Final Project Report

Hell Of Balance

Balance Assessment using Microsoft's Kinect

Angelos Oikonomakis	s161216
Riccardo Cannistrà	s161155
Nikolaos Zafeiridis	s161885
Rasmus Christiansen	s123344

Technical University of Denmark
Department of Applied Mathematics and Computer Science
02515 - Healthcare Technology
December 10, 2017

Contents

1 Introduction

The current document represents the group's official report for the final project carried out within the Healthcare Technology course ran at Technical University of Denmark, A.Y. 2017/18. As part of the project, a research has been carried out within a specific healthcare context to identify a background to work on.

The aim of the project and the report is to present, describe, analyze and discuss how can a device such as Microsoft Kinect V2 assess rehabilitation of patients affected by some known health issues. Narrowing the scope, the group decided to address patients affected by balance impairment. For this purpose, a preliminary research has been conducted to better understand the context, to find out how this project could fit in it and what are the problems that the proposed solution addresses and aims to solve.

On top of the research, a concrete application has been developed using the Unity game-engine, that integrates Kinect's functionalities in a game-oriented application to assess balance improvement and rehabilitation. The main idea of the game is to provide a concrete aid to patients: by practicing several times per week, the game would represent an efficient, convenient and motivational way to help them improve their stability. This would involve ad-hoc training with specific exercises focused in spine and core strengthening.

Goal-oriented, task-specific training has been shown to improve balance skills; however, it can be difficult to maintain patient interest. A lack of interest or a short attention span also can impair the potential effectiveness of therapeutic exercises. Conversely, the use of rewarding activities has been shown to improve people's motivation to practice. The goal of the gamified approach is to introduce a rehabilitation protocol to enhance patients involvement, increase self-esteem by providing real-time feedback and scaled assessment to be evaluated on a periodic basis.

The current report is organized and structured as follows. Section 2 provides a detailed overview of what is the problem that the project targets, what is the context and the health-care background, and additionally includes references to similar previous work that has been inspirational to the group. Section 3 refers to what are the requirements of the target users of the application. Section 6.1 documents how it is verified that the system the group implemented actually works. Section 4 provides an in-depth analysis of the game logic, along with a description of the game creation and development process, describing all non-trivial choices. Section 5 summarizes the key-points related to the technical implementation and an overview of the non-trivial design choices. Section 6.2 documents how the application has been tested with real users and what has the outcome been. Finally, sections 7 and 8 evaluate whether the goals of the project have been achieved, discuss eventual issues encountered, and summarize the overall experience.

2 Problem Analysis

This section gives a thorough understanding of the problem that is being addressed, meaning the balance impairment, by providing extended information concerning the reasons of choosing this approach, the health background of the problem itself and finally the implementations already conducted in the past relative to this study.

2.1 Motivation

The motivation behind the current project lies on the fact that rehabilitation exercises are essential in progressing the recovery of people with impaired motor control skills. That is clearly stated in the literature findings that we based this project on.

Many unforeseen factors can lead to the degradation of our balance system, such as a decrease in processing efficient sensory information with age [6] and disabling neurological and musculoskeletal conditions. Events of falling or close falls can reduce self- efficacy; in turn this may cause functional limitations [5].

Basic human functions as standing and walking demand balance skills that involve many essential sensory and motor processes. Feed-forward cognitive controls are responsible for the initiation of preparatory movements in order to maintain balance for expected disturbances. Feedback information is responsible for handling unexpected disturbances or when the preparatory movements fail. Improvement to the reaction of balance disturbances can only be addressed with practice and experience [9].

There is notable evidence from studies that targeted physical training reduces disability and increased training produces better outcomes [10, 13]. Various theories have been advanced to explain behavioral recovery after central and peripheral nervous system disorders [4].

In addition, many VR environments have been developed to include video games. For example, Broeren et al [3] made use of a game in a virtual environment, along with a force-feedback haptic device, to improve control of a stroke subject's left hemiparetic arm. Their results showed that the subject was motivated to practice and exhibited improved dexterity, grip force, and motor control. The positive effects are particularly observed when a game is incorporated into the immersed virtual environment.

2.2 Health Background

Balance impairment, mobility restriction, and falling are serious problems facing older adults and many people with neurological disorders (eg, stroke, traumatic head injuries, incomplete spinal cord injuries, Parkinson's disease, multiple sclerosis, diabetic peripheral neuropathy)[7, 2, 8]. For many of those people, even small disturbances result in a fall, increasing the likelihood of an injury. Increased fall risk and mobility limitations will precipitate patient dependency in instrumental and basic activities of daily living; this in turn results in reduced levels of physical activity. Strengthening those patients by engaging them to gamified daily physical activities is the goal of this research.

2.3 Previous Work

The literature research provided us with two interesting approaches. Inspired by those previous implementations we tried to develop a new functional and gamification aspect of our rehabilitation solution. The two previous prototypes are mentioned below.

The first one conducted in 2006 was based on the use of a pressure mat which gives input of how much pressure the patient puts to each leg while playing a game addressing balance impairment. The scope of that research was to investigate whether coupling foot center of pressure – controlled video games to standing balance exercises will improve dynamic balance control and to determine whether the motivational and challenging aspects of the video game would increase a subject's desire to perform the exercises and complete the rehabilitation process. Their findings demonstrated that graded, dynamic balance exercises on different surfaces can be effectively coupled to video game play. In addition to the training program being enjoyable, all subjects improved their fall rate performance as a result of the video game–based exercise therapy[1].

