SCHOOL OF MECHANICAL
ENGINEERING

UNIVERSITY OF LEEDS

Innovative Approaches to Balance Rehabilitation: A Study of Microsoft Kinect-based Motion Capture Technology

William Terry - 201425471

Prof. Shane Xie

Academic Year 2022/2023

SCHOOL OF MECHANICAL ENGINEERING

UNIVERSITY OF LEEDS

MECH3895 – Individual Engineering Project

PROJECT TITLE: Innovative Approaches to Balance Rehabilitation: A Study of Microsoft Kinect-based Motion Capture Technology

PRESENTED BY William Mark Patrick Terry

SUPERVISED BY Prof Shane Xie

If the project is industrially linked, tick this box and provide details below

COMPANY NAME AND ADDRESS:

STUDENT DECLARATION (from the "LU Declaration of Academic Integrity")

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance. I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance. I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Date 08/04/23

Signed

Table of Contents

Nomenclature

ToF - Time of Flight

BP - Blueprints

FPS - Frames per Second

RMSE - Root Mean Squared Error

List of Figures

Abstract

Post-medical care for patients after a fall is at an all-time high, costing a fortune in medical bills for all nations. It has been verified that training patients to balance pre-fall is an excellent way to reduce the risk of a fall in patients over 65 and up. Conventional training is lacking due to the resources required to train a single patient. Robotics could help patients in this pre-fall stage to free up costs and time wasted treating unnecessary falls. From background research, the initial plan for this program was laid out as a possible solution to this significant problem. The project's goal was to make sure the device was easily accessible to all people to mitigate the need for expensive and advanced methods used in hospitals today. The project used a Microsoft Kinect game that explicitly trains the required muscle groups to improve balance, making the player the controller for ease of use for a non-technical user. Validation of the motion capture accuracy was taken to confirm the method of balance training as a viable option in a home. A small study group took part in a survey questioning their individual experiences of the game over seven days, confirming if the use in a home is a viable option.

Chapter 1 - Introduction

1.0 Project Background and Motivations

A fall can happen to anyone, anytime, and generally, recovery is easy. However, for some, the impact of falls can be life-changing. The world health organisation states that "Falls are the second leading cause of unintentional injury deaths worldwide" [1] and that "37.3 million falls are severe enough to require medical attention occur each year" [1]. These falls can cause permanent damage, forcing people to live with disabilities for the rest of their lives. The elderly demographic (>65) is most affected by falls each year [1].

Data from the US suggests that 20-30% of older adults who fall suffer moderate to severe injuries, with many facing a considerable risk of hip injuries. Furthermore, Hip injury risk will likely increase with age [2]. The impact of a hip injury is significant and can immobilise a person for months or potentially the rest of their lives, leading to poor quality of life. In addition, there are indirect impacts while immobile; blood clots can form in the legs or lungs, causing further injury and discomfort, such as bedsores.

Falls do not just affect individuals. Health organisations spend a tremendous amount of money every year dealing with falls. For example, the WHO reported, "For people aged 65 years or older, the average health system cost per fall injury in the Republic of Finland and Australia are US$ 3611 and US$ 1049, respectively" [1], although this excludes the indirect expenses falls cause. Healthcare systems already face significant resource constraints, and reducing falls and their impact could alleviate the pressure on finances and staff, redirecting money to more urgent cases. So what can be done to prevent falls?

1.1 Aims

The aim is to produce an effective Microsoft Kinect motion capture program that uses exercises to significantly decrease the chances of elderly persons (65+ years) fall rate while making it easily accessible worldwide, cheap and engaging.

1.2 Objectives

- Use desk-based research to identify the most common falls and the main muscle groups that can help reduce the impact of these falls through training.
- Use motion capture tools to create a program containing two games that train targeted muscle groups and engage users over a sustained period.
- Ensure the device is easily accessible to the general public and affordable.
- Validate the motion capture program's accuracy to ensure correct form during exercises is executed.
- Use a focus group to engage with the project deliverables and conduct a questionnaire to see if their confidence has increased while participating in the study.
- Use a focus group to form positive opinions on the enjoyability and engagement of the games and test if users feel an increase in balance confidence.

1.3 Project Layout

This report outlines the six project sections:

- Chapter 1: The introduction, providing background information and the motives of the project
- Chapter 2: The theoretical background, providing research behind the most common falls and the maths behind Kinect motion capture.
- Chapter 3: Motion Capture, explains how motion capture data is collected and used in unreal engine
- Chapter 4: Goes through the implementation of the motion capture software in the games
- Chapter 5: Analysis of Kinect motion capture through validation
- Chapter 6: Conducting and evaluating the survey.

Chapter 2 - Theoretical Background

2.1 Introduction

In this chapter, observations will be outlined on ‘what the most common fall is’, ‘what muscles are used for balance’, and finally ‘how can we train these target muscles’. Another part of the section will cover how a Microsoft Kinect captures motion and what the data from this will go on to do.

2.2 Common Falls and Muscle Analysis

This section explicitly covers the three main topics of falls listed above. In each subchapter, research will be conducted to find the project’s best target area and the game development’s primary focus.

2.2.1 Most Common Fall

Measuring how stable someone is can help prevent the most common falls, which in the elderly happens to fall forward at 44% [3]. Forward falls mainly came down to trips and sudden movement [3]. The average person’s centre of mass (CoM) is in their lower abdomen, so falling forward is just when the lateral distance between the centre of mass is beyond the point of stabilisation. Falling sideways is the leading cause of hip injuries (95% of hip injuries are sideways [4]) which is one of the most concerning health risks to an elderly person as it is one of the hardest to recover from at a late age. From this, choosing a focal point of training sideways movement should help reduce this statistic.

2.2.2 What Are the Targeted Muscles for Training?

Reducing the risk of falls is as simple as increasing balance and training those muscle groups. The stabiliser muscles are not used in any movement but act to stabilise one joint, improving stability [5], so exercise usually focuses on improving those muscles. The most prominent stabiliser muscles involved in balance are **pelvic floor muscles, transversus abdominis, multifidus, obliques, rectus abdominis, erector spinae and the diaphragm**[5, 6] seen in Fig 2.2.1.

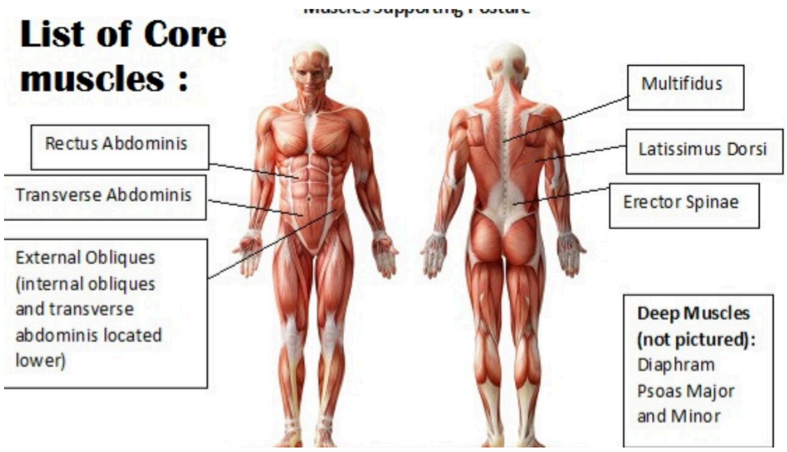


Fig 2.2.1 Core Stability Muscles [6]

Balance is also affected by the stabiliser muscles in the legs, with the main one being the **Gluteus medius**, with others having less effect seen in Fig 2.6.2. The lower muscles help maintain proper biomechanics while walking and running [6]. In contrast, the torso muscles help with optimal body function [6], which targets the TUG movement.

