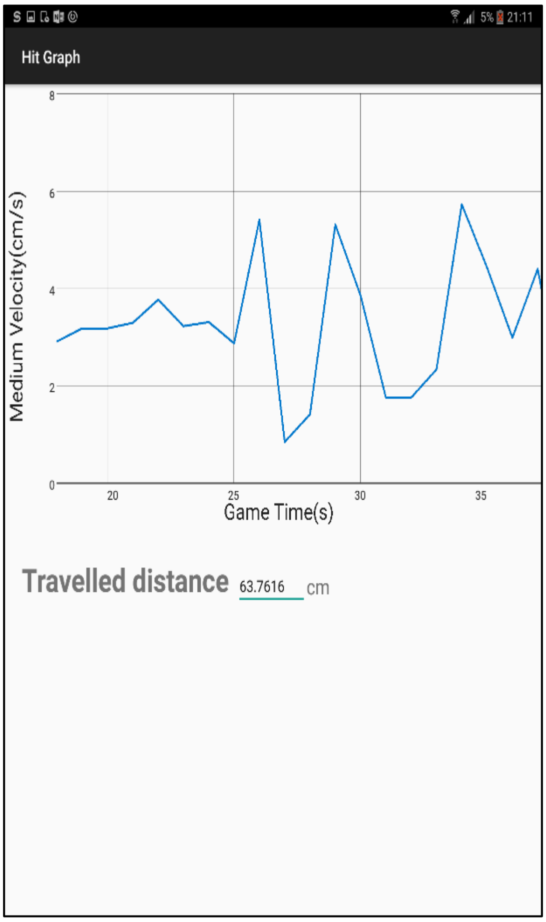

Figure 52. Step the tile graph interface

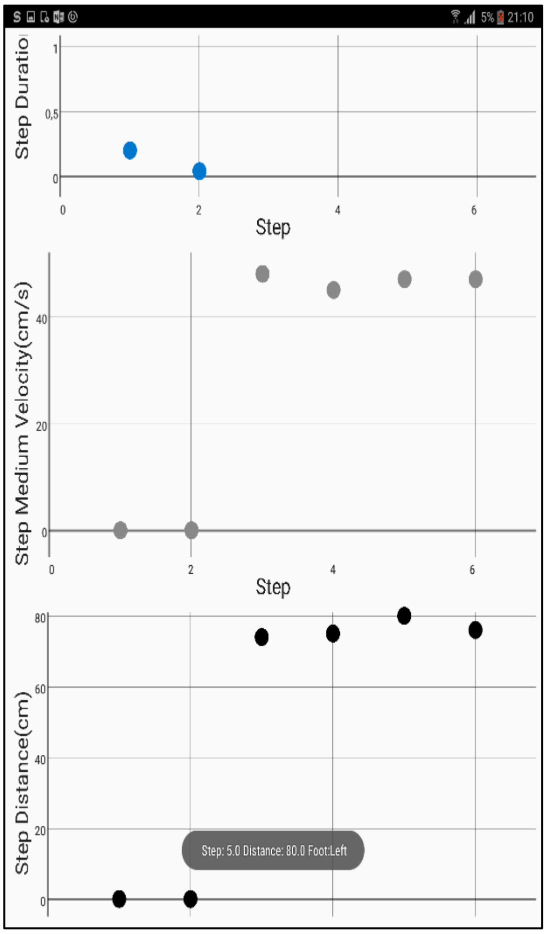

Figure 53. Hit graph interface

6.4 Conclusion

In this chapter an overview of the mobile application structure is explained. The different UIs are explained and the different features each one of them present as well as their permissions towards the type of user. It is also presented the different types of data and the means to display it in order to give the user a better experience.

7 Results

In this section are presented the results associated to the set of tests made with the Kinect Sensor for all the four games within "The Serious Game".

For each section, a range of results related to each game will be presented. Displaying, data input and output data results, presented in graph form. All game results are associated with one Male subject.

7.1 Step the Tile

The game results presented next, are based on the following inputs and on the **Step Horizontal** game, having the **Vertical** side of the game presented similar results.

- Time: 50
- Score Objective: 6
- Tile Speed: 3
- Fire Rate: 2

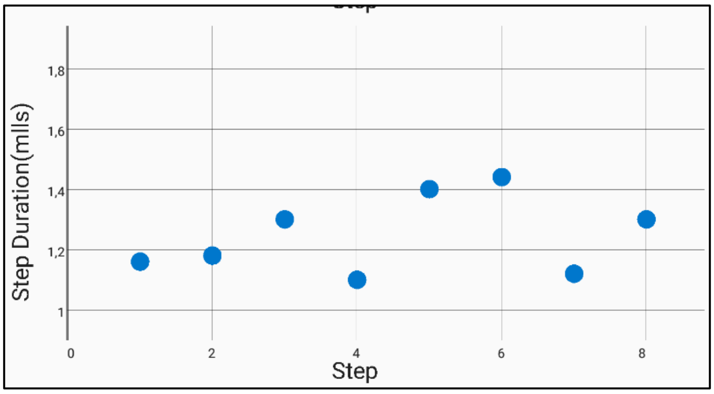

Figure 54. Step Duration per Step graph

In Figure 53 is represented the numbers of steps made as well as the distance reached by the test subject. For a more detailed data the therapist has the option of selecting each data point to have access to it. Normalized data regarding the step duration, and the foot used to do so are displayed.

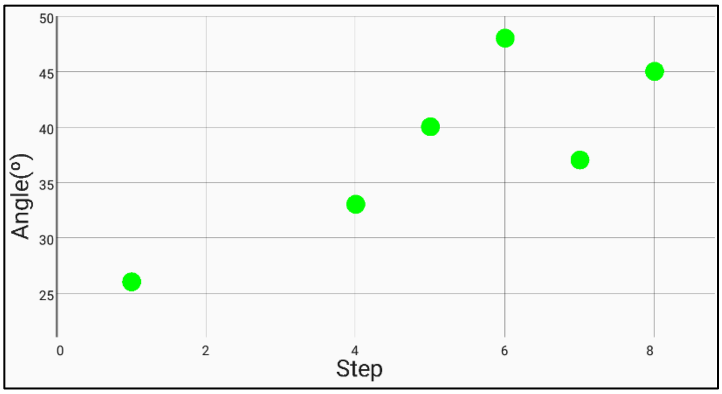

Figure 55. Angle per Step graph

Figure 54 present the angle made by each step and foot associated to it. More detailed on the importance of angle representation can be found in Game related section.

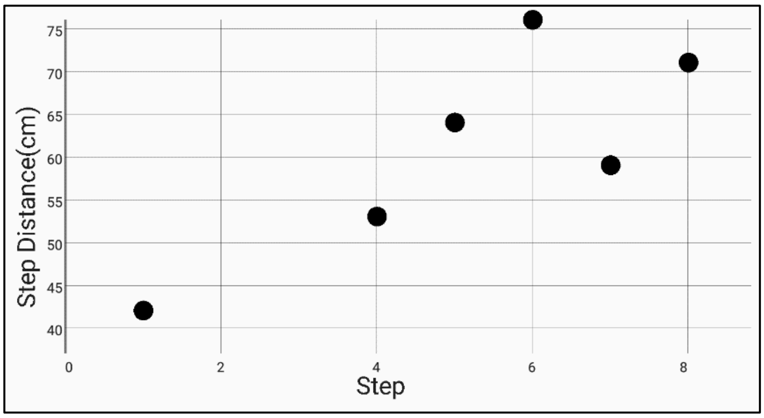

Figure 56. Step Distance per Step graph

Figure 55 illustrates the step distance related to each step made by the patient.

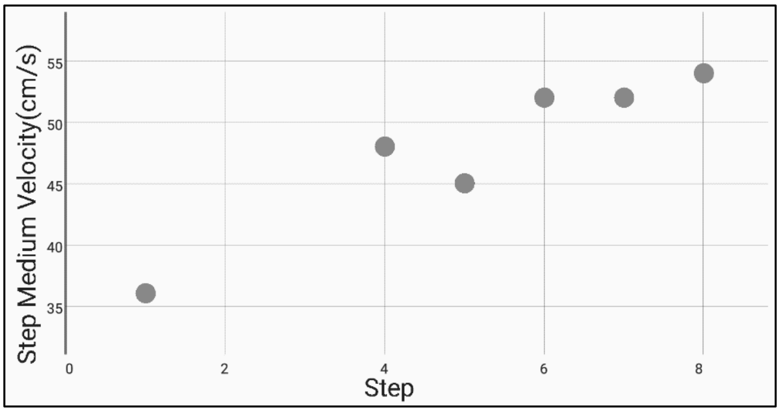

Figure 57. Step velocity per Step

The graph illustrated in Figure 56, displayed for this game type, the relation between each data point with the velocity, when the subject effectuate each step. By analyzing these graphs is expected that the therapists obtain conclusive data related to the patient step range, mobility and their evolution through each session.

7.2 Leaning Game

This game, focus mainly on body balance and the lower limbs movements. In Figure 57 is represented some results of the game playing. To obtain this results the following set of inputs were used:

- Time: 50
- Player Speed: 3
- Points Required: 2
- Object Spawn Range: 2
- Fire Rate: 2

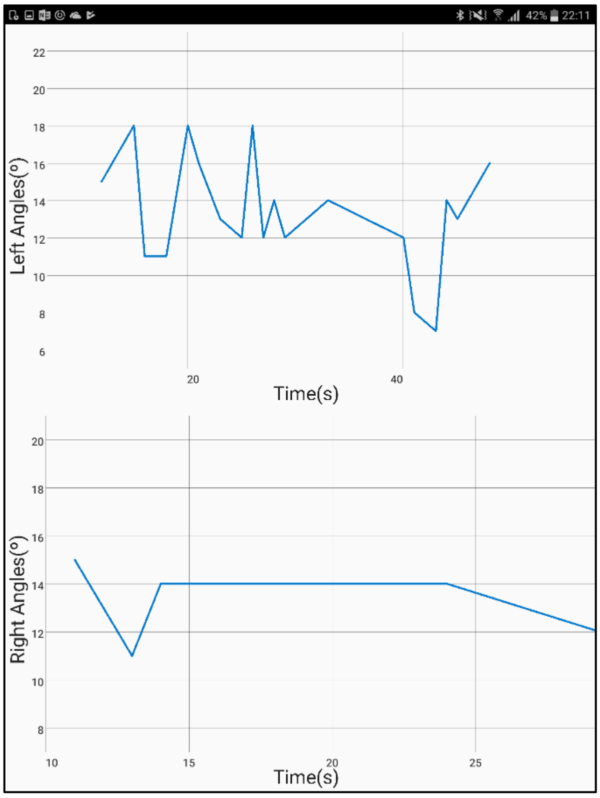

Figure 58. Leaning Left/Righ graph interface

Observing Figure 57, we can conclude that during most of the game time, the patient tended to lean more to the left side. Not any big values variation is notice when it comes to right side leaning. The therapist can observe which side of the patient's body is more comfortable for patient by regarding the data presented.

7.3 Infinite Runner

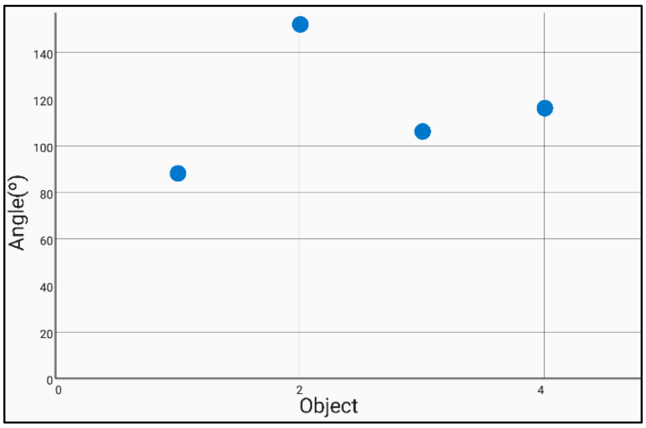

Figure 59. Infinite Runner Angle per Object caught graph

In Figure 58 are displayed data points related to patients caught objects. By clicking in each data point the therapist can see a more detailed information related to each data point, display of the arm used and angle value. As we can see the tested subject caught objects from a range of 80 to 160 degrees. Just based on this data we can assume the patient has a good left/right arm mobility, capable of reaching a set of different ranges taking into account that the higher angles are always most difficult to reach

7.4 Hit Wall

The game inputs used for testing this game are the following:

- Lives :5
- Time :50
- Lives Left: 2

- Score Objective:4
- Hand Used: Left

Figure 59 illustrates the path trace by the player (catching the sphere). This was achieve by saving each player position since it began to move until the arm movement is stopped (game ended). The image that can be accessed by therapists and can be used as auxiliary material for the analysis of the graph is represented in Figure 58.