The second comes with the advent of Microsoft Kinect, a 3D depth sensing technology that provides a low-cost capturing of user's full body movement in 3D space for interaction within game activities without the need for the user to hold an interface device, lead developers, researchers and clinicians to have access to low-cost tracking technologies. A study conducted back in 2011 [11] demonstrated a game that was developed to specifically train reaching and weight shift to improve balance in patients using the Microsoft Kinect sensor. The study provided a preliminary exploration into the usability of a tailored, specific game-based rehabilitation tools. The goal of the developed prototype was to address patients with neurological injuries and indeed the evaluation of the efficiency of the prototype was encouraging both of patients and clinicians perspective. That strengthen our decision to work on the specific rehabilitation field and challenge our team to give a new dimension and functionality to what had already been developed.

Rehabilitation programs conventionally include standing activities, steppers, and overground and treadmill walking with and without body weight support. Messier et al[12] found that an aerobic walking and strength training program helped to decrease postural sway of elderly subjects with knee osteoarthritis. This type of exercise becomes important when we move from standing balance control (stationary base of support) to situations where the center of mass must remain within a moving base of support.

3 Requirements

In the procedure of the game development we had to consider all the possible balance exercises that could benefit our target group. It is obvious that patients with different health background respond differently to various stimuli. What is common among the patients is the consistency and the volume of exercises they are been undertaking. That said, the focus is also on making the gamification of the tasks enough entertaining for them. A few experiments have been conducted to identify what is the optimal set of exercises for various patient groups. In our case we decided to implement a set of exercises targeting people who have already completed a preliminary rehabilitation assessment, meaning that their level of motor control is good enough to handle the complexity and the physical difficulty of the game. We assume that the user already has adequate skills in order to respond to the game's challenges. The exercises prompted by the game are:

- left and right body tilting
- left and right leg raising

This system does not require the user to hold an interface device or move on a pad as the source of interaction within the game is the Kinect. The user's body is the game controller operating in 3D space. In order for the above set up to operate there is a need of a personal computer with the game(software) installed, a monitor and the Microsoft Kinect v2 device. What is also not required is the constant supervision by specialized medical staff during the game sessions, that enables the users to practice on their own pace.

It should be mentioned that the developed game is not strictly addressing people with balance impairment, but it could also be beneficial for healthy people willing to strengthen their core and augment their muscle flexibility.

4 The Game: *Hell Of Balance*

This section describes the game-making process, starting from the idea of the concept to the final working implementation of the Unity application. All the specific game-related information are hereby explained and discussed.

4.1 The Idea

In order to be able to assess balance improvement by exploiting the features and the potential that a device such as Kinect could offer, the main idea behind the game was to make the user perform specific movements and receive real-time feedback. Therefore, main attention has been put on the core movements and exercises that are typically targeting patients suffering from balance impairments. The game expects the user to be able to perform two core movements: lateral bending (right or left) and one-leg stand (by raising either the right or the left knee). Once the exercises have been identified, there was the need to find a way to make them fit within a gamified approach and provide feedback based on the quality of the exercise (e.g. how much is the user getting close to its correct execution).

4.2 Game Mechanics

The next step in the game-development process, was to identify the game mechanics. The main need was to figure out how to combine the rehabilitation background with the general game mechanics rules that typically characterize a game. A lot of inspiration has been given by already existing games with similar mechanics, which means that they have not been reinvented, but somehow re-used in a different context. In terms of specific game development terminology, the mechanics expect the user to avoid hazards in order to proceed in the game. However, unlike one would expect, hazards would be avoided by performing the exercises mentioned in section 4.1, all fit into a well defined game logic that is soon explained.

4.3 Context and Scenery

The scene of the game takes place in an imaginary hell, where the ground is all covered by flowing lava, and an enemy that has the form of a devil is willing to fire hazards to the player. The user's representation in the game is reflected by a skeleton tracked by the Kinect. Indeed, the user sees him/herself constantly projected into the game scene. This choice has been made to provide an additional aid to the user, that can see in real-time how he/she is performing. After this being said, it makes now sense that the user plays the role of an imaginary target that the devil tries to hit with the hazards.

4.4 Game Logic

There's still an important question that needs to be answered: how does the user avoid the hazards? At this point of the description, it's mandatory to explain what's the general flow of the game, according to the specific logic that has been implemented:

1. The game starts by having a single user tracked by the Kinect;
2. The devil moves alternatively from left to right and right to left; once he stops, he starts firing hazards, represented as food items (e.g. hamburgers, hot-dogs, bananas, fries etc.) for an apparent random amount of seconds, with a fixed fire rate of 1 second;
3. The wave of hazards being shot from the same position by the devil has an important attribute: all these hazards target one specific body part. In order to better visualize and understand this logic, please refer to table 1;
4. Instructions on the top-left side of the screen will pop up for every hazard being shot, and they will provide essential information to the user to avoid the hazards. Indeed, before the hazard reaches the bottom edge of the screen (imagined as a boundary after which the hazards would hit the player), the user should perform the requested movement;
5. A relatively numerous wave of hazards reflects into a relatively high amount of seconds that the user would ideally spend in trying to maintain the same position.

Devil's Target	Right Leg	Left Leg	Upper Body (Left)	Upper Body (Right)
Expected User's Exercise	Raise Right Knee	Raise Left Knee	Tilt Body Right	Tilt Body Left

Table 1: Relationship between targets and expected user's exercise

It now makes sense where the degree of difficulty comes from: the user will have to strive to maintain the same position for a high number of seconds. If, for whatever reason, the user fails, he/she would be hit by the hazard and a visual feedback is shown as a blurred full-screen red flash. See in figure 1 an overview of the scenery and User Interface elements that have just been described and some others that are explained in the following subsections. Figures 2, 3, 4 and 5 show examples of game-play, with