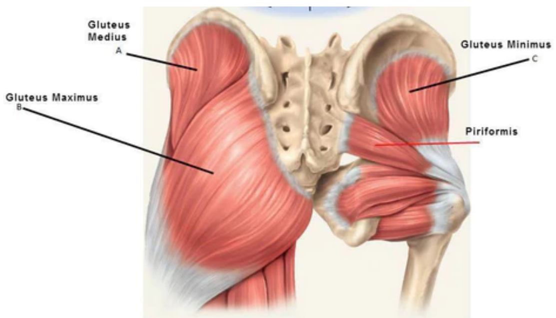


Fig 2.2.2 Hip Stabiliser Complex [4]

2.2.3 How to Target These Muscles Through Exercise?

Yoga [7] and tai chi [8] are two great general ways to target stabiliser muscles. Tai chi targets core and stabiliser muscles, training balance, while yoga similarly focuses on lower stabiliser muscles.

The yoga poses that targeted the right muscle groups consisted of planks, warrior, downwards facing and upward facing dog, forward fold, a mountain with arms up, chair and half lift [9]. While all these positions were analysed and showed improvements to the Rectus abdominis and erector spinae muscles, they are not viable for elderly people. The most accessible and advantageous positions are warrior, chair, halfway lift and mountain pose (as they can be done standing up) and then dog facing up and down. Plank and forward fold should only be introduced once they have developed a more comprehensive range of motion. These relatively easy motions can develop a good base for these stabiliser muscles.

Tai chi focuses on building a sturdy core, working on the diaphragm, pelvic muscles, lower abdomen, small spine, and pelvic floor muscles [8]. Positions targeting these muscle areas include slow movements in standing positions. Working on this range has been shown to improve gait control and will not only improve balance but reduce back pain [8]. Working on both tai chi and weekly yoga exercises, all the motions needed for balance will improve the quality of life for any elderly person.

2.3 Microsoft Kinect Motion Capture Analysis

In this section, an analysis of how a Microsoft Kinect works will be explored as a form of skeletal tracking inside UnReal engine. Research on the Kinect's sensors and the full capabilities of this widely available hardware will be conducted to find its benefits and limitations.

2.3.1 Hardware of a Microsoft Kinect

The hardware of a Kinect is straightforward and consists of 4 main components; an RGB camera, an IR camera, an IR projector and a microphone array [10]. The only focus of the project is motion capture, so the projector's microphone array is unused. The RGB camera captures 1080p footage at 30 frames per second. The project's major components are the IR camera (depth sensor) and the IR projector, in charge of motion detection and capture.

2.3.2 Software of a Microsoft Kinect

For this project, the Microsoft Kinect will solely be used for tracking the skeletal bodies of the patients. Microsoft breaks skeletal tracking into different sections to achieve usable skeletal mesh data.

Firstly, using the IR projector and the depth camera, a depth map is created using the method ToF (time of flight). On a basic level, "infrared light is emitted with modulated waves and detects the shifted phase of the returning light" [11].

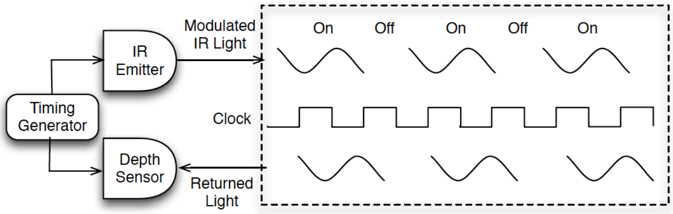
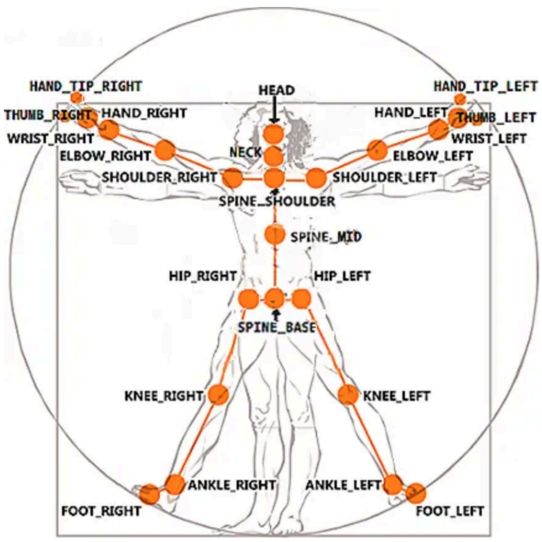


Fig 2.3.1 shows the Microsoft Kinect's time of flight depth detection [12]

Fig 2.3.1 shows the modulation of the emitted light and the depth sensor picking up the reflected light on a different phase to the original emission. A basic depth map is created by calculating the time difference between the emission and receiving of the light.

Next, using Microsoft's supplied SDKs, skeletal data is estimated using the depth map. Microsoft achieves this using a complex machine-learning algorithm [13]. However, the project's only interest is how the data is displayed so it can be manipulated in the games later. The Kinect calculates joint positions for 25 specific body joints and outputs an array, each containing 11 properties [14].



SPINEBASE	0
SPINEMID	1
NECK	2
HEAD	3
SHOULDERLEFT	4
ELBOWLEFT	5
WRISTLEFT	6
HANDLEFT	7
SHOULDERRIGHT	8
ELBOWRIGHT	9
WRISTRIGHT	10
HANDRIGHT	11
HIPLEFT	12
KNEELEFT	13
ANKLELEFT	14
FOOTLEFT	15
HIPRIGHT	16
KNEERIGHT	17
ANKLERIGHT	18
FOOTRIGHT	19
SPINESHOULDER	20
HANDTIPLEFT	21
THUMBLEFT	22
HANDTIPRIGHT	23
THUMBRIGHT	24

Fig 2.3.2, Fig 2.3.3 shows each joint tracked by the Kinect and its associated value [14]

Each joint above is assigned their specific number in the outputted array, shown in Fig 2.3.3. In the array of the 25 joints, each joint has a sub-array made up of; colour (x, y); depth (x, y); camera (x, y, z, w); and orientation (x, y, z, w) [14]. The colour and depth coordinates are for the joint location in each respected camera on a

2d plane (colour - RGB camera, depth - IR camera). The camera coordinate system is essential for this project as it gives 3d coordinates for each joint with the origin based on the lens of the IR camera [14] (as shown in Fig 2.3.4). The coordinates are given in metres between the tracking range of 0.5 - 4.5 m [15], with positive and negative values, depending on where the camera is set up.