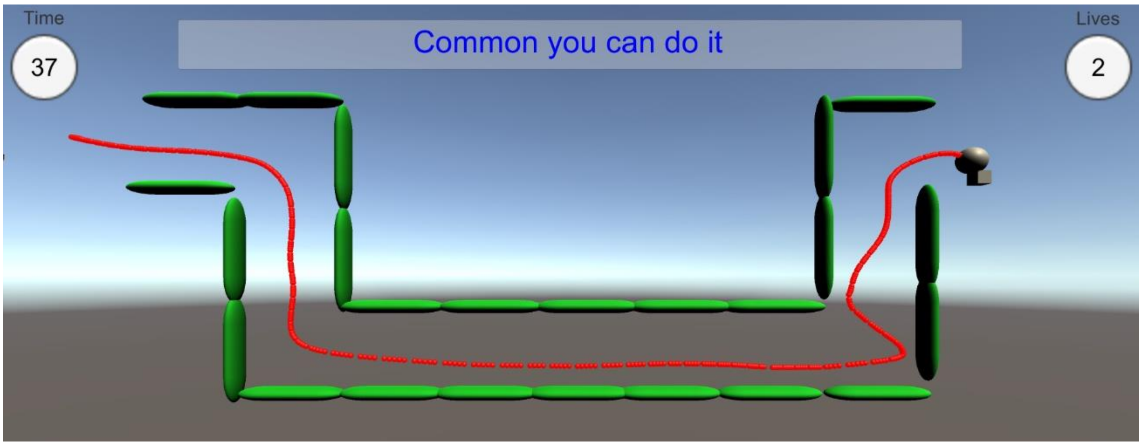

Figure 60. Game path image display

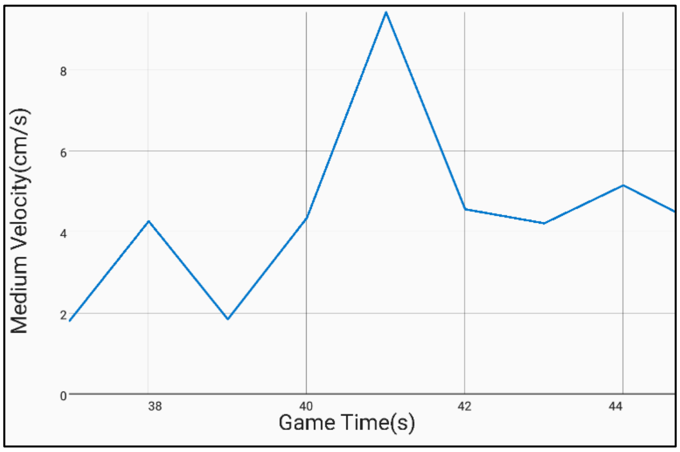

Figure 61. Velocity per game time graph

Figure 62. Path travelled distance

In Figure 59 is displayed the path, referring to the path travelled distance represented in Figure 61.

Observing the graph in Figure 60, and complementing with Figure 59 we can conclude the patient has a good left arm mobility. This information is derived from the time it took to complete the game (13 seconds). By also consulting the game image we can observe where in the course, he has gone faster.

7.5 Usability Tests

Evaluation of the developed software was realized. The usability tests were conducted using a Portuguese translation of the Questionnaire for User Interaction Satisfaction (QUIS). This questionnaire developed by a multi-disciplinary team of researchers in the Human-Computer Interaction Lab (HCIL) at the University of Maryland at College Park, assess users' subjective satisfaction with specific aspects of the human-computer interface. The QUIS is highly reliable across many types of interfaces. The QUIS questionnaire was included in the questionnaire developed by the team of research project, financed by Fundação para a Ciência e a Tecnologia, TailorPhy – Smart Sensors and Tailored Environment for Physiotherapy. The questions of TailorPhy questionnaire address the emotion that the application produced (by using validated Portuguese questionnaire of PANAS-VRP, Positive and Negative Affect Schedule), the arousal, and the socio-demographic data. The questionnaire for therapists and the questionnaire applied to other type of participants in usability tests (healthy volunteers or physiotherapy patients) has 23 questions among these 13 (those related to QUIS, PANAS-VRP, and arousal scale) are equal in both questionnaires.

The tests of developed theragames were realized with 9 participants (3 therapists and 6 healthy volunteers). In the preliminary results the participants greatly appreciated the *Step the Tile* game. For all games the participants positively appreciate the facility of games configuration, the tailoring of the games to the user needs and preferences as well as to the type of therapy. The game ideas, aim of game, scenes, colors, sound included in games as feedback, help messages, variety of games, games interaction was described as the most three positive things in the application. Some problems were identified but the results were inconclusive because it was not clear for what game these problems were registered. The team of research project proposed that other tests should be realized in the future for each of the game in order to better identify and correct the application features that subjectively the participants found as inconsistent with the game purpose and user need. General opinions on the application were good and the participants consider that this tool may increase the motivation of patients for training movements for motor rehabilitation. [38][39]

8 Conclusion and Future Work

8.1 Conclusion

Theragames are having great importance for physical rehabilitation. Microsoft Kinect sensor is an important tool for Theragames, because it allow a different, and yet appropriate, patient examination and evaluation during rehabilitation sessions. By integrating Microsoft Kinect with Unity 3D game engine, permitted us to integrate real life movements into different game environments, giving to the patient a completely different approach of a rehabilitation session. The developed software based on Kinect and Unity 3D was complemented with a mobile application in order to have a user-friendly tool for therapists manage the patient's data through each session.

8.2 Future Work

In the future, is intended to focus more on the game environments and movement precision, so the therapist can have better quality data, to provide better support for their patients. Also, is planned to improve the application interface as well as the data display to provide more meaningful data and to turn the application in one with more easy interaction.

The game has many ways of improvement. The Kinect~~,~~ allow more gestures to be configured, and thus more games can be set up, with different environments and characters that could be tailored to the need of a wider set of patients.

9 References

[1] Apta.org. (2017). Role of a Physical Therapist. [online] Available at: http://www.apta.org/PTCareers/RoleofaPT/

[2] O. Postolache, F. Lourenco, J. Dias Pereira, P. Girao, P. "Serious game for physical rehabilitation: Measuring the effectiveness of virtual and real training environments", Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2017.

[3] P. Rego, P. M. Moreira, L. P. Reis "Serious_Games_for_Rehabilitation A_survey_and a classification towards a_taxonomy", Proceedings of 5th Iberian Conference on Information Systems and Technologies (CISTI), 2010

[4] F. Cary; O. Postolache, P. M. Girão, "Kinect Based System and Serious Game Motivating Approach for Physiotherapy Assessment and Remote Session Monitoring", Proc International Conf. on Sensing Technology - ICST, Liverpool, UK, Vol. 1, pp. 1 - 5, 2015

[5] N. Duarte, O. Postolache, J. Scharcanski, "KSGphysio – Kinect Serious Game for Physiotherapy", Proc International Workshop on Electromagnetic Compatibility and Engineering in Medicine and Biology - EPE-W, Iasi, Romania, Vol. 1, pp. 1, 2014.

[6] D. Webster,O. Celik, "Systematic review of Kinect applications in elderly care and stroke rehabilitation", Journal of NeuroEngineering and Rehabilitation,No.11, 2014

[7]] Physiopedia.com, The emerging role of Microsoft Kinect in physiotherapy rehabilitation for stroke patients - Physiopedia. [online] at https://www.physiopedia.com/The_emerging_role_of_Microsoft_Kinect _in_physiotherapy_rehabilitation_for_stroke_patients, 2017

[8] Mousavi Hondori, Hossein and Khademi, Maryam, 2014, A Review on Technical and Clinical Impact of Microsoft Kinect on Physical Therapy and Rehabilitation. *Journal of Medical Engineering*. 2014. Vol. 2014, p. 1-16. DOI 10.1155/2014/846514. Hindawi Publishing Corporation. Available from: https://www.hindawi.com/journals/jme/2014/846514/

[9] 3 Physical Therapy Technologies Improving Patient Care | SHC Blog, 2016. *Blog.supplementalhealthcare.com* [online], Available from: http://blog.supplementalhealthcare.com/patient-care-forum/3-physical-therapy-technologies-improving-patient-care

[10] Dobkin, Bruce H., 2005, Rehabilitation after Stroke. *New England Journal of Medicine*. 2005. Vol. 352, no. 16, p. 1677-1684. DOI 10.1056/nejmcp043511. New England Journal of Medicine (NEJM/MMS) Available from: http://www.nejm.org/doi/full/10.1056/NEJMcp043511

[11] How the Kinect, Azure and Gamification are helping VirtualRehab to rehabilitate patients worldwide - Microsoft UK Developers, 2016. *Microsoft.com* [online],Available from: https://www.microsoft.com/en-gb/developers/articles/week05may14/how-the-kinect-azure-and-gamification-are-helping-virtualrehab-to-rehabilitate-patients-worldwide/

[12] 2016. *Asian Healthcare Management System* [online], Available from: https://www.asianhhm.com/articles/role-information-technology-medical-sciences

[13] Laamarti, F., Eid, M. and El Saddik, A. (2017). *An Overview of Serious Games*.

[14] Hatem, S., Saussez, G., della Faille, M., Prist, V., Zhang, X., Dispa, D. and Bleyenheuft, Y. (2016). Rehabilitation of Motor Function after Stroke: A Multiple Systematic Review Focused on Techniques to Stimulate Upper Extremity Recovery. *Frontiers in Human Neuroscience*, 10.

[15] M.D.J McNeill, D.K. Charles, P.J. Morrow, J.H. Crosbie, S.M. McDonought J.w. Burke,"Optimizing engagement for stroke rehabilitation using serious games," The Visual Computer, pp. 1085-1099,Aug.2001

[16] Wiemeyer, J. and Kliem, A. (2011). Serious games in prevention and rehabilitation—a new panacea for elderly people?. *European Review of Aging and Physical Activity*, 9, pp.41-50.

[17] Learning Solutions Magazine. (2017). *Gamification, Game-based Learning, Serious Games: Any Difference?*. [online] Available at: https://www.learningsolutionsmag.com/articles/1337/gamification-game-based-learning-serious-games-any-difference

[18] Physio-pedia.com. (2017). *The emerging role of Microsoft Kinect in physiotherapy rehabilitation for stroke patients - Physiopedia*. [online] Available at: https://www.physio-pedia.com/The_emerging_role_of_Microsoft_Kinect_in_physiotherapy_rehabilitation_for_stroke_patients#cite_ref-Microsoft2015_2-1

[19] Mickey, L. (2017). *Physical Therapists Use Wii Golf to Treat Patients*. [online] Nytimes.com. Available at: http://www.nytimes.com/2012/09/09/health/physical-therapists-use-wii-golf-to-treat-patients.html?_r=0

[20] Alankus, G., Lazar, A., May, M. and Kelleher, C. (2010). Towards customizable games for stroke rehabilitation. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*.

[21] Staiano, A. and Flynn, R. (2014). Therapeutic Uses of Active Videogames: A Systematic Review. *Games for Health Journal*, 3(6), pp.351-365.

[22] Saposnik, G., Teasell, R., Mamdani, M., Hall, J., McIlroy, W., Cheung, D., Thorpe, K., Cohen, L. and Bayley, M. (2010). Effectiveness of Virtual Reality Using Wii Gaming Technology in Stroke Rehabilitation: A Pilot Randomized Clinical Trial and Proof of Principle. *Stroke*, 41(7), pp.1477-1484.

[23] Physiopedia.com, The emerging role of Microsoft Kinect in physiotherapy rehabilitation for stroke patients - Physiopedia. at https://www.physiopedia.com/The_emerging_role_of_Microsoft_Kinect_in_physiotherapy_rehabilitation_for_stroke_patients, 2017

[24] Huang, J. (2011). Kinerehab. *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '11*.

[25] Softkinetic.com. (2017). *SoftKinetic - 3D Vision Leader*. [online] Available at: https://www.softkinetic.com [Accessed 27 Oct. 2017].

[26] Walter, D. (2017). *Product Comparison – Kinect and Leap Motion | Ashland Tech*. Available at: http://ashlandtech.org/2014/04/02/product-comparison-kinect-and-leap-motion/

[27] ComputerWeekly.com. (2017). *Smartphones or handheld computers?*. Available at: http://www.computerweekly.com/feature/Smartphones-or-handheld-computers

[28] Boulos, M., Wheeler, S., Tavares, C. and Jones, R. (2011). How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. *BioMedical Engineering OnLine*, 10(1), p.24.

[29] D. Pagliari, L. Pinto, "Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors", Vol 15, Issue 11, pp. 27569-27589, 2015.

[30] M. J. Landau, B. Y. Choo, P. A. Belling, "Simulating Kinect Infrared and Depth Images" ,IEEE Transactions on Cybernetics, Vol: 46, Issue: 12, pp. 3018 - 3031, 2016.

[31]O.Wasenmüller,D.Stricker,"Comparison_of_Kinect_v1_and_v2_Depth_Images_in_Terms_of_Accuracy_and_Precision" , Computer Vision – ACCV 2016 Workshops pp 34-45, 2017

[32]Gamesparks, "Game Engine Analysis and Comparison | Game Engines | GameSparks" Available at: https://www.gamesparks.com/blog/game-engine-analysis-andcomparison/ 2017

[33] Assetstore.unity3d.com. (2017). *Asset Store*. Available at: https://www.assetstore.unity3d.com/en/#!/content/18708

[34]Unity. (2017). *Unity - Learn - Modules*. [online] Available at: https://unity3d.com/pt/learn/tutorials [Accessed 29 Oct. 2017].