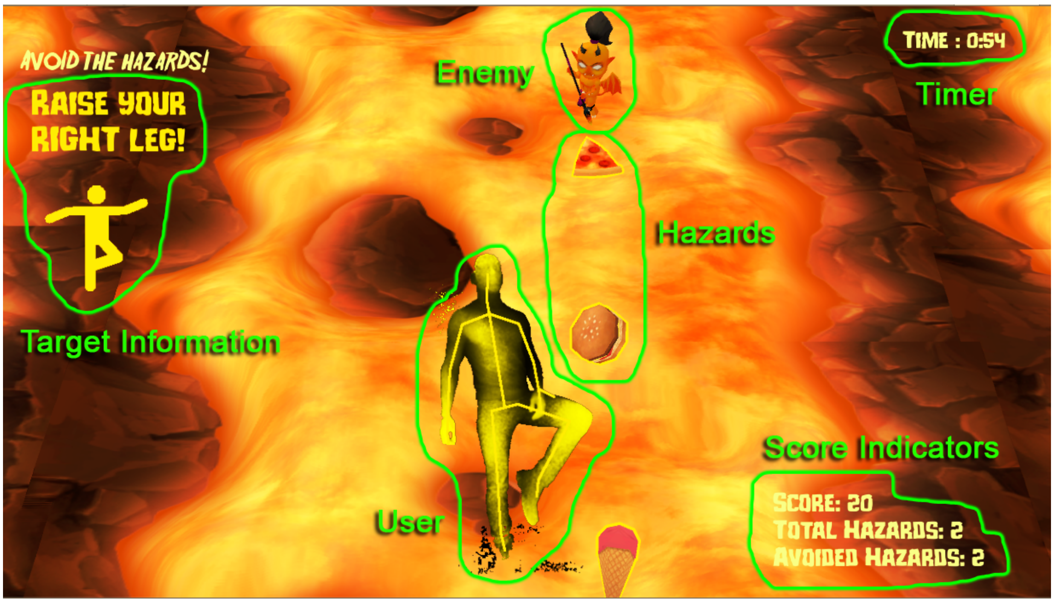

Figure 1: Scenery and User Interface elements

the different exercises to perform.

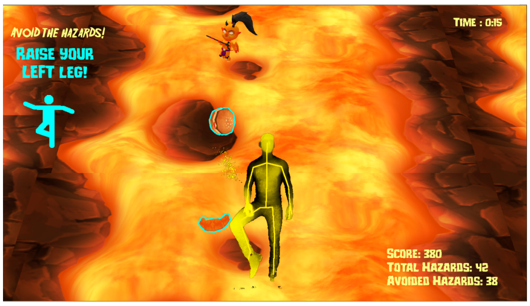

Figure 2: Raise Left Knee Exercise

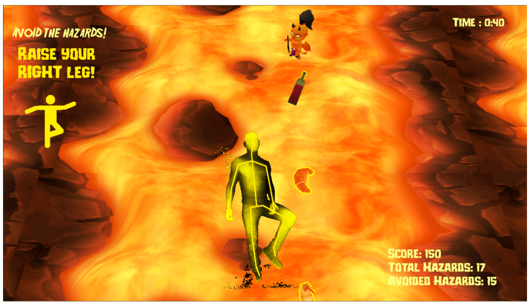

Figure 3: Raise Right Knee Exercise

4.5 Progressing in the Game

What have been described so far represent the core game mechanics and logic, which basically address the question *How to play the game?*. However, there's one important

Figure 4: Tilt Body Left Exercise

Figure 5: Tilt Body Right Exercise

aspect that needs to be presented, which explains how to *progress* in the game. This again introduces some standard game development techniques and aspects that are already well known.

The user progresses in the game by completing *levels*: a level, in this context, is meant to be a game session defined by some specific parameters. The basic idea is that, as long as the user completes a level and progresses, a difficulty coefficient increases and forces the user to strive more to achieve success. It has been decided that the difficulty coefficient had to be represented by the *amount of seconds that the user would stay still*, maintaining one of the core positions already mentioned in table 1. Reflected into the game logic, this means that the higher the level, the longer the devil will keep firing from the same position to the same body target (and by consequence, the higher the amount of hazards to avoid).

A second levels parameter is represented by the *timer*: at the beginning of each level, a timer in the upper-right corner starts, which indicates the duration of the current level. When the countdown ends, the conditions that need to be satisfied in order to successfully pass the level are checked. Again, the higher the level, the higher the duration of the level.

The third parameter is the maximum number of attempts the user has in order to successfully pass the current level. This value varies depending on the level, such that a higher level is featured by a lower number of maximum attempts: in other words, it will be more challenging for the user to pass the levels the more he proceeds in the game. As a final building block of the game logic, hereby the steps to evaluate the user's performance and whether he/she should proceed in the game are listed:

1. When the timer starts (set by default with the specific timer for Level 1), the user plays normally until the timer is over, as described in section 4.4;

2. When the timer is over, the condition for a successful pass is evaluated; given the number of avoided hazards (AH) and the number of total hazards shot (HS) *for the current level*: $\frac{AH}{HS} \geq 0.75 \rightarrow$ the level is considered completed and the next level will automatically start; otherwise, the same level restarts, but the number of currently failed attempts is increased by 1 (section 4.6 explains the meaning of the mentioned parameters and the scoring mechanism);

3. Given the number of currently failed attempts as CFA and the maximum number

of attempts for the current level as MA: if $CFA == MA \rightarrow$ the game stops, and the user is recommended to have a break and retry the next day;

4. The game ends when the user successfully passes the last available level: at that point, a summary screen is shown with the stats collected throughout the game-play; scoring is explained in section 4.6.

The developed prototype comes with a limited, hard-coded set of levels, each one with its own specific parameters, as table 2 reports: the column *N° of seconds of activity* indicates the total number of seconds for which the user is expected to perform a balance exercise. The meaning of the values from the column *Penalty Weight* is explained in section 4.6.