Fig 2.3.4 The kinects coordinate system [14]

With the camera coordinates, the data needs mapping onto a skeletal mesh inside the game for body tracking inside unreal to function.

Chapter 3 - Motion Capturing

3.0 Introduction

This chapter includes how the motion capture from the Microsoft SDK was mapped onto the skeletal meshes and adjusted for UnReal engine. Using an external plug-in on UnReal, the program helps by creating a function labelled 'get socket location'. The function returns data to be able to manipulate the skeletal meshes, with minor adjustments.

3.1 Mapping Data to Skeletal Meshes

Getting motion tracking into a playable character in Unreal is more complex than copying and pasting the given Microsoft SDK software onto a mesh. Unreal Engine uses Blueprints (BP), which act as different code files. The work for this project was mainly actor BP's, which act as coding files for sprites in the game. Communication between these BPs is challenging but can be done using the 'Cast' function, which copies that BP's data and makes a copy of that file for use in the current BP. Finally, each BP has three main ways of running code; Code run on startup, code run during every tick/frame and code run on exit. Code can run on multiple other events like collisions, but this was used sparingly in the project.

3.1.0 Unique Created Functions and Initial Setup

The first setup requirement in getting skeletal data in UnReal Engine is setting up a body to map the data onto its sockets [16]. A model of a body is created in game, and then sockets can be added as rotation points. Fortunately, Unreal provides an example model which will be used throughout this project. Body mapping can now commence, with few changes to socket locations. The Unreal body has 50+ sockets, but as required for Kinect motion tracking, only 25 were used [14].

Correcting the body's position in UnReal Engine is required during the main execution of code. A simple function was created that offsets the original position of the body to match the ground plane (see Fig 3.1.0 in the appendix for more detail). The function takes the location and rotation of the player using the NeoKinect Plug-ins [16] and applies an offset, correcting the body's position.

3.1.1 Event Begin

In UnReal Engine, resource management is a crucial aspect in ensuring tasks are not running in the background unnecessarily. Therefore, the Kinect is initialised at the start of every game and uninitialised at the end. Initialisation of the body is also essential before starting the game (see Fig 3.1.1 in the appendix for more detail). Initialization includes toggling the skeletal mesh so it is visible in the game, and a transform to set it to an idle position (T pose in this case) is applied so that relative transforms can be applied later.

Next, using the NeoKinect Function [16], the skeletal data is retrieved from the Kinect by using the "get bodies" function. Note that this returns data for up to 6 bodies, but as we want to make this a simple game, we only access the first body in the array, setting up the body for the event tick.

Finally, the skeletal mesh data is cast to the animation BP which preps the mesh by allowing changes to be made to each joint location. Then, the data is added to an external variable that will be manipulated during the event tick cycle.

3.1.2 Event Tick

The event tick function is where most of the code is, as it constantly updates the mesh with the coordinates of where the player's position is. The first task is checking if the Kinect is on and functioning correctly (see Fig 3.1.2 and 3.1.3 in the appendix for more detail). Next, the compensation function is applied, the joint information is retrieved, and the variable "Bodies" is set. The first body is retrieved in the array of 6, and the Kinect again checks if a body is tracked before starting the main loop to change each socket location and orientation. The loop only commences if a trackable body is found as the function is memory intensive, so we want to limit use.

In the loop, the joint information from the animation BP array is retrieved to be updated with the player's socket rotation/ location. A NeoKinect function is used that breaks up the player data of each socket into rotation and location. The only interest is the rotation of each joint, as this is how the body is manipulated to mitigate joints moving away from the body. The loop filters through each joint in the array in the order from above [14] and copies the rotation into each relative element so that player movement is now copied onto the meshes array of joints. The function constantly ticks every 0.025 seconds (40 Hz) to produce a player tracked body.

3.2 Conclusion

With a trackable body set up as a controller, games are ready to implement and use the function for balance control by casting the actor BP into each separate game file.

Chapter 4 - Application of Motion Capture

4.0 Introduction

Once motion capture has been taken care of, simple games were created to enforce balance training with a more enjoyable experience. These games only train a few of the target muscles discussed earlier but act as an example of what could quickly be done with the game in the future.

4.1 Game 1 (Pose Holding)

The basis of the game is a simple pose-holding game. First, the player faces a pose on the screen that they must replicate to pass on to the next pose. The game awaits for the player to "hit" the pose for the countdown to start. Once held, the "yoga instructor" on the screen will glow green, indicating the players in the correct position. Opposingly the instructor will glow red if a change in posture is required. Once the countdown is finished, the instructor switches position and again waits for the player to "hit" the pose. The instructor changes pose five times during the whole game, and a final score is based on how long each pose was successfully held.

Fig 4.1.1, 4.1.2, 4.1.3 In game display during game one

4.1.0 Muscle Targeting

The game currently has five poses that can be held. Implementing new poses is a simple process, so more could easily be added, training even more targeting muscles. The poses consist of the following:

- Mountain pose + raised leg (raised right leg and raised left leg)
- Chair Pose
- Sideways lifted leg pose (raised right leg and raised left leg)

The mountain pose + raised leg helps train the gluteus medius and general core muscles [18]. The gluteus medius muscles help with general stability while walking, running and standing on one leg [19]. In addition, the targeted core muscles and the gluteus medius muscles help keep good posture and upper body stability [19]. The same muscles are trained during the sideways lifted leg pose with similar benefits.

Finally, the chair pose increases the strength of pelvic floor muscles [6]. Pelvic floor muscles control core stability and posture [20]. Good posture helps reduce the risk of falling, and all the muscles discussed should increase stability over a long period.

4.1.1 Unique Created Functions

Extra functions were created for repetitive use in the main bulk of the code. A function to compare both the instructor's and the player's joint positions, outputting a boolean value on whether all joints match. The function takes two input arrays of both the meshes. Comparisons between each joint check if the player's joint position is in the instructor's joint position range by ±15, only returning true if all joints match (see Fig 4.1.2 in the appendix for more detail).

4.1.2 Event Begin

"Event Begin" is a small part of the setup but the most important. It starts by toggling the visibility of the first pose so that it comes into view for the player (see Fig 4.1.0 in the appendix for more detail). The game works by having all the available poses on top of one another, and once started, only the pose required is displayed. The player position is required for the event tick, so during the "Event Begin", the position is copied from the class made in the motion capture script.

4.1.3 Event Tick

The event tick runs off many boolean variables to check if specific actions have taken place. The first check is if the player has matched the position on the screen or if the pose has been "hit" (see Fig 4.1.0 in the appendix for more detail). Nothing starts until the position is hit. The game is constantly waiting for a matched pose.

On pose "hit", a counter starts; the counter is in charge of how long the pose is on screen. On each tick of the circuit, the counter is increased by one until it reaches 2,000, where it changes the boolean variable, telling the script to reset all values (except score) and toggle the visibility of the current and next pose. Next, the score is calculated using the position check function created earlier. When a boolean value of true is outputted, the score increases by one on each tick; false adds nothing to the score. With all variables except the score, the program repeats four more poses until reaching the end, where a score out of 10,000 (2,000 ticks per cycle times 5) is displayed for the player before returning to the main menu after 10 seconds.