[35] SearchSQLServer. (2017). *What is database management system (DBMS)? - Definition* Available at: http://searchsqlserver.techtarget.com/definition/database-management-system

[36] WampServer. (2017). *WampServer*. [online] Available at: http://www.wampserver.com/en/

[37] UX Planet. (2017). *Mobile UX Design: Key Principles – UX Planet*. [online] Available at: https://uxplanet.org/mobile-ux-design-key-principles-dee1a632f9e6

[38] John P. Chin, Virginia A. Diehl, Kent L. Norman. Development of a Tool Measuring User Satisfaction of the Human-Computer Interface. Proceeding of SIGHI Conference on Human Factors in Computing Systems, CHI'88, pages 213-218, 1988.

[39] Galinha, I. C., Pereira, C., & Esteves, F. Versão reduzida da escala portuguesa de afeto positivo e negativo – PANAS-VRP: Análise fatorial confirmatória e invariância temporal. *Revista PSICOLOGIA*, *28*(1), 53-65, 2014

Appendices

Appendix – Application Manual

This manual is intended to present all the functionalities of the mobile application as well as explain how to navigate through it. The application was conceived mainly to be a tool for physiotherapists, for patient examination and evaluation during therapy sessions. The application has also a patient component, where the same app developed for physiotherapist is presented with some little less features.

To execute the application a network connection is required on the mobile device where the app is running.

Login Interface:

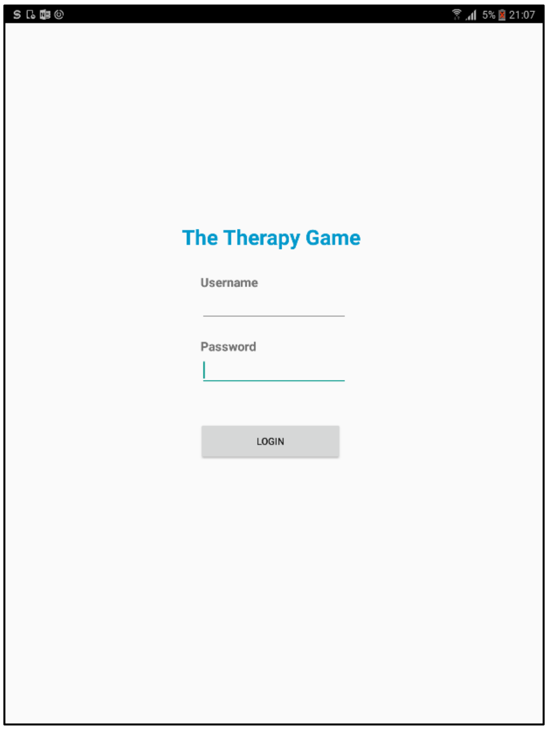

Figure 63.Mobile application login interface

Either the patient or the therapist should provide the username in the first field and the password in the second field. Then the login button should be pressed, if the authentication is succeeded. In the patient's case all their data will be retrieved in order to present it. In therapist case all patients associated are fetched.

User Area Interface:

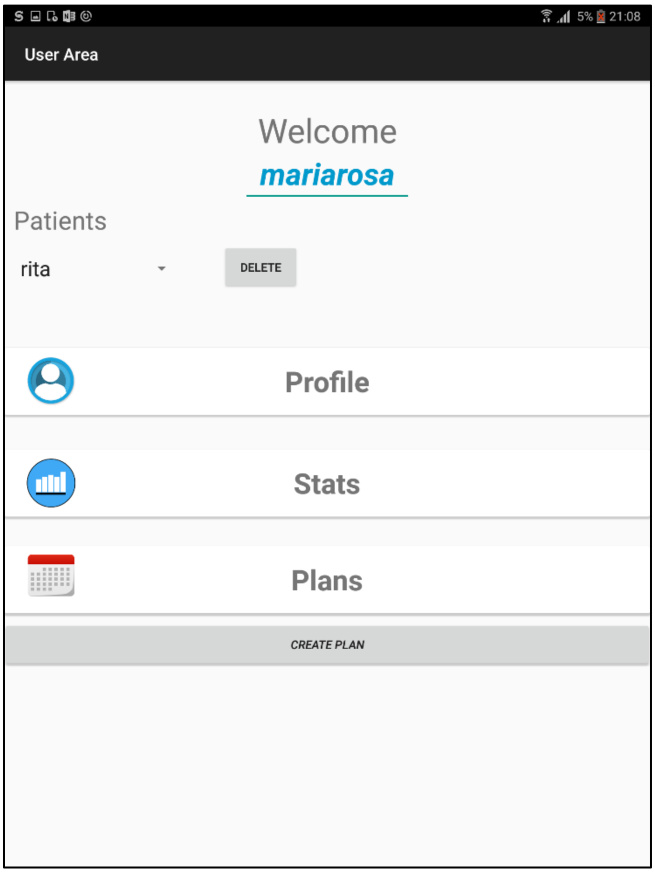

Figure 64.User area interface

The user interface, as said before is different from patient to therapist, the one presented in Figure 64 is regarding the therapist. The therapist can have access to a spinner containing all patients, as well as a button to have the possibility to delete them from the system, if no longer necessary. A *Create plan* button is also presented under the plans tab, to give the therapist the possibility of creating a user training plan. The three card views in the middle lead: the first one to *Profile Interface*, the seconds to the *Stats* and the third to the *Plans*.

Profile Interface:

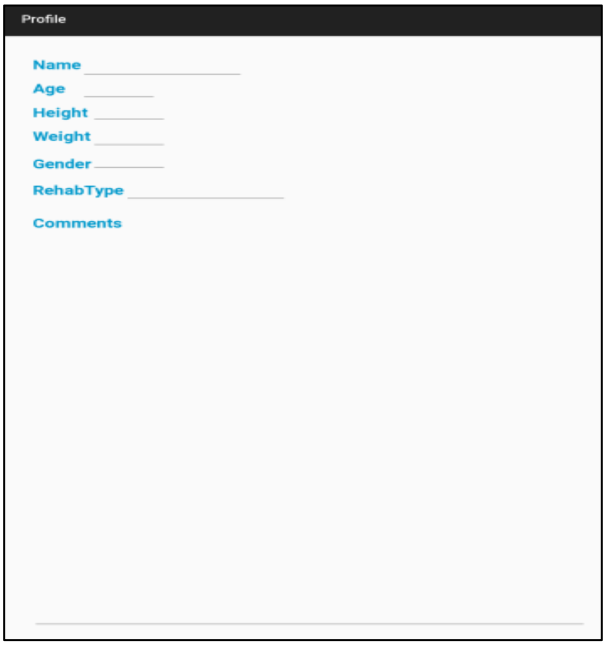

Figure 65.Patient profile interface

In patient *Profile* interface is where patient data is displayed for consulting. Both therapist and patient have access to this interface. The data displayed is related to the data inserted by the therapist while registering the patient in the game application. Personal data is also displayed, as well as some notes related to patients condition when starting sessions. Any type of relevant information regarding the patient or the therapist can also be added.

Plans interface:

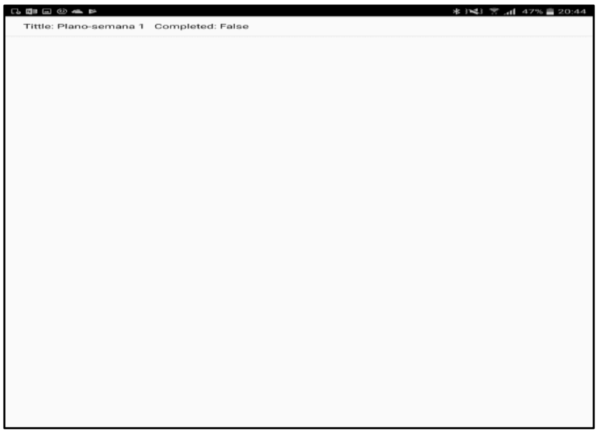

Figure 66.Patient plan list interface

The *Plans* interface is where the plans that therapist created are going to be displayed. In the top of Figure 66, is a training plan. The plan displayer contains the plan name for better distinction, and a field related to the state of the plan, if it's completed or not. In this way the therapist can track if the patient completed their plans, having access to their sessions in clinic or remotely.

Add new plane interface:

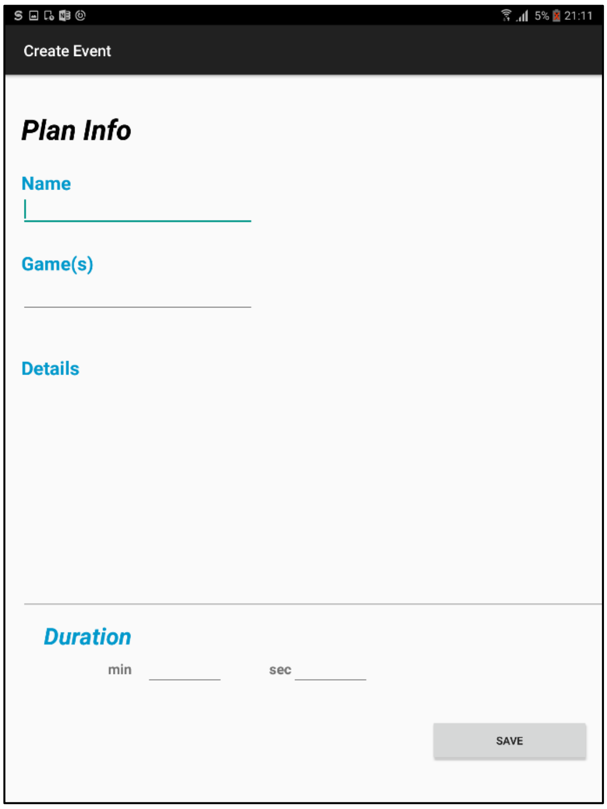

Figure 67.Therapist create plan interface

The *Plan Info* interface is only for therapist and is where he can create a training plan to the patient that was selected in user area spinner. The therapist is able to provide a plan name, the games that patient should train therapy exercises, some details regarding game configuration (how the games should be played, repetitions, game inputs), and finally the session duration. Provided all the inputs the therapist should press the *Save* button, and a message will appear to confirm the plan creation.

Stats Interface:

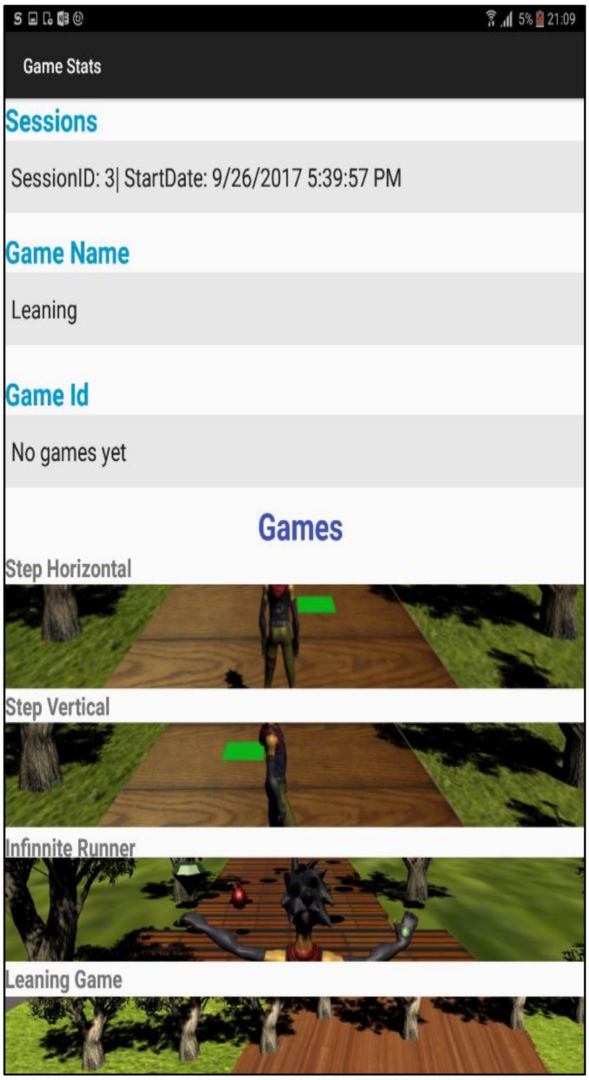

Figure 68.Game stats interface

In *Stats* interface is displayed the number of sessions and games played, for posterior analysis in the respective game interfaces. The therapist can see each associated patient info by selecting the one in the user interface. In this screen the different patient sessions are presented in a spinner list, identified with id, and start date. After selecting the session, the game must be selected in the *Game Name* spinner. Upon selecting the game id spinner will be filled if there were any correspondent games played in that session. If so, it can be selected the game in game id spinner and to click on the respective image card view to view more details.

Game Interfaces

One of the feature present only for therapist both users is the **Graph button** which allows the therapist to consult data, in graph form, collected by the Kinect sensor. It can always be found in the bottom of each game stats screen.