Level	**Timer (minutes)**	**Max. N° Attempts**	**Seconds of activity**	**Penalty Weight**
1	5	5	5	0.5
2	8	3	8	0.35
3	10	1	11	0.15

Table 2: Level Parameters

4.6 Scoring and end of the game

As also shown in figure 1, the bottom-right corner displays the *score indicators*, including scored points, total hazards shot and avoided hazards:

- The *Scored points* increase by 10 for each avoided hazard;
- *Total hazards shot* increase by 1 for each hazard being shot by the devil *in the current level*;
- *Avoided hazards* increase by 1 for each hazard the user successfully avoid *in the current level.*

As mentioned in section 4.5, these parameters are essential to evaluate the progress of the user in the game. At the end of the game, when all levels are completed, the summary screen shows a set of valuable information for the user:

- Total minutes played;
- Total scored points;
- *Success score*: this is obtained by calculating the overall success ratio. Given the total number of hazards shot (THS) and the total number of avoided hazards (TAH), the success score (SS) will be calculated as: $SS = \frac{TAH}{THS} \cdot 100$. It will always be at least 75%;
- A *Penalty Score* (PS) is calculated on top of the success score, depending on how many failed attempts the user scored throughout the game;
- The *Final Score* (FS) is the real final indicator, calculated as: $FS = SS - PS \cdot SS$.

It's worth pointing out the calculation of the PS.
Given: $i \mid i \in \{1, 2, 3\}$, FA_i is the number of *failed attempts* for level i, and PW_i is the *penalty weight* of each level $\rightarrow$ a *weighted average of failed attempts* (FA_{avg}) is calculated as:

$$FA_{avg} = \frac{\sum_i PW_i \cdot FA_i}{\sum_i PW_i} = \frac{\sum_i PW_i \cdot FA_i}{1} = \sum_i PW_i \cdot FA_i$$

The *Penalty Score* PS is then given by:

$$PS = \frac{FA_{avg}}{TFA} \cdot 100$$

where TFA is the *total number of failed attempts*. See in figure 6 an example of a summary screen at the end of the game.

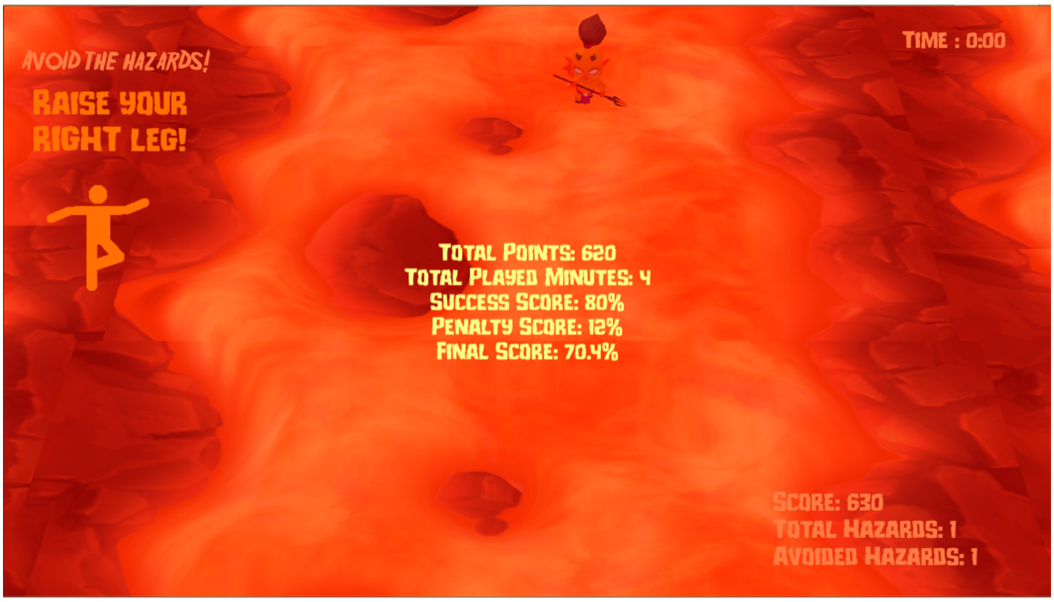

Figure 6: Summary screen with game stats

4.7 Fine-tuning

A preliminary version of the game was conceptualized without the coloring of the outline of the hazards, which can be clearly noticed in figures 2, 3, 4 and 5. Indeed, the original idea was to distinguish whether the target was the upper body or a leg by using different scale sizes of the hazards. However, it was observed that this criteria was not enough evident.

For the above mentioned reason, in order to ease the obviously existing learning phase of the user, we included a cognitive factor that expects the user to be able to recognize the colors and correctly associate them with the corresponding exercise. Four main colors (yellow, green, magenta and cyan, highly contrasting with each other and easy to distinguish) have been identified and each one associated with an exercise. As soon as the hazards are shot, their outline gets colored and an iconized body picture is shown on the left side of the screen, that suggests the user which exercise to perform in order to avoid the hazards. This aspect of the game is meant to represent an aid for the user in a preliminary phase of the game, and it's expected that the more he/she plays, the more he/she will get confident with the colors and be able to pay attention to the rest of the screen instead of focusing only on the suggestions.

Besides this, the rest of the game more or less maintains the same look and principles since it was originally designed.

5 Implementation Details

What follows are a few implementation details, including non-trivial design choices and details about the integration of the Microsoft's Kinect with the game; relevant code snippets are included in the appendix (A).

5.1 User's Body Tracking

User tracking is an essential element of the game development process, and high priority has been given to it. Indeed, depending on the choices and the results concerning this aspect, the rest of the game had to be consistent with it. The main outcome is that the logic of the Kinect's tracking feature and the logic of the game have been decoupled as much as possible, in order to avoid conflicts and minimize the risk of encountering unexpected issues or behaviors with Unity's core elements.