4.2 Game 2 (Fruit Catching)

This game aims to catch as many falling "fruits" with your hands as possible. The game starts by asking the player to raise either the left or right leg to get into a raised leg pose with both arms out to the side. The gameplay then starts by randomly dropping the "fruit" anywhere between the left or right of the screen. Then, with either leg still raised, the player catches as many fruits as possible without dropping any. The game can end in two ways; the player misses more than three fruits or the player places their foot back on the floor. A score of how many caught fruits is displayed and the cause of the fault, before returning to the main menu.

Fig 4.2.0, 4.2.1 In game display during game two

4.2.0 Targeting Muscles

The game only requires one pose for the player; however, the lateral movement for the player on one leg significantly increases the balance compared with just holding a stationary pose.

As said earlier, this game uses the raised leg pose, which trains the gluteus medius and core [18]. However, the significant benefit compared to the others is the increased torso movement leading to a stronger core, helping posture drastically and balance.

4.2.1 Unique Created Functions

Again, specific functions repeated multiple times were required. The first being a position-checking function that takes the player's joint position for either the left or right foot (see Fig 4.2.0 in the appendix for more detail). A boolean value of true is outputted if either of the feet is in the range -110 and -70, which has the foot raised just enough to balance on one foot.

The other function detects if the player's hand collides with the fruit. It takes the inputs of the fruit's world coordinates and the player's joint array. The function takes the player's left-hand and right-hand positions and checks if they are in the range of the fruit's x-coordinates by ± 10 (see Fig 4.2.3 in the appendix for more detail). The first boolean variable is outputted and put in an AND gate with the sphere falling to the correct z-coordinate at -10 to 10, which is about shoulder level. The function returns true if the sphere meets both of these conditions.

4.2.2 Event Begin

Once again, the event begins as a small part of the script, and the only command executed is retrieving the joint data from the motion capture software created earlier, done in the same way as in game 1.

4.2.3 Event Tick

Firstly, the script uses the position check function on every tick to ensure the player is raising a leg, starting once raised. Boolean values are used to check if the leg is lowered or the counter has reached three misses, showing a true value which will start the game's end game script (see Fig 4.2.1 in the appendix for more detail). The endgame script consists of showing the final score of how much fruit was caught, using the boolean values from earlier to display the cause of the fault.

If, instead, the boolean values don't trigger the endgame script, the main game script is started (see Fig 4.2.2 in the appendix for more detail). First, the fruit is checked to see if it is visible in the viewport and can choose two paths based on this information.

1. If the fruit is not visible, the visibility is toggled. A random integer value is used as an x-coordinate in the "set world location" function, giving the fruit a random lateral position.
2. If the fruit is visible, the fruit is passed through a "set world offset" function with a z-coordinate set to -0.5, triggering the fruit to fall slowly with every tick. The player is passed through the fruit collision function made earlier after each world offset to see if a collision occurs. If a collision occurs, a score of 1 is added, and the fruit's visibility is toggled, resetting the fruit's position. If a collision is not detected, the script keeps trying every tick until the fruit reaches a z-coordinate of -70, in which a miss is added and the visibility of the fruit is toggled again.

The script ensures that only two outcomes can occur. A score is either added or a miss is added to avoid confusion in the script, leading to both values increasing in one go.

4.3 Conclusion

As talked about previously, these games were developed as a concept. Once the initial motion capture code is created, games to train other muscles can be easily implemented using Unreal. One obvious limitation of Kinect is that it only captures front-body movement using a single camera, limiting the exercises that can be accomplished. However, there should still be plenty of options to train all the muscles discussed in the theoretical background [5, 6].

Chapter 5 - Validating the Kinect Motion Capture

5.0 Introduction

Valid motion capture data is one of the main goals of the project. Motion capture validation is crucial to ensure the form of the positions is held correctly to reduce the chances of injury. Currently, the problem is that only trainers can aid in the correct form, but if the motion capture analysis proves accurate, the need for a trainer is redundant. The Kinect motion capture data was validated using the University of Leeds 8 camera Vicon setup for accurate motion capture.

5.1 Data Collection

The Microsoft Kinect setup was straightforward. Angle outputs for each joint were taken every 0.025 seconds (or every frame at 40 FPS) using the "get socket rotation" UnReal Engine function. The UnReal function takes each of the 25 joints set up to track [14], giving an X, Y and Z value. The data was exported as a CSV file and uploaded to Excel for further analysis.

The Vicon setup was more complicated but provided very accurate results. First, calibration using a wand with IR lights helped set up the simulation's correct camera position and orientation. Correct calibration reduces the likelihood of markers displaying multiple times in the simulation. Vicon uses a plug-in gait model for body tracking, which needs 37 IR markers placed in specific positions [21]. Aligning the markers on the body is crucial as the plug-in gait model uses them to create the model of the player. Furthermore, the system requires accurate body measurements to adjust the simulated joint locations, making them unique to each player. A 200 Hz tick rate was chosen so that every 5th bit of data could be taken to be compared with the Kinect motion capture, effectively creating a matching 40 Hz tick speed so that the Kinect and the Vicon system could be aligned properly.

To record both the motion capture systems simultaneously, the Kinect was set up in the Vicon motion capture lab facing the wall. A T-pose position was chosen to start as a way of helping data alignment later on. The motion chosen to capture consisted of; a T pose, touching the nose with both hands and squatting. The choice of movement was based on common yoga positions for improving balance and movements on which the Kinect showed weakness.

5.2 Analysis

Once data was collected and imported into a spreadsheet, data processing and offsets needed to be applied to align the data correctly. Analysis using RMSE values and calculating the percentage of time each joint was within ±15 degrees was calculated and reviewed.

5.2.0 Data Cleaning

The data imported needed processing before any analysis was performed. Firstly, UnReal Engine and Vicons coordinate systems are slightly different. The Z-axis is the same, but the X and Y-axis need switching. However, when switching, the Unreal x coordinate system becomes the negative Vicon y coordinate system, so a flip of the data was alos required.

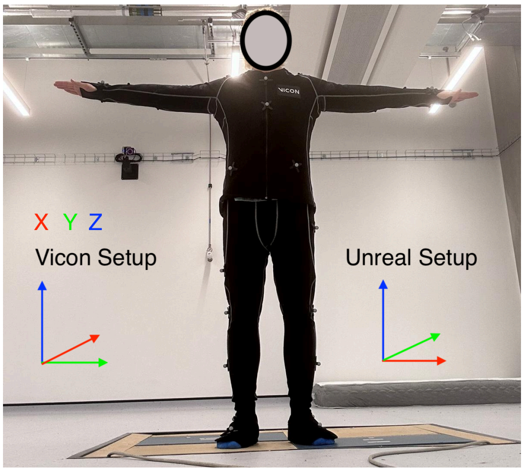


Fig 5.2.0 Kinect Coordinate System [14] Fig 5.2.1 Coordinate System Comparison

The data ranged from -180 to +180 degrees. An offset of +180 degrees was added to all the data to change the system from 0 to 360 degrees for easier data handling. Working with only positive values in Excel makes it easier to calculate the RMSE values without changing the relationship of the data.