Step Horizontal/Vertical Game Stats and Graph:

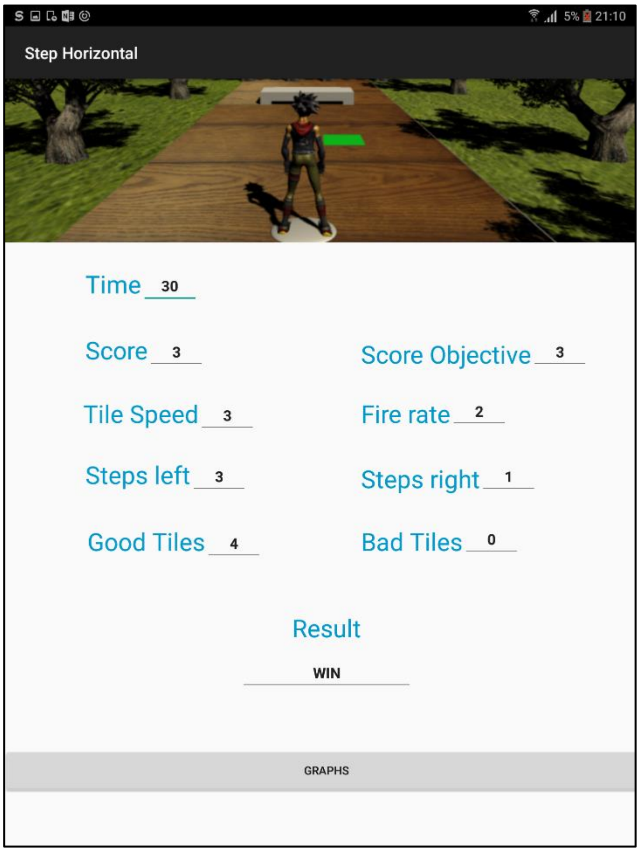

Figure 70.Step horizontal game stats interface

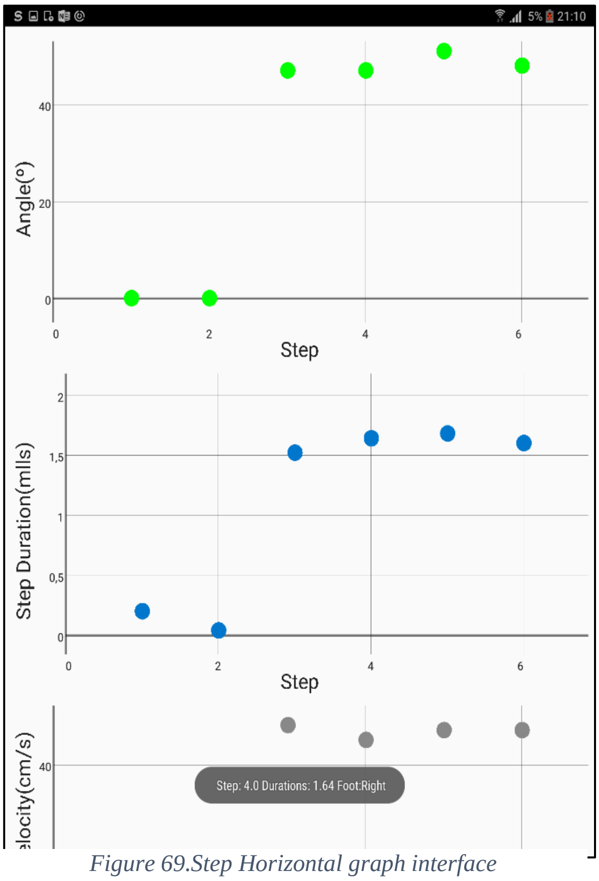

Figure 69.Step Horizontal graph interface

Figure 70 represents the interface that is the same for both *Step Horizontal* and *Step Vertical* games, having the same game logic. *Time*, *Score Objective*, *Tile Speed*, and *Fire Rate* are customization inputs provided before playing the game. *Steps Left*, *Steps Right*, *Good tiles*, *Bad Tiles* and *Result* are inputs filled after playing the game.

By pressing *Graphs Button*, it will take the therapist to the correspondent game graph area.

Figure 69 is represented how the therapist can recur to graphs related to patient's game and analyze their data. By pressing each data point, a *Toast message* as the one in the bottom of Figure 69 will appear giving a more detail view of the data point.

Infinite Runner Game Stats and Graph Interface:

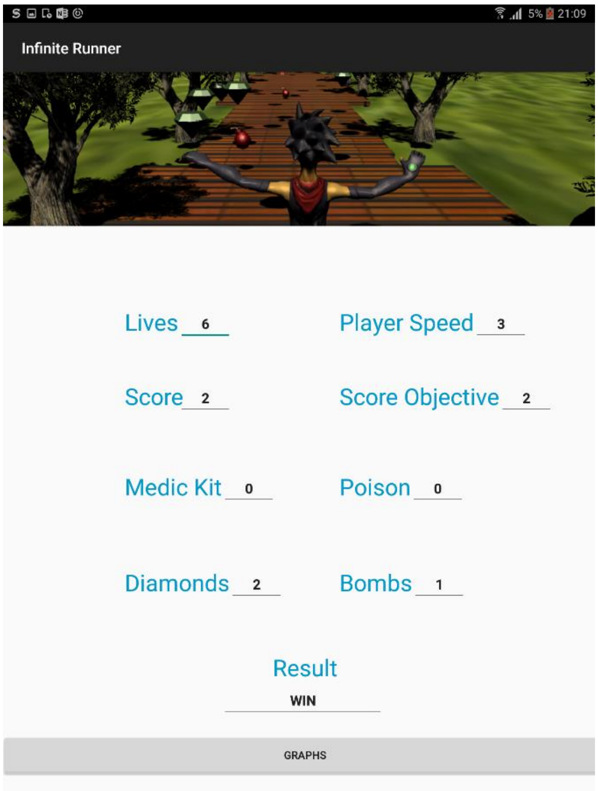

Figure 72. Infinite Runner game stats interface

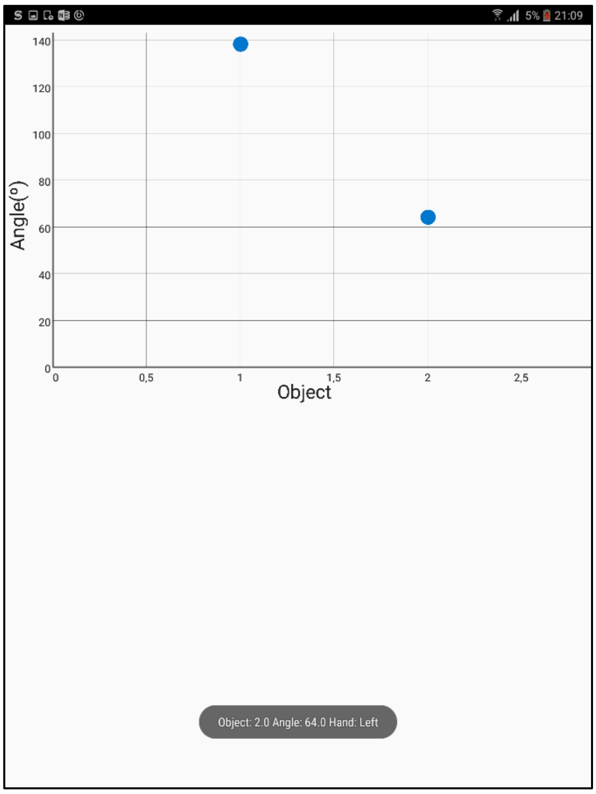

Figure 71. Infinite Runner graph interface

In Figure 72 is represented the *Infinite Runner Game Stats*. *Lives*, *Player Speed*, and *Score Objective* are inputs provided upon game customization. *Score*, *Medic Kit*, *Poison*, *Diamonds*, *Bombs* and *Result* are data collect after game play. The right screen is referring to the *Infinite Runner* graph area where data related to the objects caught is presented.

Leaning Game Stats and Graph Interface:

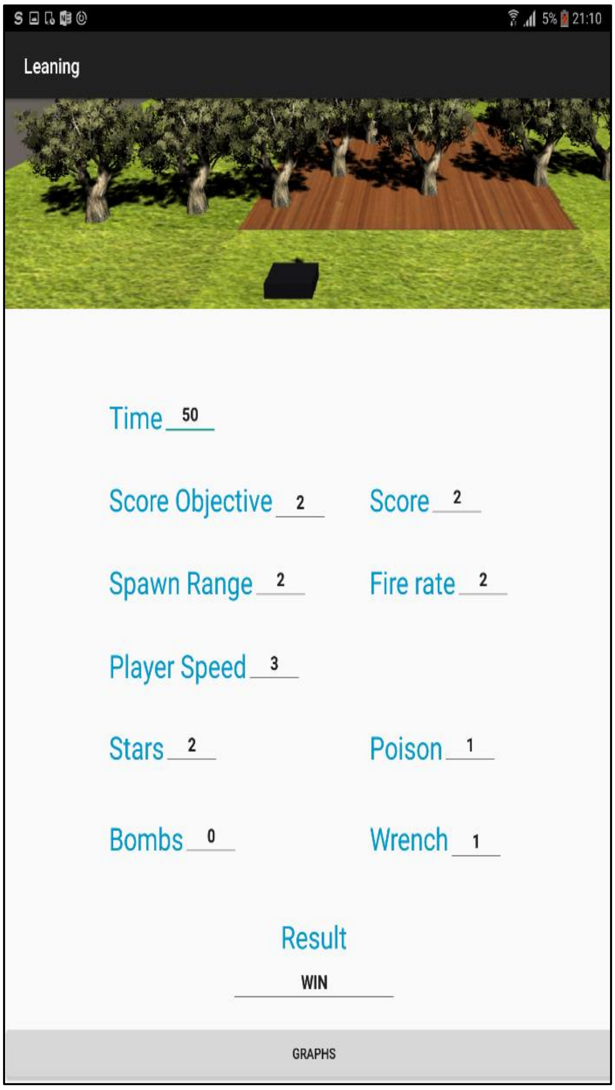

Figure 74. Leaning game stats interface

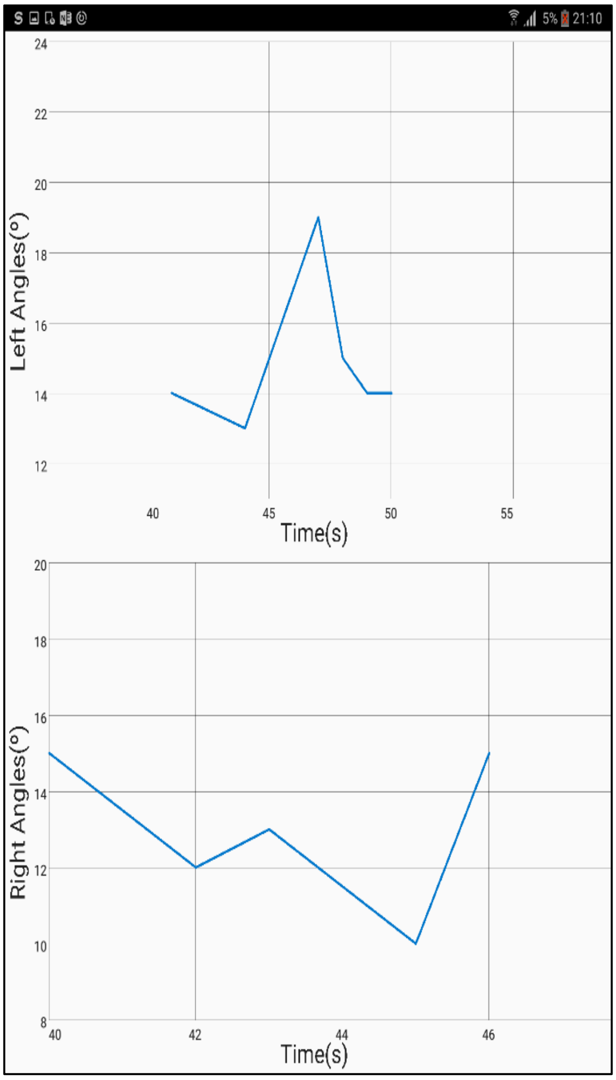

Figure 73. Leaning graph interface

In Figure 74 is represented the *Leaning Game Stats* screen. *Time*, *Score Objective*, *Spawn Range*, *Fire Rate* and *Player Speed* are game inputs provided before game play. *Score*, *Stars*, *Poison*, *Bombs*, *Wrench* and *Result* are data collected after game. On the right figure is the *Leaning Graph* area where data collected by the Kinect sensor is graphically represented - in this case leaning periods during the game.