The user's body is continuously tracked by the Kinect and fully projected inside Unity's scene; however, there's no spatial relationship between the projected skeleton and the scene: the only reason behind this is to provide real-time feedback to the user, who would be able to judge whether the exercise is performed correctly by combining his mirrored position with the feedback about being hit or not. The idea is to mimic the situation of somebody performing exercises in front of a mirror.

The real interaction between Unity's scene elements and the user's tracked body occurs as an atomic step performed when the hazards hit the imaginary boundary located just beneath the bottom edge of the screen. At the time of the hit, the Kinect's logic checks if the user is performing the exercise correctly as expected by the hazard. Sections 5.1.1 and 5.1.2 explain more in depth the principles behind the tracking of the user's movements.

5.1.1 Legs tracking

When the target of the hazards is either the right or left leg, the mechanism for tracking the key elements of the Kinect's skeleton goes in action. The relevant body-joints are the *knee joint* (either left or right) and the *spine-base joint*. Since the Kinect manifests difficulties in tracking and understanding depth distances, the tracking principle only works if the user raises the knee laterally, and not by bringing it in front of him/her. The (x, y) coordinates of the mentioned joints are obtained from Kinect's internal coordinate system; given KJ_y as the y coordinate of the knee joint and SBJ_y the y

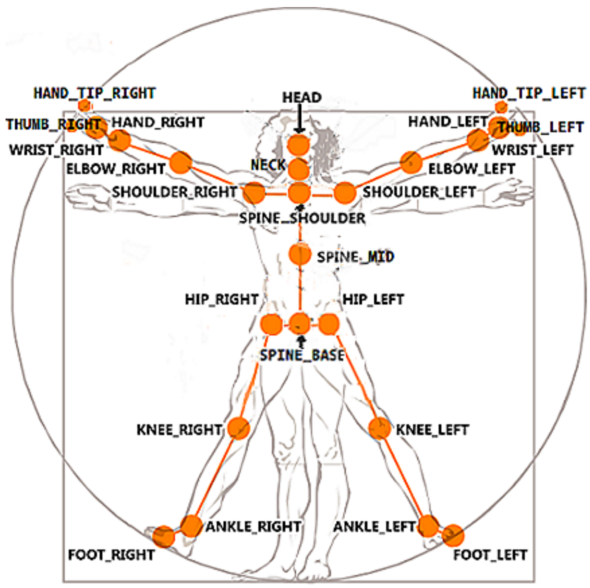

Figure 7: Joints enumeration defined by Kinect

coordinate of the spine-base joint, a simplified calculation of the sufficient condition such that a hazard is avoided is:

$$KJ_y \geq SBJ_y$$

Please refer to figure 7 for information about naming conventions of the joints detected by the Kinect.
It's quite evident that this calculation doesn't take into consideration body calibration adjustments, which are explained in section 5.2.

5.1.2 Upper body tracking

In this case, the relevant body joints are the *neck joint* and, again, the *spine-base joint*. However, the logic to detect whether the user is performing the requested exercise correctly or not is a bit tricky and more complex than the previous one explained in section 5.1.1. Unlike it happens with the legs, this principle lies on the *slope of a straight line in the Cartesian plane*. Figure 8 shows the basic mathematical principle. The Kinect again tracks the (x, y) coordinates of the mentioned joints, obtaining nothing but the (x, y) coordinates of two points in an imaginary Cartesian plane, consistent with Kinect's coordinate system. By tilting the upper body, indeed, the user simulates the behaviour of a straight line with a positive (tilting right) or negative (tilting left) slope.

According to figure 8, point $P2 = (x_2, y_2)$ represents the neck joint position, while point $P1 = (x_1, y_1)$ represents the spine base joint position. Again relying on math principles, it's easy to calculate the slope m between the two points $P1$ and $P2$ as:

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

The slope m of a straight line has then a dependency with the angle θ:

$$m = tan(\theta) \implies \theta_{rad} = tan^{-1}(m) = arctan(m)$$

At this point, the last step is calculating the actual body tilting angle, which can be seen in the Cartesian plane as the angle that the line forms with the positive y axis. This value changes depending on the value of the slope:

- $m \geq 0 \implies \alpha_{deg} = 90° - \theta_{deg}$
- $m < 0 \implies \alpha_{deg} = -90° - \theta_{deg}$

Notice that α is either positive or negative, hence the condition to be checked to evaluate a correct execution of the exercise is:

- User should tilt *left* $\rightarrow$ User avoids hazards $\iff \alpha < -\beta$
- User should tilt *right* $\rightarrow$ User avoids hazards $\iff \alpha > \beta$

Where β has been used to refer to a hard-coded threshold: $\beta = 20°$.

On top of all these mathematical interpretations and calculations, the practical outcome is that the user avoids the hazard if the inclination of his/her body is of at least $\pm 20°$ with respect to the orthogonal line to the ground. Please refer to figure 9 for a better visual understanding of the above mentioned angles.