Graphs of each separate joint were made, and they showed that there were slight offset distances between some of the data points. Therefore, separate small offsets were introduced so that the data fit accurately and better represented the relationship in motion capture validation. During the implementation of the small offsets, only multiples of 90 were used to keep the data unbiased. The reason for the data to need these offsets could be slight differences in the coordinate systems relative to the world or that the joint directions are different. After applying the offsets, the data showed more evident signs of correlation from one motion capture system to the other.

5.2.1 Outliers and Data Discrepancies

The processed data still showed outliers and data errors between both systems, as expected when handling large quantities of data. The Vicon system was relatively smooth and showed accurate data except for two joints. Both elbows y and z data (UnReals x and z data, Fig 5.1, 5.2) did not record anything and output 0 for each joint. This could have been because the system may not have been set up correctly or the downloaded plug-in gait model was set up incorrectly. With these "messy" results, RMSE values will be taken with and without elbow and ankle data.

Similarly, the Unreal data showed many discrepancies. Some joints produced seemingly random data that did not show fluid movement. The Elbow and wrists X and were sporadic, as shown in the figures below.

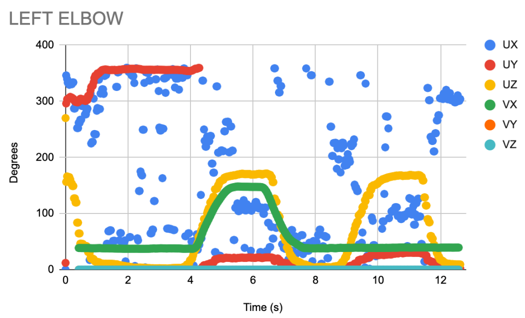


Fig 5.1 Left elbow angle against Time (s)

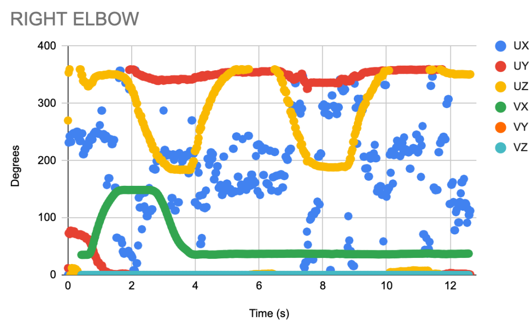


Fig 5.2 Right elbow angle against Time (s)

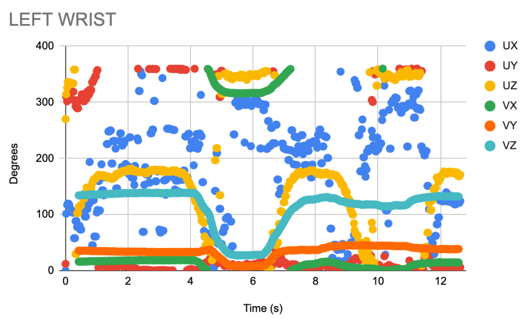


Fig 5.3 Left Wrist angle against Time (s)

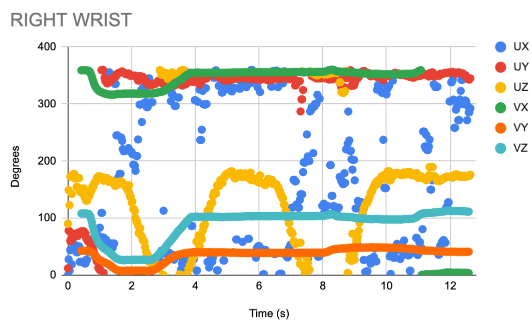


Fig 5.4 Right Wrist angle against Time (s)

As shown, there is no correlation between these data points. Discrepancies could be down to bugs in the given function on Unreal Engine. However, it is more likely that Kinect only sees the front plane (Y) and struggles to pick up values in the

X. Therefore, these data points will be ignored entirely for data validation. Building on this, data from Unreal Engine only partially lined up and had a few bugs, which caused double readings.

The elbow's Z and Y data were also mixed. The table above shows clear signs that there was no correlation between the elbows UnReal and Vicon data. Vicons only readable data for the elbows show accurately that each elbow is flexed once on the X plane (Unreal Y plane). However, the UnReal data for the elbows showed differently. UnReal picked up movement in the Z plane despite no movement on the plane. Therefore, data from the Z axis should be ignored as it showed two large movements. The data discrepancy could be a bug in the angle function as shown before, leading to these repeated motions. Similarly, the wrist data showed the same discrepancy as it moves with the elbow.

5.2.2 Calculations and Discussion

Two measurements were taken to compare and validate the motion capture system. The first is the RMSE values between the joints X, Y and Z coordinate systems. The second is the percentage of time the joints were within ±15 degrees of each other. The following calculations were used:

$$RMSE = \sqrt{\frac{\Sigma(O_i - E_i)^2}{n}} \quad (1) \qquad Time\% = \frac{no.within \pm 15}{n} \times 100 \quad (2)$$

n	Total number of data points
O_i	Observed Value (Unreal Values)
E_i	Expected Values (Vicon Values)
$no.\ within\ \pm 15$	Count of values that lie in Vicons $\pm$15 degree range

Using the RMSE equation (1), The following results were recorded:

	Full Body (Deg)	Upper Body (Deg)	Lower Body (Deg)
X	60.82	68.88	52.76
Y	36.08	30.01	42.15
Z	58.05	64.58	51.51

X, Y, Z	51.18	54.49	47.89

Table 5.0: RMSE results including the elbow and ankle data

The same results were calculated, excluding the ankle and elbow data:

	Full Body (Deg)	Upper Body (Deg)	Lower Body (Deg)
X	48.11	56.73	39.48
Y	24.10	17.08	31.11
Z	50.40	54.38	46.42
X, Y, Z	40.24	42.73	37.75

Table 5.1: RMSE results excluding the elbow and ankle data

Looking at the data above, it is clear that the UnReal motion capture could be more accurate. However, focusing on just the Y plane, the data shows a higher correlation than the X and Z plane. The lower correlations could be due to the Kinect only seeing the front of the body, so motion captured in the Y plane is easy to pick up. While inaccurate, the X and Z data can still be manipulated to show accuracy at specific points. Excluding the data taken from the ankle and the elbow, a clear drop in the RMSE value shows, confirming our earlier prediction that the ankle and elbow data would skew the results.

Results also showed that, in general, looking at our most accurate plane (Y Plane), the upper body showed the most significant results compared with the lower body. The difference in results is expected as most upper body movements are along the front plane so that the Kinect can pick them up easily. Compare this with lower body movement, showing nearly double the RMSE value in the Y plane, which could be because there is less front plane movement, and most of the movement from the knees is occurring towards the camera. However, considering all planes, the lower body is more accurate. The Kinect data in the X and Z axis is less accurate, so conclusions will not be made for these planes.