Hit Wall:

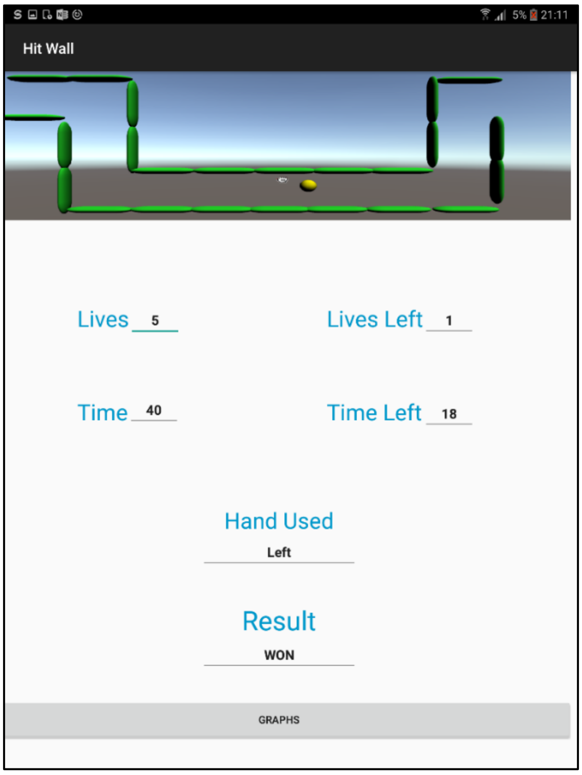

Figure 75.Hit wall graph interface

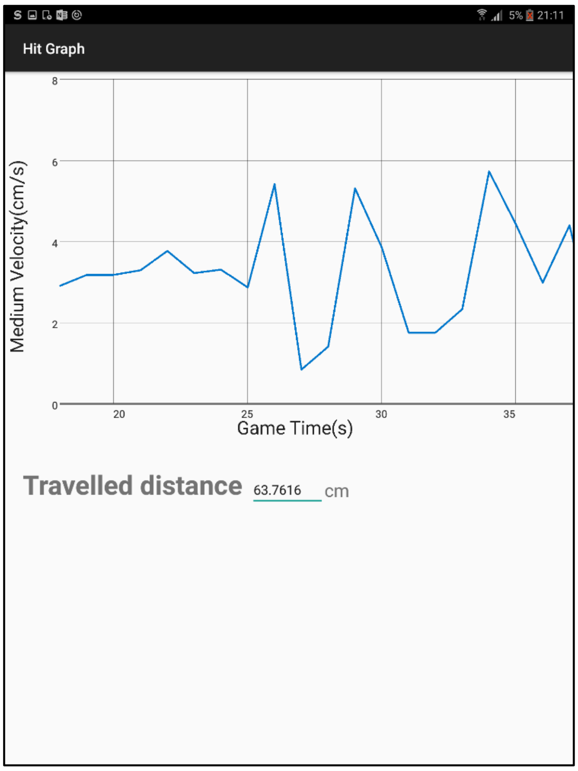

Figure 76.Hit wall game stats interface

In Figure 75 is represent the *Hit Wall Game Stats* interface where patient can see their game stats. *Lives*, *Time* and *Hand Used* are inputs provided before game play. *Lives Left*, *Time Left* and *Result* are variables collected only after play.

In Figure 76 is the graphical representation of Kinect collected data for the *Hit Wall* game as also the travelled distance by the player during game play.

Appendix B – Game Application Manual

The game was built and compiled in an **.exe**. To play the game a network connection is required on the PC where the game is running.

After running the .exe file, the game will start and the login page will appear:

Login Screen:

Figure 77.Unity game Login panel

In this panel the login is possible for three types of users: **Administrator**, **Therapist,** and **Patient**. On the top corner is represented the **list of therapists** registered in the system. In the middle are the login input fields. On the top right is a camera representative window where a QR code can be presented to facilitate the login method. To activate the camera press **ReadQR Button**.

If there is no therapist registered in the game, someone with higher command (as a leading therapist), should login with the admin credentials (username: admin; password: admin. (Figure 77).

If a therapist logs in it will appear the therapist login panel represented in Figure 80. In the patient case it's the same as the one illustrated in Figure 80, but without the patients list. Every time a patient logs in, a session is created, so every time a game is played, it will become associated to that same session. The panel also allows the registration of patients.

If a therapist is already registered, when starting the game, the **therapist spinner** on the top left corner is filled with the name of the current registered therapists. The therapist should select his name before pressing the **Register/Create Profile** button in order to associate the patient upon registration (Figure 80). **Quit** button exits the game application.

Admin Menu panel:

Figure 78. Admin menu panel

In the screen, preceding the *Login* panel is where the administrator can register therapists in the system by pressing the **Register Therapist** button, taking the admin to the register therapist panel represented in Figure 78.

This view corresponds to the therapist registration panel, where therapists can be registered in the system by the admin. After providing the required input that is required for the admin to press the **Generate QRcode** button, a QRcode is generated and saved a, and if needed, print it, so the user can have the possibility to login with the QRcode.

Register Therapist Panel:

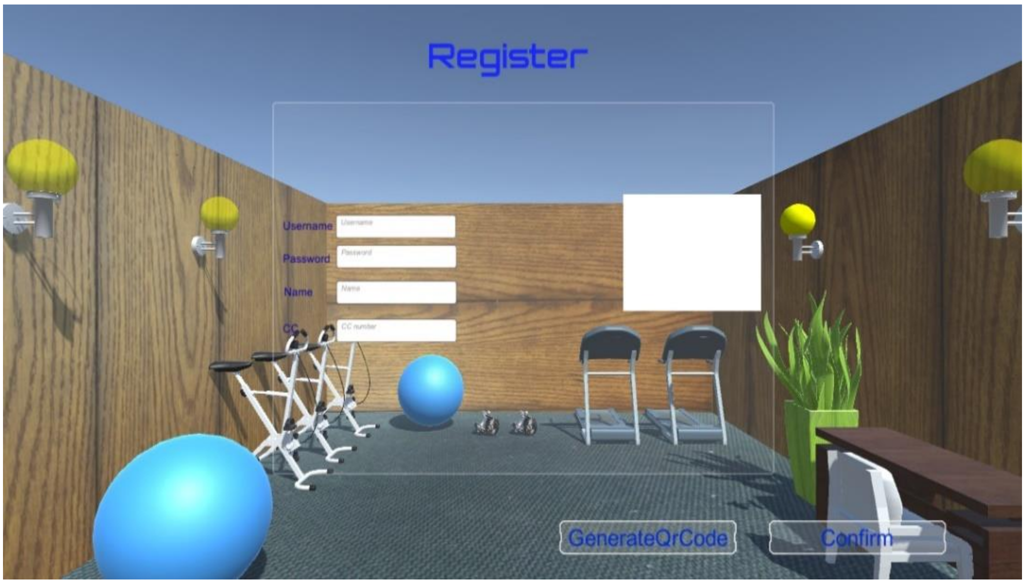

Figure 79.Register therapist panel

Register Patient Panel:

Figure 80. Register patient panel

Therapist Menu Panel:

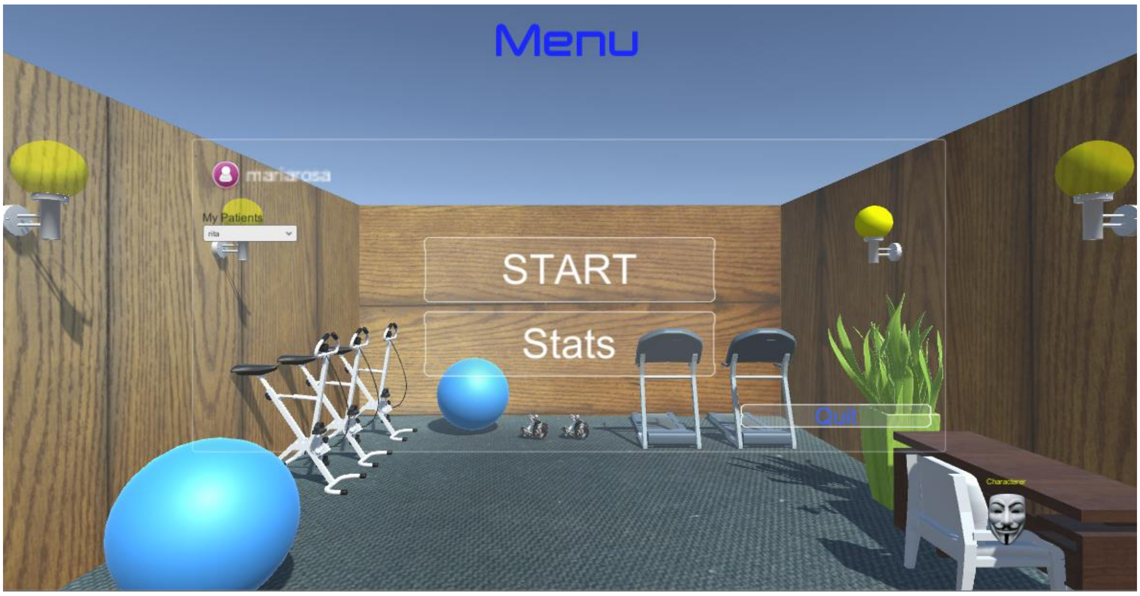

Figure 81.Therapist Menu panel

By pressing **the Register/Create Profile Button** in the login panel the therapist should provide all the input necessary for registration, including the **rehab type toggle field** where it decides if the patient should have access to only **inferior**, **superior** or **both** game types.

The QRcode functionally is also applied to patient user.

Figure 81 is represented a panel when a therapist logs in. In the top right corner is placed an **indicator of the username** currently logged, rightly below is the **list of patient** associated with the correspondent logged therapist. **Stats button** gives access to the selected patient game stats.

Stats panel:

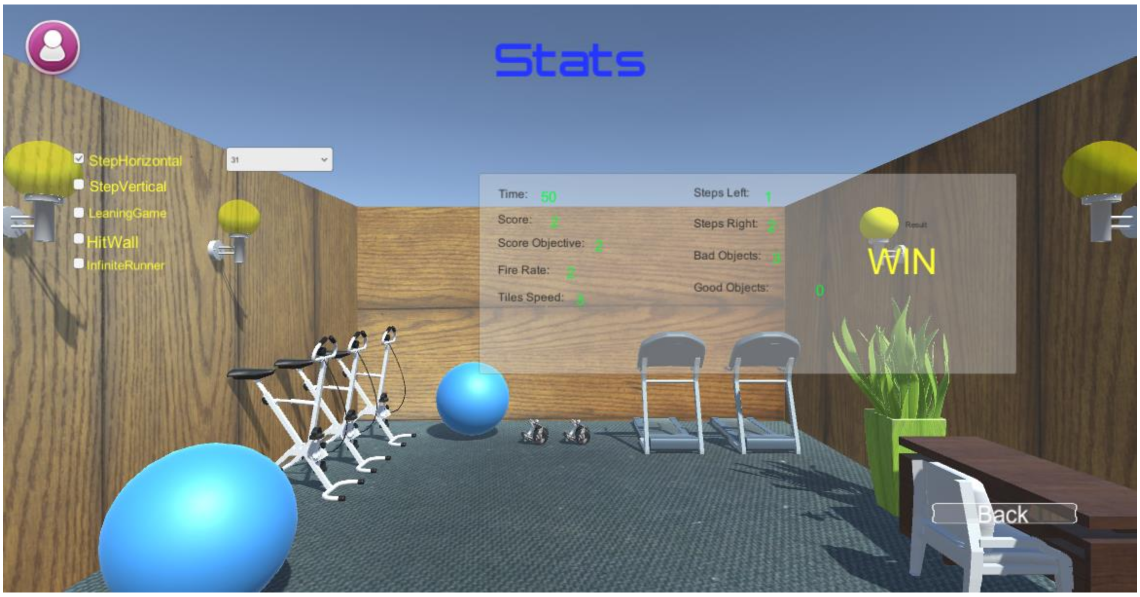

Figure 82.Game stats panel

In Figure 82 is represented the *Stats* panel where both patient and therapist can have access to the different game stats. On the left side of the panel there are five game representative toggles, that when clicked, will fetch all the games associated to the patient in question and populate the spinner with the respective game ids.