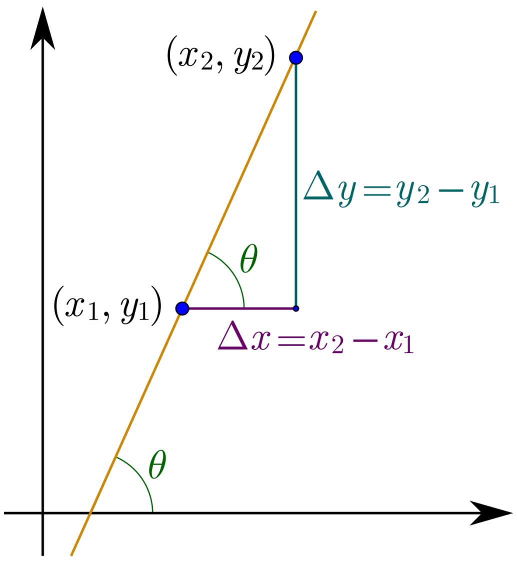

Figure 8: Cartesian plane and calculation of the slope of a straight line

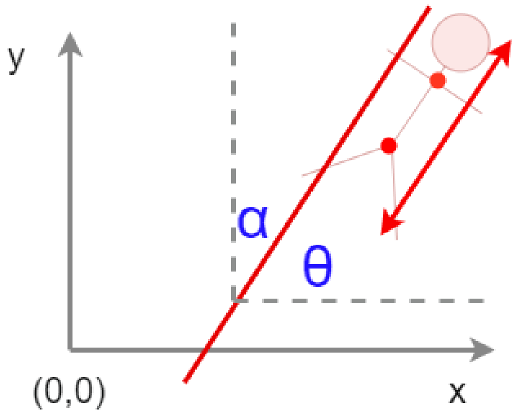

Figure 9: Simplified representation of body tilting and involved angles

5.2 Calibration

As mentioned in section 5.1.1, calibration is a technique that should not be underestimated while integrating Kinect's body tracking within a game or more in general an application. By calibrating the user, the developer's would guarantee that the application works as expected with bodies of different size and shape. The main principle lies on normalization of distances, and that's why a common calibration positions is the *T-pose*. Although the group agrees this is quite crucial, the developed game didn't have to rely on a calibration step in order to be suitable to different people. The reason behind this is that, as explained in section 5.1, what is mainly tracked are the positions of body joints and they are not related in any way with the Unity's scene elements and distances. Instead, all the calculations are made in a spatial system related to the body itself, where there's no explicit reference to elements of the surrounding environment.

In the case of leg tracking, however, a calibration factor has been introduced. The previously explained formula $KJ_y \geq SBJ_y$ (see 5.1.1) expects the user to lift the knee at least at the same height of the spine. This might seem easily feasible by somebody, but user's flexibility matters at this point. In order to calibrate this movement, it has been decided to introduce an offset depending on the length of the leg of the user.
Given j_1 and j_2 the position of the two joints (knee and spine-base):

$$Offset = d(j_1, j_2) = \sqrt{(j_{1,x} - j_{2,x})^2 + (j_{1,y} - j_{2,y})^2 + (j_{1,z} - j_{2,z})^2}$$

In light of this result, the complete condition such that a hazard is avoided becomes:

$$KJ_y \geq SBJ_y - \frac{Offset}{2})$$

6 Testing

6.1 Test Specification

A preliminary study had been designed, consisting of a short verbal introduction to the purpose of the game and how to control it. Each subject were scheduled to play the game for a session of 30-minutes and afterwards complete a short questionnaire about their experience.

#	Question	
1	Were you able to completely play and understand the game after the short verbal intro?	Y/N
2	Did you manage to progress to the second level during your first try?	Y/N
3	Did you manage to progress to the second level during the full duration	1-10
4	How well did you feel like you improved your balance during the test?	Y/N
5	Did you feel increased motivation because of the gamification elements such as obtaining a score.	Y/N
6	Which exercise was most difficult in your opinion?	Leg/Tilt
7	Did you during the game understand the different colours used together with the different exercises, thus improving your reaction time?	Y/N
8	Would you consider playing this game every day as part of rehabilitation?	Y/N
9	Would you consider playing this game regularly after ended rehabilitation, to further improve your balance?	Y/N
10	How well did you overall like the game?	1-10

Table 3: User Test Questions

6.2 User Tests

The scope of the prototype was to directly cope with patients with severe balance issues. In the testing phase, the developed prototype was introduced to testing subjects that they haven't be diagnosed with balance disorders. However they seemed concerned and not very confident about their balance performance.

Subject 1
A 67-year-old male. He always had good balance and has been an avid runner since he was very young, first orienteering and later regular running in the ranges of 5-10 km. During a run approximately four years ago, he broke one of his left toes. While doing his best regularly trying to rehabilitate himself with assistance from physiotherapist, he broke another bone in the same foot mere one and a half year after the first injury. After more physiotherapist assisted rehabilitation he is now able to run the same distances as before and also participates in weekly gymnastics for men his age where he, according to himself, is amongst the best, balance wise. His balance is however not as good as it used to be before his foot injuries.

Subject 1 was able to finish the scheduled 30 minutes after a short verbal introduction. He managed to reach the second level after three level one attempts. He stated that he would have succeeded reaching level three had he been given longer time.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
No	No	Yes	5	Yes	Leg	Yes	Yes	No	7

Table 4: Subject 1: Questionnaire Answers

The subject further expressed that while he could see how the game would be beneficial for people who wouldn't actively engage in physical activities on their own, he would rather prefer exercise in a more social way, like he is currently doing.

Subject 2
Our second subject is a 73-year-old woman. She has never been physically active which might have been one reasons she suffered from a blood clot nearly six years ago. Due to other personal issues she had a hard time following her rehabilitation and were barely walking a year after the incident. Luckily the municipality realised this and eventually assigned her a social worker who would take her on short daily walks in the neighbourhood. She progressed very well and soon became overly brave and while biking down a tunnel ramp she fell and broke her collarbone. Her second recovery went better compared to the first as her mental state was better at the time, yet her physical abilities remain rather poor to this day and she never bicycles nor drives a car and walks very slowly.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
No	No	No	7	No	Leg	No	Yes	Yes	9

Table 5: Subject 2: Questionnaire Answers

The subject was not able to play the game for the scheduled duration of 30 minutes, due to fatigue. After completing the questionnaire, the subject expressed that this was very new to her, as she in general is new to computer games and information technology. She did not rule out that she would play it more if she were able to, but she would most likely need some kind of motivation as she did not feel like she would progress very quickly. Despite physical exhaustion, the subject had an exciting experience.