To consolidate this data using equation (2), the following results were obtained:

	Full Body (%)	Upper Body (%)	Lower Body (%)
X	36.75	31.30	42.16
Y	41.82	60.28	23.36
Z	24.53	31.32	17.72

X, Y, Z	34.37	40.97	27.75

Table 5.2: % in range results including the elbow and ankle data

The same results were also obtained, excluding the elbow and ankle data:

	Full Body (%)	Upper Body (%)	Lower Body (%)
X	45.11	39.41	50.83
Y	62.75	77.71	31.16
Z	24.59	27.57	21.60
X, Y, Z	44.15	48.22	40.07

Table 5.3: % in range results excluding the elbow and ankle data

Looking first at the data that includes the elbow and ankle data, the most accurate result is the upper body Y data, with 60% of the data falling within ±15 degrees of the Vicon data set. Compare this to its X and Z counterparts which are around half the accuracy of the Y plane, consolidating that the Y plane is the most accurate.

When the elbow and ankle are not included, the upper body Y data is the most accurate at around 77%. The least inaccurate data points were those recorded in the Z and X axes. However, the X's lower body results unexpectedly showed better accuracy than the upper body, which could be down to less movement in the lower body front plane, so results should be more constant.

5.3 Conclusion

The Kinect motion capture is relatively good, but only with movement along the front plane. The Kinect X and Z data was inaccurate to the point where it should not be used to help train people, as further injury from malpractice could occur. However, the Y plane's accuracy was strong enough, with RMSE values as low as 17%, so that it could aid training. The major downfall of the Kinect motion capture system is picking up movements in the legs, mainly due to the difficulties of picking up motion towards the single camera. The data processed required much cleaning to compare with the Vicon system, with many unsolved discrepancies. Further investigation should be taken using a more comprehensive range of motion to improve results. While the validation showed the Kinect's difficulties with picking up specific movements, it is still an excellent option considering the size and cost, as most people will need access to Vicon facilities.

Chapter 6 - Post Game Development Analysis

6.0 Introduction

The project's other main goal was to ensure the games were enjoyable, engaging, and affordable. Therefore, a focus group of ten people with a frailty score of 1 or 2 using the rockwood frailty scale[22], were asked to respond to open questions based on playing the game.

6.1 Survey Overview

All ten participants agreed to play the games over a week and answer the questions anonymously. The Questions were as follows:

1. How enjoyable did you find the experience?
2. How engaging did you find the experience?
3. Did you find the games simple and easy to understand? If not, what were the confusing aspects of the games?
4. Do you feel your motion was captured effectively?
5. Do you feel your confidence with balance has increased over the week period?
6. What are your thoughts on incorporating this game into your balance training routine?
7. Are you likely to spend £45 or more on a device that trains you in a similar way to this?
8. Is there anything you would like to see in the future for this game?

It is expected that each person should not feel any confidence change in balance over the seven days due to the short period they played the games. However, the purpose of the questionnaire is not to prove that the games increase balance but to provide an engaging and enjoyable experience for the user.

6.2 Results

The results came back as expected. The objective of the focus group was to show that users found the experience generally enjoyable and engaging. 9/10 users had positive answers surrounding the enjoyability and engageability of the games (see Fig 6.2.0 in the appendix for more detail). Only one focus group member had a bad experience playing the games, suggesting that future work should introduce more exciting games than current ones. The consensus was that the games were primarily enjoyable and mostly engaging, but diversity in games is required, showing that the product can be used in homes to help train in the near future.

To build on this, 7/10 users said they would incorporate this into a balance training routine (see Fig 6.2.1 in the appendix for more detail), further consolidating that healthcare could use this method as a prevention technique for a lower cost compared to current rehabilitation . The questionnaire asked if they would pay £45 for it (the cost of the Kinect and adapter [23, 24]) to which 6/10 replied saying they would, and two were on the fence about whether they would or not. The lower-cost method of using a Kinect could be an ideal solution for preventing falls. The price did not change the mind of anyone's opinion on whether they would purchase it.

The users were asked whether they found the games simple to understand. Due to the general demographic that the device is catered to, simplicity is crucial as the elderly population generally has a low grasp on how to use technology. Although anonymous, no group members were over 65, so it is assumed they were more technologically competent. However, all ten results returned positive, saying there were no problems with understanding both games and that the whole experience was simple, showing that it is more likely that an elderly user would be able to use this as little understanding is required for the games to function.

The users were also asked how effectively they felt their motion was captured. Most users said something like "for the most part", with two users specifying that the squat position was problematic. Referring back to the validation results, this was expected as the lower body was much less accurate than the upper body. However, it further consolidates that the motion capture is primarily accurate but should be reviewed in future studies.

Question 5 asked users whether they felt their balance had increased over the seven days. As expected, no one felt their confidence increased drastically over the seven days. Only three people said they felt increased confidence, but only minorly. The short period should not have made any users feel any more confident with their balance. A more extended and more data-intensive survey should be conducted to get a definitive answer to this question in future work.

Finally, users were asked if they would use this as part of a balance training routine. Only one user said they would not, showing that the project succeeded in getting people to train their stability muscles. With further research and a more finalised program and device, the Kinect could be a part of fall prevention in the future.

Conclusion

7.0 List of Achievements

- The research identified the most common fall types and muscle groups that affect balance.
- A unique motion capture program was created, containing two games that support common exercises to train muscle groups that are shown to improve balance.
- Identified off-the-shelf products - Xbox Kinect - to ensure the project deliverables were cheap and easily accessible to the public.
- Validated that motion capture programs using Xbox Kinect were accurate enough for specific muscle training.
- Positive feedback from the focus group indicated that users are willing to engage with the project deliverables consistently.

7.1 Discussion

In chapter two, research showed that the leading cause of hip injuries is a sideways fall, with the elderly at the most risk of permanent damage [4]. In addition, research showed that targeting the following stability muscles pelvic floor muscles, transversus abdominis, multifidus, obliques, rectus abdominis, erector spinae, gluteus medius and the diaphragm[5, 6] increased balance performance. Furthermore, Yoga and tai chi [7,8] showed the best results for training these muscle groups, and solutions should focus on holding yoga positions like the mountain pose and chair pose.

Chapter 3 outlined the pros and cons of different motion capture tools. The Xbox Kinect captures 25 joint locations and angles [14] every frame (30 FPS) [10] for use in external programs, but the single camera causes some discrepancies, which limited positions. However, compared to other motion capture methods, its low-price point meant validating and testing the motion capture viability was more practical than using the Vicon system whilst supporting the project goal of affordability and accessibility for the general public. Furthermore, the joint locations of the Kinect supported skeletal mesh mapping with Unreal Engine through a third-party plug-in which acted as a great starting point for the application's development.

Chapter 4 introduced simple balance games based on yoga poses that targeted stability muscles. These simple and quick games helped the project flow and user engagement and could be a base for future game development and muscle groups.

Chapter 5 investigated the Kinect's motion capture capability and validated that a player's position could be tracked. The results showed adequate system accuracy for tracking the upper body, with an RMSE value of around 17. However, results also showed that lower body accuracy is inadequate and may lead to further injury.

Finally, Chapter 6 observed a game-playing focus group. In summary, the group felt the games were engaging and enjoyable. In addition, most users said they were happy to pay what was spent on the project, indicating that the Kinect is an affordable method of motion capture tracking for balance training. However, the group agreed that the solution could be improved with greater game diversity.