Game Focus Panel:

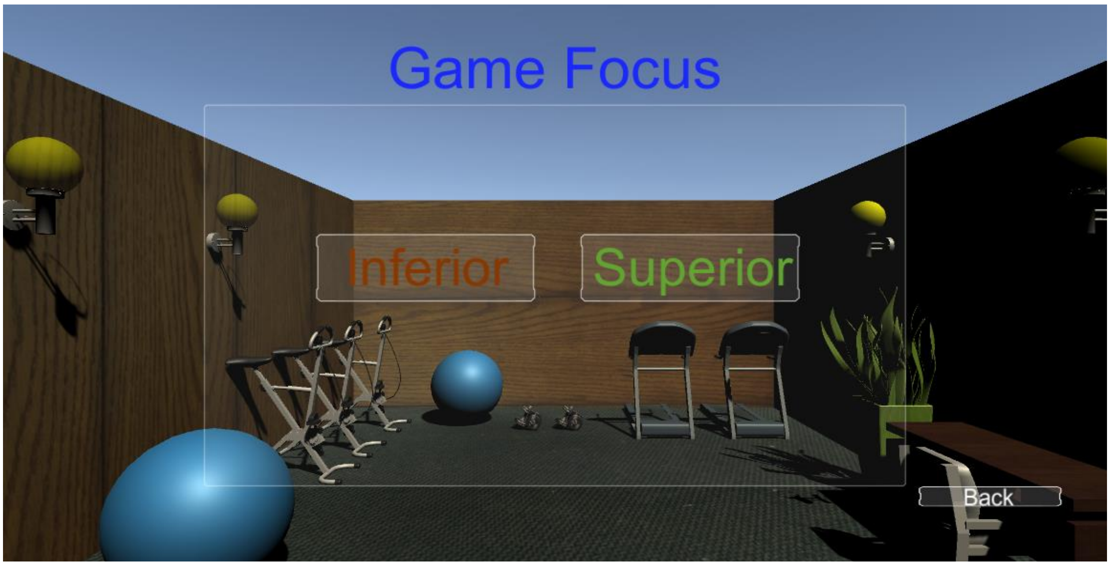

Figure 83. Game focus panel

By selecting a game id, the correspondent game stats will be display on the right panel. The panel represented in Figure 83, are displayed the **buttons relative to the patient rehabilitation type,** upon registration. There are two buttons associated with each rehabilitation type, for superior or inferior limbs. If only inferior or superior was selected on registration, only one button would be activated.

Inferior Member Games panel:

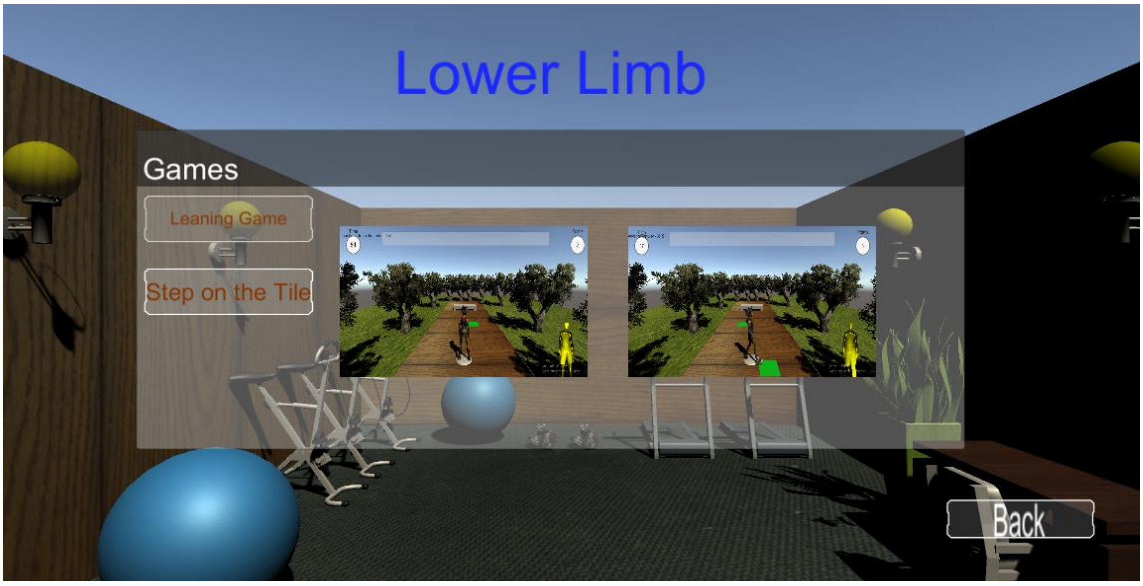

Figure 84. Lower limb game panel

In the Figure 84 is represented the panel where the lower limb games are presented. By passing the mouse over each game button, an image relative to the game will appear.

Superior Members Games Panel:

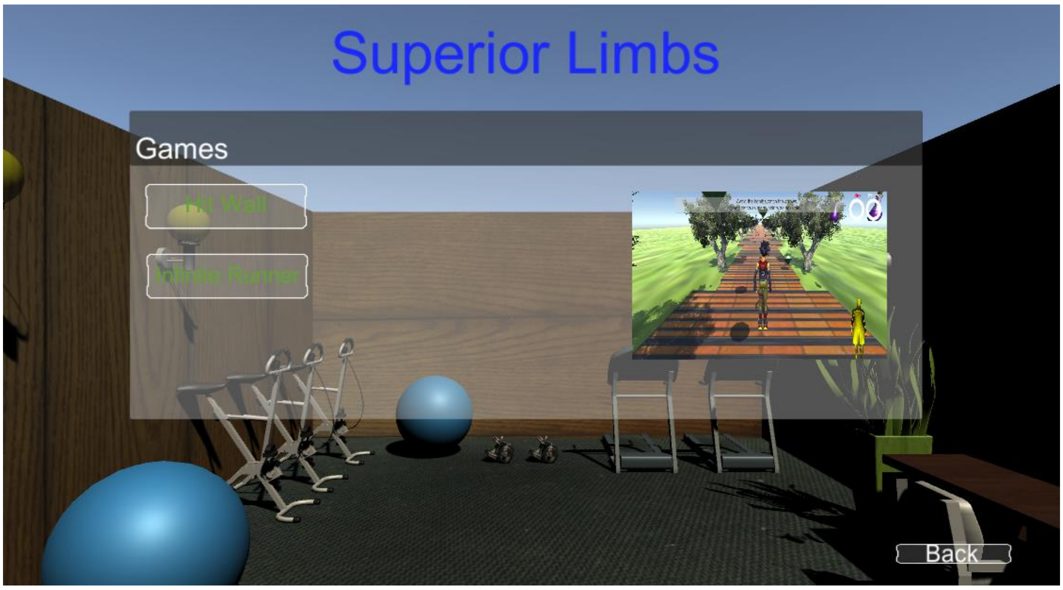

Figure 85. Superior Limb game panel

Figure 85 represents the games for superior limbs where each button leads to the customization panel of the corresponding game.

Game customization panels and scenes

All the customization scenes presented next, have some parameters in common, like the **game information** where the **Objective**, **How to Play** and **Training** parts are displayed. The **Back** and **Play** button are also common features and operate as their name states.

Step the Tile Customization panel:

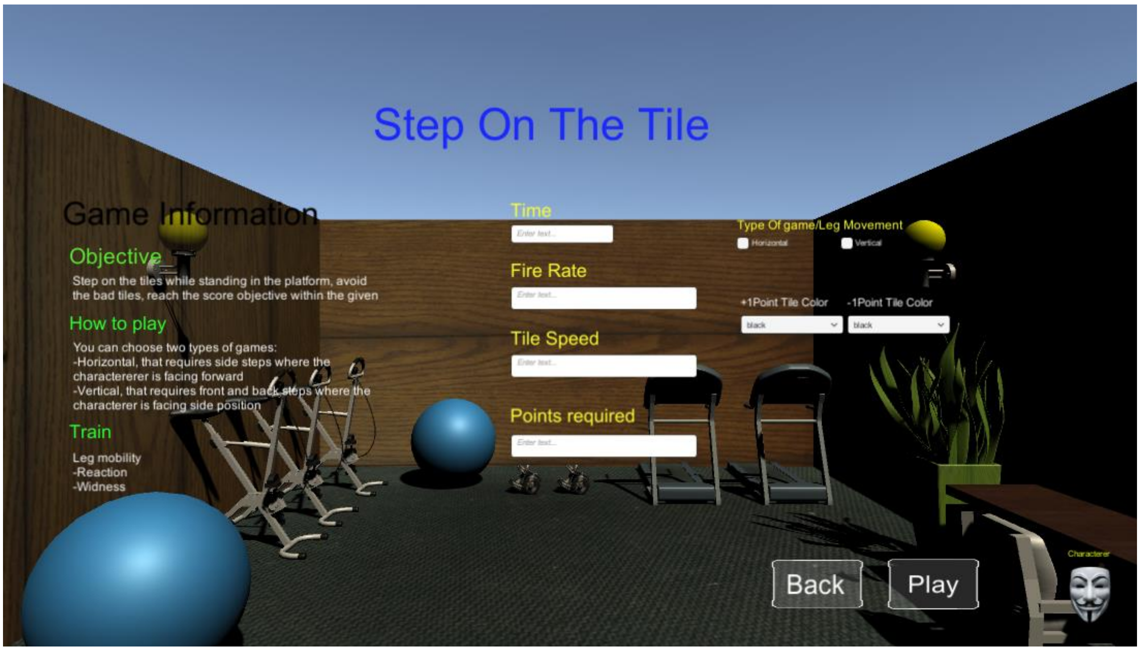

Figure 86. Step the Tile customization panel

The Figure 86 represents the panel where *Step the Tile* game can be configured, by providing all the required inputs and pressing the **Play** button. Inputs **game time**, **fire rate** represents the interval of spawn of each tile, **tile speed** indicates the speed of the tile spawn when moving towards the player and the **points required** to win the game. There are also other features presented, namely **toggles representing the type of game** to be played. If toggle **Horizontal** is enabled, the game will basically consist in side step movements (Figure 87). If **Vertical** toggle is enabled the game will consist on front and back steps (Figure 88).

There is also a possibility to change color of the two different tiles. This is important in neurorehabilitation when cognitive processes are evaluated. There are a set of seven **different colors** that is displayed on the **spinners** that can be seen in Figure 86.

Step Horizontal Game Scene:

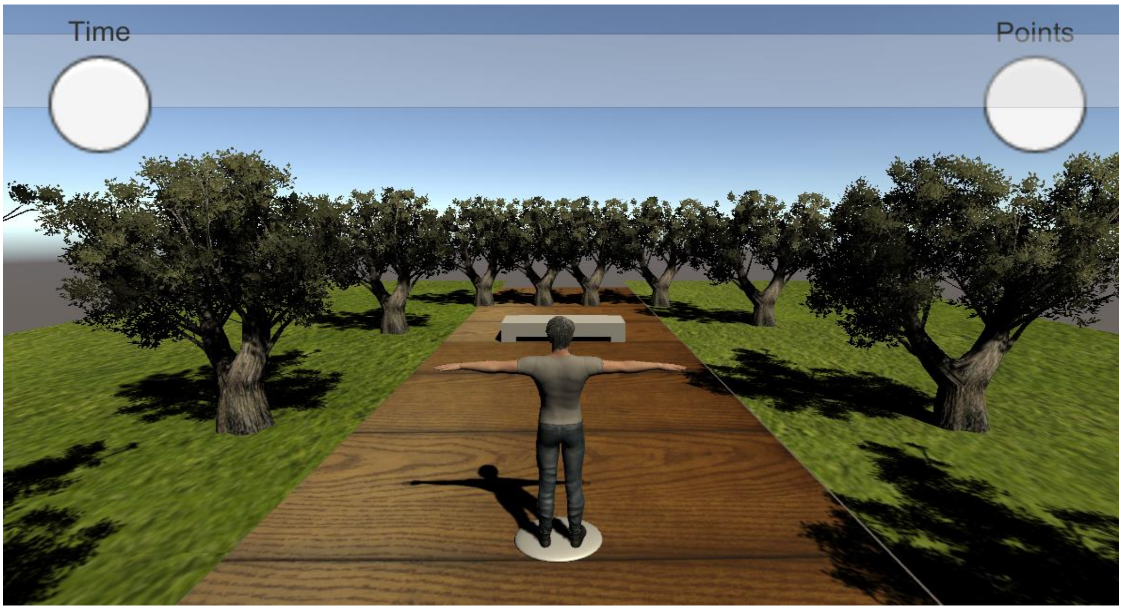

Figure 87.Step the tile Horizontal game scene

The scene from Figure 87 represents the step horizontal game where the controlled avatar lays facing the tile object spawner and should do the sides steps. To **win** the game, score objective must be reached. The player **loses** if the time runs out.

Step Vertical Game Scene:

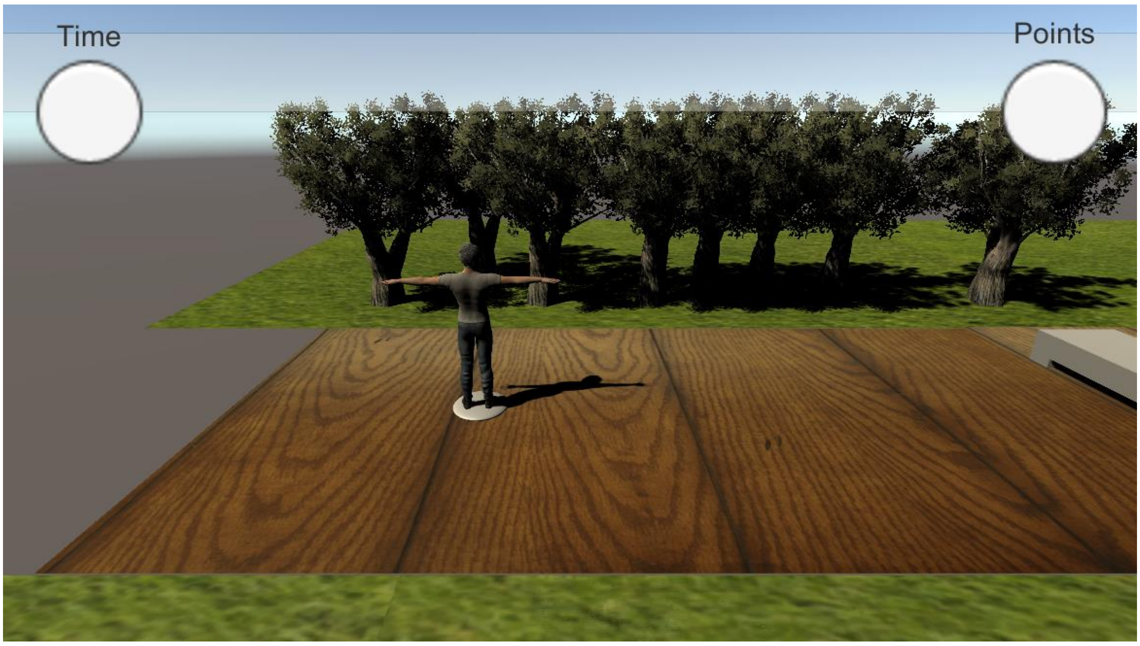

Figure 88. Step the tile Vertical game scene

The scene from Figure 88 represents the step vertical game where the controlled avatar has the object spawner on the right side, so he can effectuate front and back steps.