7 Evaluation

The results, described in the previous chapter, were in overall very encouraging. The subjects indicated that the game was challenging, fun, and would be a welcome addition to current treatment programs. The goal of the trial to investigate if the kinect-controlled video game–based exercises would improve dynamic balance control in 2 different subjects were partly met. The group hypothesized that the inclusion of motivational and functional gaming to rehabilitation should increase the patient's desire to perform their rehabilitative exercises; and thus the patients should exhibit improved dynamic balance control post-exercise. Both the test subjects experienced improved balance control during the trials ($Q4 \geq 5$) while only the first subject felt the gamification elements induced increased motivation. As the second subject didn't complete the full experiment, it is likely that this whole experience were so unfamiliar to her, that all her attention were used performing and identifying the correct exercises. Her lack of previous experiences with computer games is also likely to have impacted her abilities understand the concept within such a short period of time.

The group was a bit surprised that she was not able to complete the scheduled trial due to physical (and perhaps mental) exhaustion. The choices made with regards to the amount of seconds performing and in-between each exercise were purely made from a hypothesis during the development and design phases of the game. It is very likely that the first level is too advanced to some of the target customers. A more comprehensive trial study, with more participants ($n \geq 20$) would bring a better understanding of different patients physical performance.

8 Discussion and Conclusions

Throughout this report, a deep analysis and description of the solution the group came up with has been presented. Supported by strong healthcare background motivations and inspired by previously existing work, this report along with the implemented application aims to provide a valid balance assessment and rehabilitation solution. All major steps of the development process have been described, diving into essential technicalities when necessary, however a simplified and straight-forward approach ideally would guide the reader to properly understand all the concepts.

As a final stage of this documentation, hereby a few notes are provided concerning the group's feedback about the overall experience.

Through the game prototyping and development process the group didn't encounter major difficulties. After a preliminary phase where the scope of the project and the healthcare background have been discussed, the next phases proceeded without relevant drawbacks or difficulties. The overall satisfaction is high although the results of the user tests didn't fully satisfy the expectations; the group do believes that evaluation on top of the user tests would mostly be suitable with a research-experimental approach, whereas the main focus of the project has been put on the implementation of a real application and the understanding of the healthcare background.

Future directions in this research will be more formal usability testing followed by a larger feasibility trial to determine if the use of the game as a training tool improves balance outcomes.

By the end of this research and prototype development it is important to mention that the majority of our initial goals are fulfilled. Based on the literature findings we pointed out what would be more beneficial, in terms of exercises, for our target group to improve or recover from a balance disorder and we managed to develop a quite interesting gameplay to keep the patients engaged.

As a conclusion, the foundation for future work is presented:

- A more complete game experience would benefit from a more advanced levels fine-tuning, whereas the current implementation relies more on a prototype-based approach (this would turn into implementing more levels and introducing more level-based parameters);
- The current version of the game doesn't save any progress, although it would be definitely essential to keep track of patient's progress in a time-series basis;
- Introducing an ad-hoc tutorial to ease the game-play and the learning phase of a broader range of people;
- Conducting a quantitative research experiment would guarantee more detailed and more advanced results in terms of user testing and evaluation.

A Appendix

See below some code snippets written in C# relevant to the Kinect's tracking functionalities and the mathematical interpretations described in section 5.

A.1 Body Tracking Code Snippets

The following snippets are extracted from the `PlayerController.cs` script that holds variables and methods related to the tracking of the user's movements, depending on which he/she gets hit or not by the hazards. Not all the referenced fields have been included, but it should be quite intuitive to skim through the code and grasp the meaning and verify that it actually reflects what has been explained in section 5.1.

```
//Relevant fields
private const KinectInterop.JointType rightKneeJoint = KinectInterop.JointType.KneeRight;
private const KinectInterop.JointType leftKneeJoint = KinectInterop.JointType.KneeLeft;
private const KinectInterop.JointType spineBaseJoint = KinectInterop.JointType.SpineBase;
private const KinectInterop.JointType neckJoint = KinectInterop.JointType.Neck;

public void TrackLeg(string targetLeg)
{
   //Check if body is tracked, otherwise skip
   if (!IsPlayerTracked())
      return;

   KinectInterop.BodyData bodyData = kinectManager.GetUserBodyData(kinectManager.GetUserIdByIndex(userIndex));

   //Get joints data
   KinectInterop.JointData rightKneeJointData = bodyData.joint[(int)rightKneeJoint];
   KinectInterop.JointData leftKneeJointData = bodyData.joint[(int)leftKneeJoint];
   KinectInterop.JointData spineBaseJointData = bodyData.joint[(int)spineBaseJoint];
   float rightKneeY = rightKneeJointData.position.y;
   float leftKneeY = leftKneeJointData.position.y;
   float spineBaseY = spineBaseJointData.position.y;
   float targetLegY = -100f; //Default value (impossible to reach, used just for initialization)
   KinectInterop.JointData targetJointData = rightKneeJointData; //Init default but it will be overwritten (cannot set to null)
   switch (targetLeg)
   {
      case EnemyController.LEG_RIGHT:
         targetLegY = rightKneeY;
         targetJointData = rightKneeJointData;
         break;
      case EnemyController.LEG_LEFT:
         targetLegY = leftKneeY;
         targetJointData = leftKneeJointData;
         break;
      default:
         break;
   }

   if (targetLegY != -100f)
   {
      double distance = LenghtBetweenTwoJoints(targetJointData, spineBaseJointData);
      bool isAvoided = targetLegY >= spineBaseY - (distance / 2);
      if (!isAvoided)
         Hit();
      else
         gameController.IncreaseScore();
      gameController.AddHazard(isAvoided);
   }
}

public void TrackTilting(string targetDirection)
{
   //Check if body is tracked, otherwise skip
   if (!IsPlayerTracked())
      return;

   KinectInterop.BodyData bodyData = kinectManager.GetUserBodyData(kinectManager.GetUserIdByIndex(userIndex));

   KinectInterop.JointData neckJointData = bodyData.joint[(int)neckJoint];
   KinectInterop.JointData spineBaseJointData = bodyData.joint[(int)spineBaseJoint];
   double m = AnglesHelper.AngularCoefficientBetweenTwoJoints(neckJointData, spineBaseJointData);
   double tiltingAngle = AnglesHelper.GetTiltingAngle(m);
   //if tiltingAngle is negative, we're inclining LEFT, otherwise RIGHT

   bool isAvoided;
   double minTiltingAngle = 20.0;
   if (targetDirection.Contains("LEFT"))
   {
      minTiltingAngle = -thresholdTiltingAngle;
      isAvoided = tiltingAngle < minTiltingAngle;
   }
   else
   {
```

```
        minTiltingAngle = thresholdTiltingAngle;
        isAvoided = tiltingAngle > minTiltingAngle;
    }
    if (!isAvoided)
        Hit();
    else
        gameController.IncreaseScore();
    gameController.AddHazard(isAvoided);
}
```