7.2 Conclusions

By using off-the-shelf tools, this project shows that independent developers can create serious games to train people to strengthen different muscle groups and improve their balance, reducing their risk of falling sideways. While the focus group did not meet the demographic for the program, results indicated that it was simple enough that the elderly should understand how to use these tools and strengthen different muscle groups. Furthermore, users had an enjoyable experience that should improve engagement and long-term sustainability despite the slight inaccuracy of motion capture. The significant drawback is that motion capture provided average results compared to its high-tech counterpart. However, the Kinect should still be considered for future health projects due to the small cost and ease of access. In short, the Kinect motion capture program successfully created games that train balance at a low cost, engage users, and decrease the likelihood of an elderly person falling.

7.3 Future Work

The significant project drawback was the lower body motion capture inaccuracy due to user movements - generally - towards the camera. Future work should focus on designing a more accurate motion capture system. For example, accuracy improvements could be designed using a dual camera setup or modifying a Kinect with two cameras at a fixed distance apart. Developing two cameras like this will capture the majority of the side profiles of the player, so the movement towards the camera will be calculated more accurately. Furthermore, motion capture accuracy could be improved through a more extensive motion capture validation technique, mitigating the unexpected results produced by the X and Z-axis.

The project highlighted the need to work more closely with the target demographic (over 65s) and incorporate their experience and capabilities into our development work. It is recommended that any future work takes notice of the time for ethical approval to allow for a sufficient length of study with more in-depth data recorded. While research supports that the games are linked to decreasing chances of a sideways fall, further research should focus on first-hand data to support that the games aid balance control.

Finally, future work should consider the idea of minigames to supplement the initial two-game approach taken in this project. Notably, user feedback in the surveys suggested providing a greater variety of exercises to create engagement and sustainability. Although this project introduced the concept of games to help users, extending the functionality with Unreal Engine to incorporate other muscle groups would be straightforward.

References

1. World Health Organization, WHO. Falls. [Online]. 2021. [Accessed 01/12/2022]. Available from: https://www.who.int/news-room/fact-sheets/detail/falls
2. Centre for Disease Control and Prevention, CDC. Hip Fractures. [Online]. 2016 [Accessed 01/12/2022]. Available from: https://www.cdc.gov/falls/hip-fractures.html#:~:text=As%20the%20U.S.%20population%20gets,are%20hospitalized%20for%20hip%20fractures.&text=More%20than%2095%25%20of%20hip,2%20usually%20by%20falling%20sideways.&text=Women%20experience%20three-quarters%20of%20all%20hip%20fractures.
3. Crenshaw, J.R., et al., *The circumstances, orientations, and impact locations of falls in community-dwelling older women.* Arch Gerontol Geriatr, 2017. **73**: p. 240-247.
4. American Hip Institute & Orthapedic Specialist. How Falls Can Injure Your Hips. [Online] 2019 [Accessed 06/12/2022]. Available from: https://www.americanhipinstitute.com/blog/how-falls-can-injure-your-hips-17450.html#:~:text=Did%20you%20know%20that%2095,different%20types%20of%20hip%20injuries.
5. Paige Waeher, CPT. Stabilizer Muscles Used in Exercise and for Balance. [Online]. 19/04/2021 [Accessed 04/12/2022]. Available from: https://www.verywellfit.com/stabilizer-muscles-1230990#toc-your-stabilizer-muscles
6. Set for Set. What are Stabilizer Muscles & How to Strengthen Them?. [Online]. 17/03/2022 [Accessed 05/12/2022]. Available from: https://www.setforset.com/blogs/news/how-to-strengthen-stabilizer-muscles#:~:text=The%20major%20muscles%20used%20for,longissimus%20thoracis%2C%20and%20the%20diaphragm.
7. Jen Jewell, JJ, Breaking Muscle. Use Yoga to Improve Your Lifts. [Online]. 2017 [Accessed 05/12/2022]. Available from: https://breakingmuscle.com/use-yoga-to-improve-your-lifts/#:~:text=In%20yoga%2C%20you%20constantly%20challenge,will%20be%20in%20the%20gym.
8. JJ HONG TAI CHI & QI GONG MISSISSAUGA. Tai Chi for Back Pain Prevention and Relief. [Online]. 2017 [Accessed 05/12/22]. Available from: https://www.taichi.ca/2017/01/05/tai-chi-for-back-pain-prevention-relief/#:~:text=Tai%20Chi%20is%20a%20Core%20Work%20Out%20for%20the%20Stabili

zer%20Muscles&text=Unlike%20power%20muscles%2C%20core%20%2F%20stabilizer,as%20a%20group%20of%20muscles.

9. Ni, M., et al., *Muscle utilization patterns vary by skill levels of the practitioners across specific yoga poses (asanas).* Complementary Therapies in Medicine, 2014. **22**(4): p. 662-669.
10. J. Jiao, L. Yuan, W. Tang, Z. Deng, and Q. Wu, 'A Post-Rectification Approach of Depth Images of Kinect v2 for 3D Reconstruction of Indoor Scenes', ISPRS International Journal of Geo-Information, vol. 6, p. 349, 11 2017.
11. O. Wasenmüller and D. Stricker, 'Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision', 11 2016.
12. R. Lun and W. Zhao, 'A Survey of Applications and Human Motion Recognition with Microsoft Kinect', International Journal of Pattern Recognition and Artificial Intelligence, vol. 29, p. 150330235202000, 03 2015.
13. J. MacCormick (2012). How Does a Kinect Work? [PowerPoint slides]. Available: https://pages.cs.wisc.edu/~ahmad/kinect.pdf
14. L. Jamhoury, "Understanding Kinect v2 joints and coordinate system," Medium, 21-Apr-2022. [Online]. Available: https://lisajamhoury.medium.com/understanding-kinect-v2-joints-and-coordinate-system-4f4b90b9df16. [Accessed: 02-May-2023].
15. G. Kurillo, E. Hemingway, M.-L. Cheng, and L. Cheng, 'Evaluating the Accuracy of the Azure Kinect and Kinect v2', Sensors (Basel), vol. 22, no. 7, Mar. 2022.
16. UnrealEngine, "Skeletal mesh sockets," Unreal Engine 4.27 Documentation. [Online]. Available: https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Types/SkeletalMeshes/Sockets/. [Accessed: 02-May-2023].
17. R. Villani, "NeoKinect Quick Start Guide," rvillani, UnrealEngine, 2018. [PDF]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiS-dr_2Kn-AhWSOcAKHfOgASIQFnoECAoQAQ&url=http%3A%2F%2Ffiles.rvillani.com%2Fneokinect%2FNeoKinect-QuickStart.pdf&usg=AOvVaw2OaA4bPz2pCR6IFNkOzZaJ [Accessed: 02/05/23].
18. Willett, G. M. (2019). Anatomy trains: Myofascial meridians for manual and movement therapists. Elsevier Health Sciences.
19. Juker, D., McGill, S., & Kropf, P. (1998). Quantitative intramuscular myoelectric activity of lumbar portions of psoas and the abdominal wall during