Leaning Customization panel:

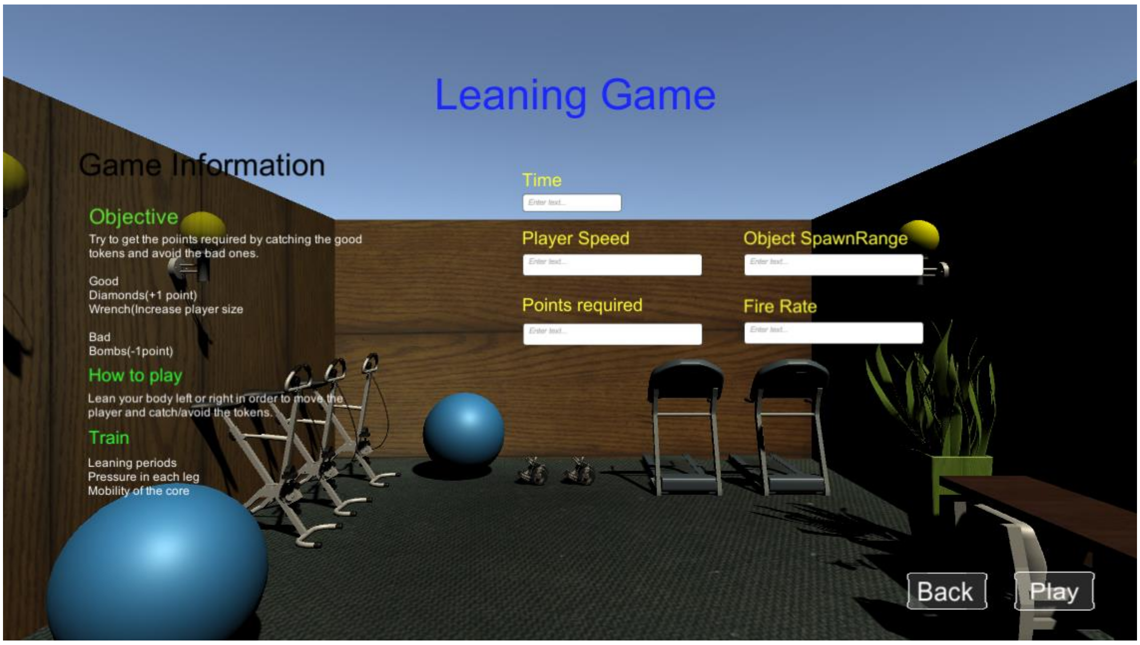

Figure 89. Leaning customization panel

The Figure 89 represents the *Leaning* game customization panel where the inputs required for play are: **Time**, **Player speed** as the speed of the player object that is represented by a parallelepiped (greater the value, more difficult will be to control the player and catch the objects), the **object spawn range** (higher the value, larger the area where the objects can spawn), **fire rate** representing the time between spawned objects and points required to win the game.

In the scene represented in Figure 90 is represent the leaning game after the **Play** button is pressed. To **win** the game score objective must be reached. **Losing** implies ending the game time without reaching the pretended scores.

Leaning game scene:

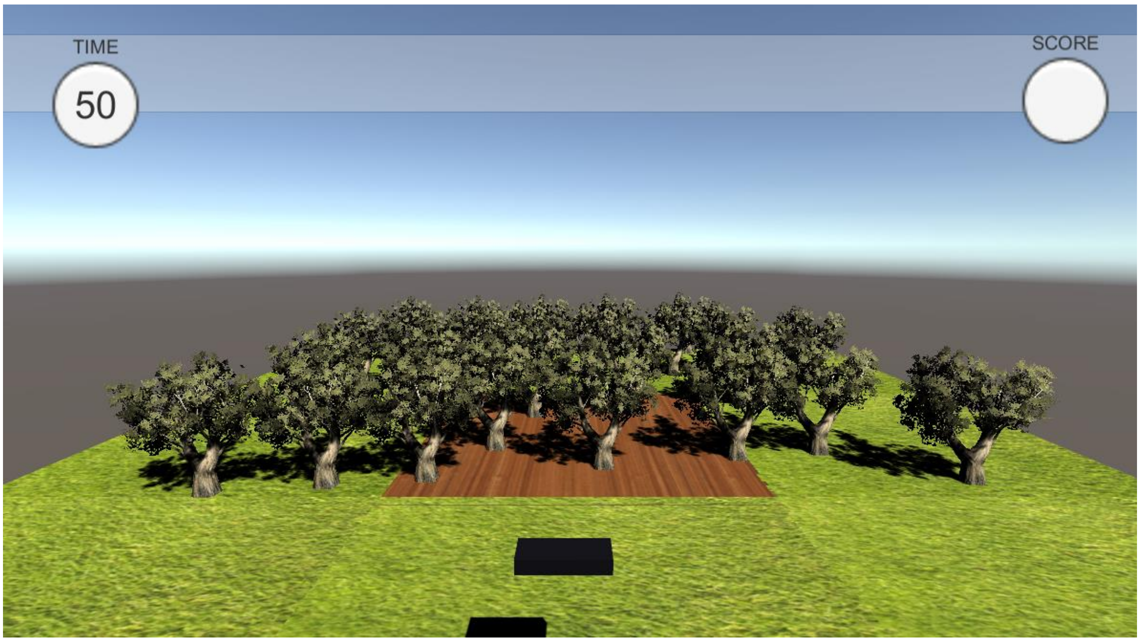

Figure 90. Leaning game scene

Infinite Runner Customization Panel:

Figure 91. Infinite Runner customization panel

In Figure 91 is represented the *Infinite Runner* customization panel where some parameters are asked as input, such as **Lives** (player lives in the game), **Score objective** (Score required to win the game), **player speed** (higher the value, greater the avatar speed moving in the scene, greater the difficulty to catch objects). On the right side of the screen are also required some object inputs. For **Bombs** (take 1 point) and **Medic Packs** (add 1 life) they will appear only on the game floor, so to avoid or catch those objects the patient should lean to each side and the avatar does the same. The **Diamonds** (add 1 point) and **Poison** (take 1 life), will spawn in 3 different areas, top, left and right sides of the map being the player (avatar) in the center.

Figure 92 is an example scene of *Infinite Runner* game play. To **win** the game, reaching the score objective is required. **Losing** the game implies losing all lives or reaching the end of the map without grasp the score objective.

Infinite Runner Game Scene:

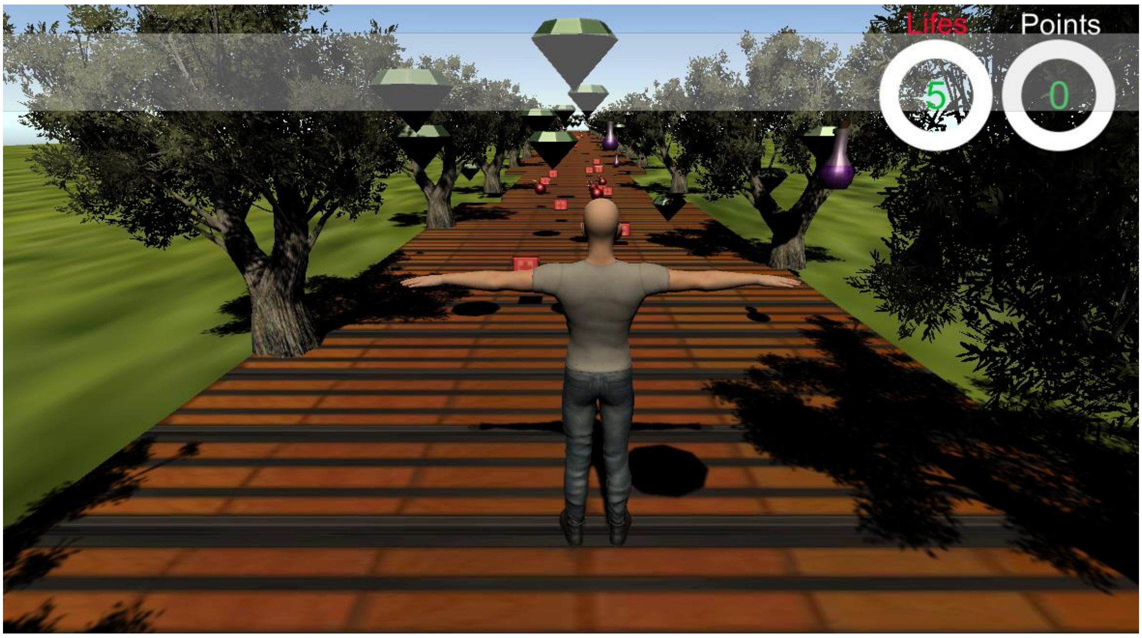

Figure 92.Infinite Runner game scene

Hit Wall Customization Panel:

Figure 93. Hit wall customization panel

In the scene represented in Figure 93 the *Hit Wall* customization panel is shown. First the therapist must select the game the patient should play, there are four already built levels that can be selected on the **Select Level** spinner. There is the **Select Custom Level** spinner where the custom levels created by the therapist, in map creator panel (Figure 94), are displayed. After selecting the pretended level this will appear on the right white image so the therapist can know witch level is the patient going to play. After selecting the level, **type of map toggle** should be selected, **Maps toggle** for default maps and **Custom maps** if select map is custom. Then **Time** and **Lives** input should be provided as well as the **hand used** to play the game. **Create Level button** is only a therapist feature, it is not shown if a patient is logged in.

Hit Wall Game Scene:

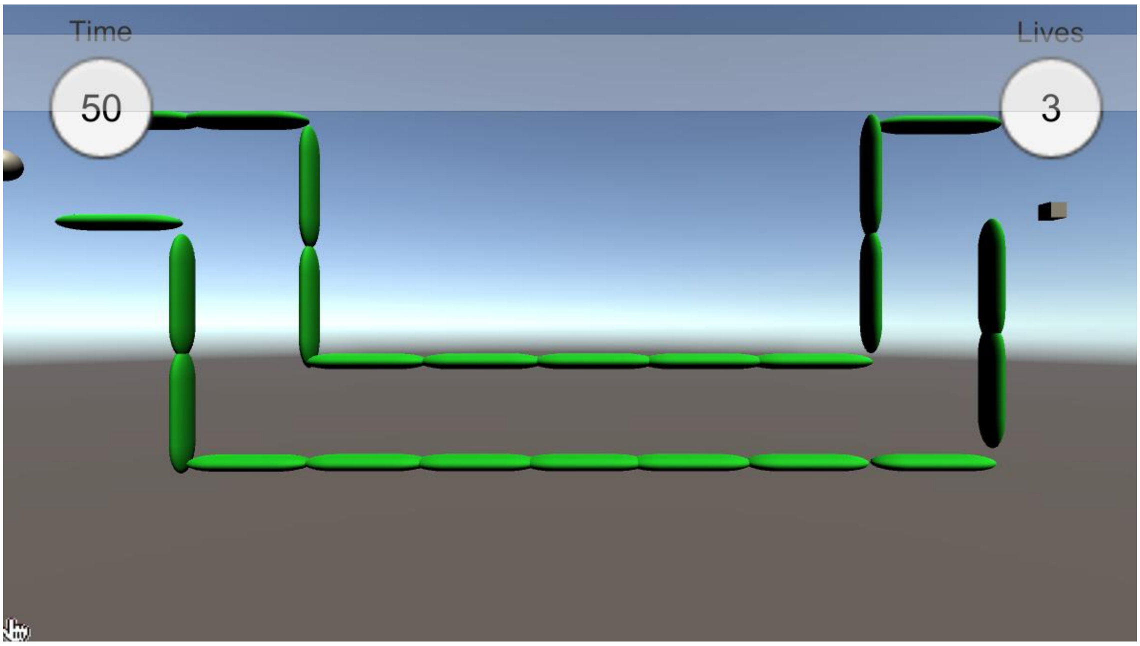

Figure 94.Hit wall game scene

The Hit Wall game will only start counting if the Kinect detects the player. The game can be configured to have less complicated scene, and what arm should be used to play the game. Before standing in front of the Kinect the user can **drag the player object** (sphere) and **finish point object** (cube) to the opposite sides. The walls can be placed wherever the therapist wants to facilitate the patients game play. After all the objects are in place the patient can **start the game** by grabbing the sphere with the hand that has an indicator represented on the bottom left screen, tracked by the Kinect sensor until reach the finish point. The game is **won** when the player touches the finish point, and **lost** if the player loses all his lives or the times runs out. Losing lives implies touching the walls, losing one life every time is that happens.