A.2 Mathematical Calculations Code Snipptes

The following snippets refer to specific mathematical calculations used to support the logic of the game. They reflect what has been described in section 5.1. The code has been extracted partly from the `PlayerController.cs` script (`LenghtBetweenTwoJoints(...)`), while the functions related to angle calculations are included in the `AnglesHelper.cs` script.

```
//From PlayerController.cs
public static double LenghtBetweenTwoJoints(KinectInterop.JointData j1, KinectInterop.JointData j2)
{
    return Math.Sqrt(Math.Pow(j1.position.x - j2.position.x, 2) + Math.Pow(j1.position.y - j2.position.y, 2) + Math.Pow(j1.position.z -
          j2.position.z, 2));
}

//From AnglesHelper.cs
public class AnglesHelper
{
    public static double AngularCoefficientBetweenTwoJoints(KinectInterop.JointData j1, KinectInterop.JointData j2)
    {
        return (double)(j1.position.y - j2.position.y) / (j1.position.x - j2.position.x);
    }

    public static double GetTiltingAngle(double m)
    {
        double lineAngleRadians = Math.Atan(m);
        double lineAngleDegrees = RadianToDegree(lineAngleRadians);
        //lineAngleDegrees will be either positive or negative
        if (m > 0)
            return 90 - lineAngleDegrees;
        else
            return -(lineAngleDegrees + 90);
    }

    private static double RadianToDegree(double angle)
    {
        return angle * (180.0 / Math.PI);
    }
}
```

References

[1] A. L. Betker, T. Szturm, Z. K. Moussavi, and C. Nett. Video game–based exercises for balance rehabilitation: a single-subject design. *Archives of physical medicine and rehabilitation*, 87(8):1141–1149, 2006.

[2] C. Boult, R. L. Kane, T. A. Louis, L. Boult, and D. McCaffrey. Chronic conditions that lead to functional limitation in the elderly. *Journal of gerontology*, 49(1):M28–M36, 1994.

[3] J. Broeren, M. Rydmark, and K. S. Sunnerhagen. Virtual reality and haptics as a training device for movement rehabilitation after stroke: a single-case study. *Archives of physical medicine and rehabilitation*, 85(8):1247–1250, 2004.

[4] J. H. Cauraugh and J. J. Summers. Neural plasticity and bilateral movements: a rehabilitation approach for chronic stroke. *Progress in neurobiology*, 75(5):309–320, 2005.

[5] L. P. Fried, S. J. Herdman, K. E. Kuhn, G. Rubin, and K. Turano. Preclinical disability: hypotheses about the bottom of the iceberg. *Journal of Aging and Health*, 3(2):285–300, 1991.

[6] J. Gill, J. Allum, M. Carpenter, M. Held-Ziolkowska, A. Adkin, F. Honegger, and K. Pierchala. Trunk sway measures of postural stability during clinical balance tests: effects of age. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 56(7):M438–M447, 2001.

[7] J. E. Harris, J. J. Eng, D. S. Marigold, C. D. Tokuno, and C. L. Louis. Relationship of balance and mobility to fall incidence in people with chronic stroke. *Physical therapy*, 85(2):150–158, 2005.

[8] D. Kriegsman, D. Deeg, J. van Eijk, B. Penninx, and A. Boeke. The additional influence of disease-specific characteristics on mobility in patients with different chronic diseases: a study in the netherlands. *J Epidemiol Community Health*, 51:676–85, 1997.

[9] A. D. Kuo. An optimal control model for analyzing human postural balance. *IEEE transactions on biomedical engineering*, 42(1):87–101, 1995.

[10] G. Kwakkel, R. C. Wagenaar, J. W. Twisk, G. J. Lankhorst, and J. C. Koetsier. Intensity of leg and arm training after primary middle-cerebral-artery stroke: a randomised trial. *The Lancet*, 354(9174):191–196, 1999.

[11] B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. S. Rizzo, and M. Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. In *Engineering in medicine and biology society, EMBC, 2011 annual international conference of the IEEE*, pages 1831–1834. IEEE, 2011.

[12] S. P. Messier, T. D. Royer, T. E. Craven, M. L. O'toole, R. Burns, and W. H. Ettinger. Long-term exercise and its effect on balance in older, osteoarthritic adults: Results from the fitness, arthritis, and seniors trial (fast). *Journal of the American Geriatrics Society*, 48(2):131–138, 2000.

[13] M. Visintin, H. Barbeau, N. Korner-Bitensky, and N. E. Mayo. A new approach to retrain gait in stroke patients through body weight support and treadmill stimulation. *Stroke*, 29(6):1122–1128, 1998.