Create Map panel:

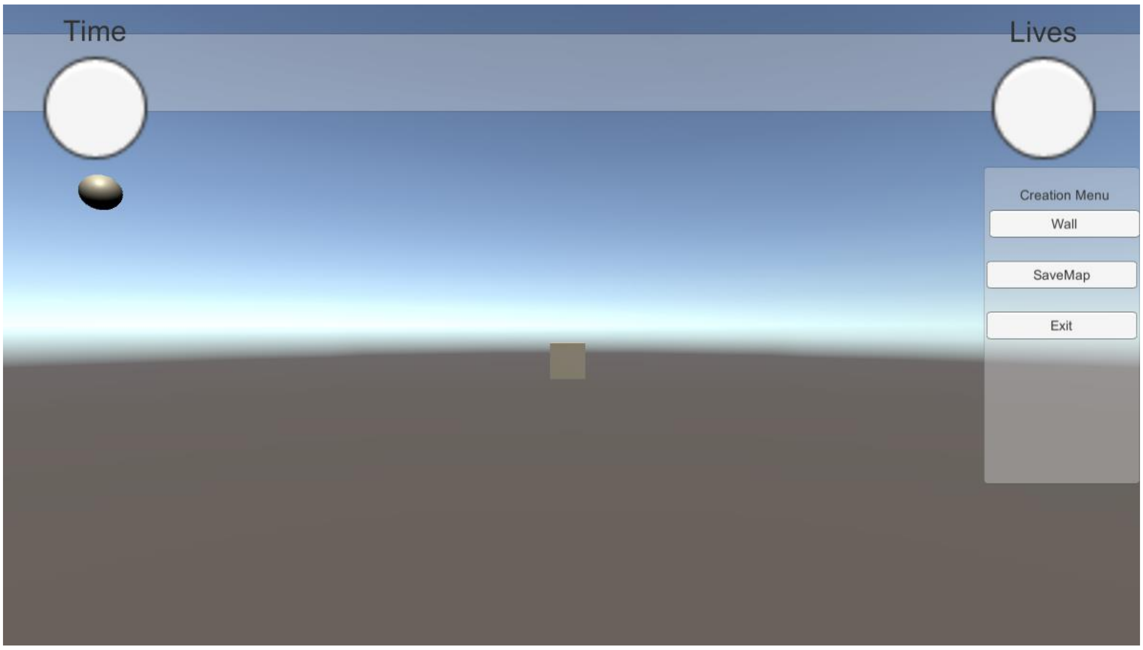

Figure 95.Create hit map scene

Figure 95 displays the scene where therapists can create custom maps for *Hit Wall* game, that player and finish point objects are ~~just~~ there to have a space notion while creating the map. On the right side of the screen is represent the **Creation Menu** where the patient can press the **Wall button** to generate a *Hit Wall* on the screen and place it wherever he wants by **dragging it with the mouse and turning it by using the left and right arrow keys while holding the wall with the mouse**. After the map is finished, the **Save Map Button** should be pressed so the map can be saved as well as a picture representing the same. **Exit** button, is used to quit the scene.

Other Scenes

There are three possible characters for patients choosing, represented in Figure 96, Figure 97 and Figure 98.

Figure 96. Male character

Figure 96. Female character

Figure 98. Other male character

These characters are used to play games where an avatar is required, like the *Infinite Runner* and *Step the Tile* games.

Character Selection Menu:

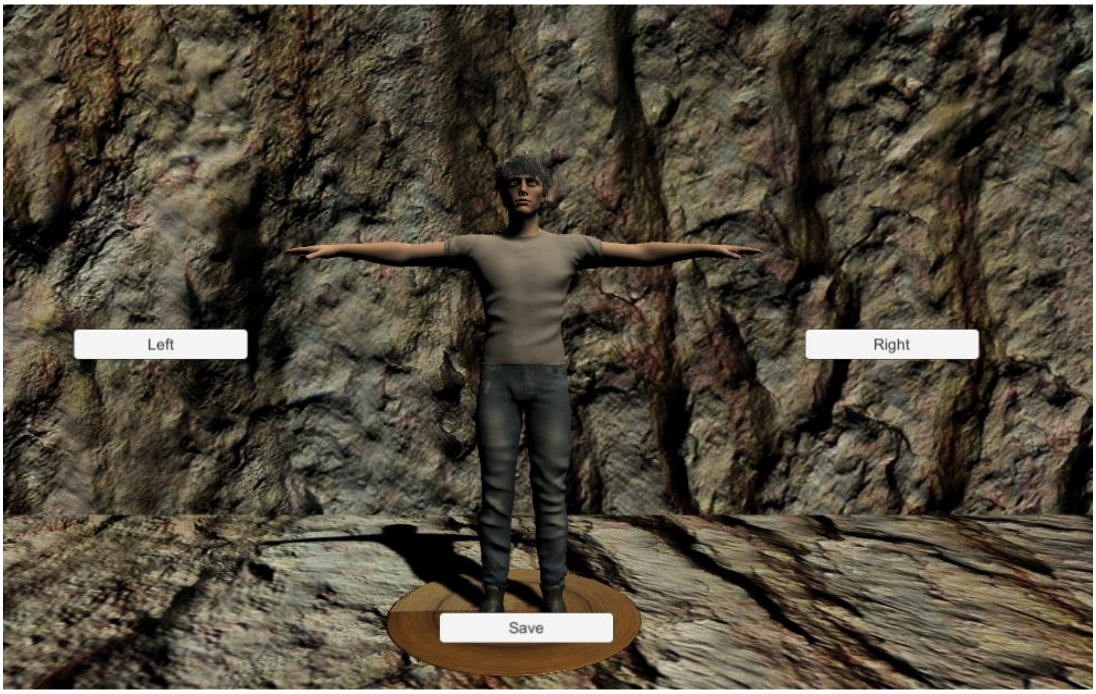

Figure 97. Chose character scene

Figure 99 is the scene where the therapist or patient can choose the character that he/she like to play the games. To go through the different characters **Left** or **Right** buttons should be pressed. **Save** button will save the actual character and exit the scene.

Other game components:

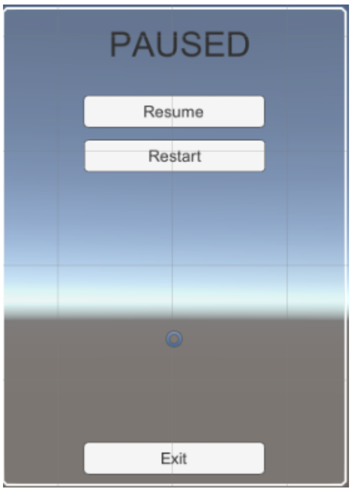

Figure 100. Pause menu

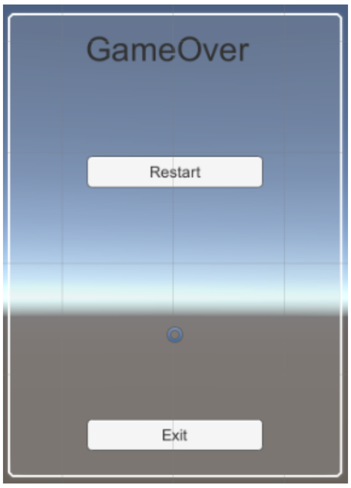

Figure 98. Game Over menu panel

Figure 102.. Win menu panel

In the figures above are displayed three sets of game panels, namely the **PAUSE**, **Game Over** and **Win** Panel. Pause panel appears if **ESC** key is pressed, **Game Over** panel if game is lost, and **Win** panel if game is won. The buttons **Exit** take the user to *Game Focus Scene*, the **Restart** button takes the user to the game customization panel of the game where the panel is activated and the **Resume** button is to resume the paused game.

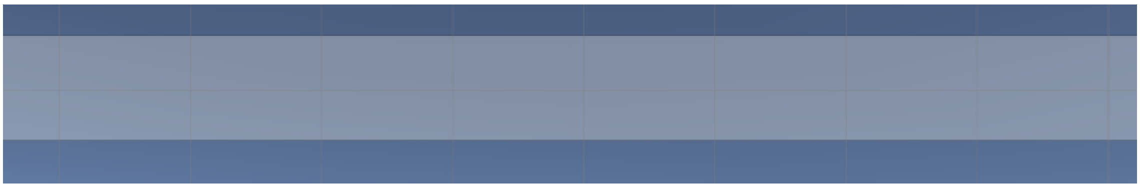

Figure 99.Pop messages display panel

Figure 103 represent the **message board** where predefined messages will appear encouraging the player if he is succeeding in playing the game well, if not, other message types will appear.

APPENDIX C – Gesture listener

```
public class SimpleGestureListener : MonoBehaviour, KinectGestures.GestureListener-
Interface
{
   [Tooltip("Index of the player, tracked by this compo-
nent. 0 means the 1st player, 1 - the 2nd one, 2 - the 3rd one, etc.")]
   public int playerIndex = 0;

   [Tooltip("GUI-Text to display gesture-listener messages and gesture information.")]
   public GUIText gestureInfo;

   // singleton instance of the class
   private static SimpleGestureListener instance = null;

   //if the gesture was detected or not
   private bool leanLeft;
   private bool leanRight;

   // private bool to track if progress message has been displayed
   private bool progressDisplayed;
   private float progressGestureTime;



public static SimpleGestureListener Instance {
      get{
         return instance;
      }}


   public void UserDetected(long userId, int userIndex){
      if (userIndex != playerIndex)
         return;

      // as an example - detect these user specific gestures
      KinectManager manager = KinectManager.Instance;

      manager.DetectGesture(userId, KinectGestures.Gestures.LeanLeft);
      manager.DetectGesture(userId, KinectGestures.Gestures.LeanRight);



      if(gestureInfo != null)
      {
         gestureInfo.text = " Lean for each side to catch the balls";
      }
   }
```

```
    public void UserLost(long userId, int userIndex)
    {
        if (userIndex != playerIndex)
            return;

        if(gestureInfo != null)
        {
            gestureInfo.text = string.Empty;
        }
    }


    public void GestureInProgress(long userId, int userIndex, KinectGestures.Ges-
tures gesture,
                                  float progress, KinectInterop.JointType joint, Vector3 screenPos)
    {
        if (userIndex != playerIndex)
            return;

        if ((gesture == KinectGestures.Gestures.LeanLeft && progress > 0.5f)) {

            leanLeft = true;

            if (gestureInfo != null) {
                string sGestureText = string.Format ("{0} - {1:F0} degres", gesture, screen-
Pos.z);
                gestureInfo.text = sGestureText;

                progressDisplayed = true;
                progressGestureTime = Time.realtimeSinceStartup;
            }
        }

        else {

            leanLeft = false;
        }

        if ((gesture == KinectGestures.Gestures.LeanRight && progress > 0.5f)) {

            leanRight = true;

            if (gestureInfo != null) {
                string sGestureText = string.Format ("{0} - {1:F0} degres", gesture, screen-
Pos.z);
                gestureInfo.text = sGestureText;

                progressDisplayed = true;
```

```
                progressGestureTime = Time.realtimeSinceStartup;

            }
        }

        else {

            leanRight = false;
        }
    }

    public bool GestureCompleted(long userId, int userIndex, KinectGestures.Ges-
tures gesture,
                                  KinectInterop.JointType joint, Vector3 screenPos)
    {
        if (userIndex != playerIndex)
            return false;

        if(progressDisplayed)
            return true;

        string sGestureText = gesture + " detected";
        if(gestureInfo != null)
        {
            gestureInfo.text = sGestureText;
        }

        return true;
    }


    public bool GestureCancelled(long userId, int userIndex, KinectGestures.Ges-
tures gesture, KinectInterop.JointType joint){

        if (gesture == KinectGestures.Gestures.LeanLeft) {
            leanLeft = false;
        }

        if (userIndex != playerIndex)
            return false;

        if(progressDisplayed)
        {
            progressDisplayed = false;

            if(gestureInfo != null)
            {
                gestureInfo.text = String.Empty;
            }
        }
```

```
        return true;
    }


    void Awake()
    {
        instance = this;
    }


    public void Update()
    {
        if(progressDisplayed && ((Time.realtimeSinceStartup - progressGes-
tureTime) > 2f))
        {
            progressDisplayed = false;

            if(gestureInfo != null)
            {
                gestureInfo.text = String.Empty;
            }

            Debug.Log("Forced progress to end.");
        }
    }


    public bool isLeanleft(){
        return leanLeft;
    }

    public bool isLeanRight(){
        return leanRight;
    }

}
```