a wide variety of tasks. Medicine and Science in Sports and Exercise, 30(2), 301-310.
20. Bø, K. (2004). Pelvic floor muscle training is effective in treatment of female stress urinary incontinence, but how does it work?. International urogynecology journal, 15(2), 76-84.
21. Nexus, "Plug-In Gait Reference Guide" Vicon, 2016. [PDF]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjUvoWixcr-AhWBWcAKHTS9DZYQFnoECA4QAQ&url=https%3A%2F%2Fdocs.vicon.com%2Fdownload%2Fattachments%2F133828966%2FPlug-in%2520Gait%2520Reference%2520Guide.pdf%3Fversion%3D2%26modificationDate%3D1637681079000%26api%3Dv2&usg=AOvVaw1hr8-fT14aRf_3enq1RtTR [Accessed: 02/05/2023].
22. “Clinical frailty scale,” Dalhousie University, 2005. [Online]. Available: https://www.dal.ca/sites/gmr/our-tools/clinical-frailty-scale.html. [Accessed: 02-May-2023].
23. CEX, “Xbox One official Kinect 2 sensor - CEX (UK): - buy, sell, donate,” Xbox One Official Kinect 2 Sensor. [Online]. Available: https://uk.webuy.com/product-detail/?id=SXB1OFFKIN2&sku=SXB1OFFKIN2. [Accessed: 02-May-2023].
24. Amazon, “Kinect power adapter for Xbox One X/s and windows 8/8.1/10 PC with UK plug,” Amazon.co.uk: PC & Video Games. [Online]. Available: https://www.amazon.co.uk/Kinect-Power-Adapter-Xbox-Windows/dp/B08K7FYZ3X/ref=asc_df_B08K7FYZ3X/?tag=googshopuk-21&linkCode=df0&hvadid=463009873159&hvpos=&hvnetw=g&hvrand=2236491013954174438&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1006864&hvtargid=pla-992842455730&psc=1. [Accessed: 02-May-2023].

Appendix

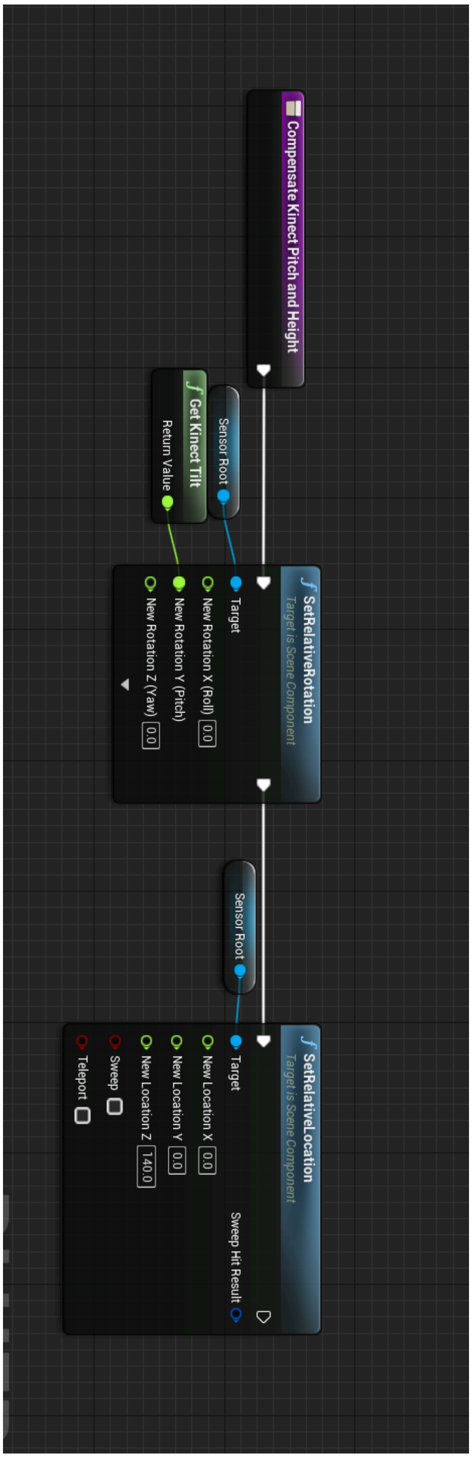


Fig 3.1.0 Kinect Compensation

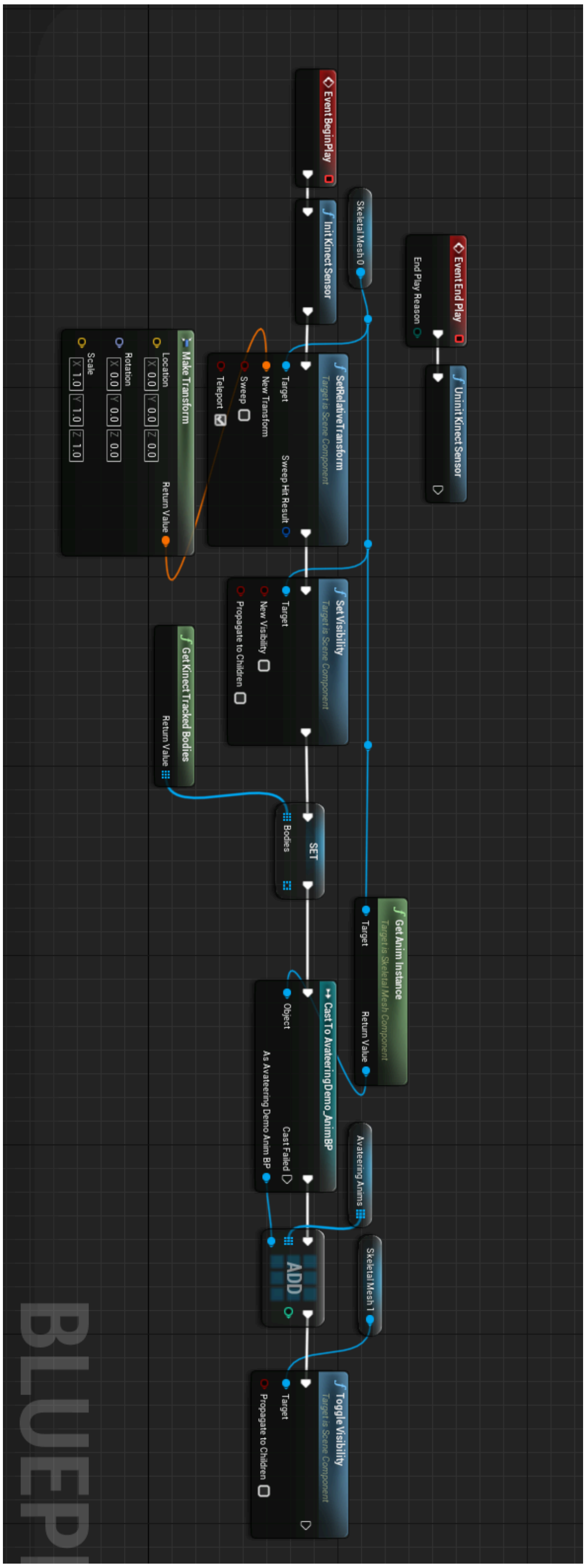
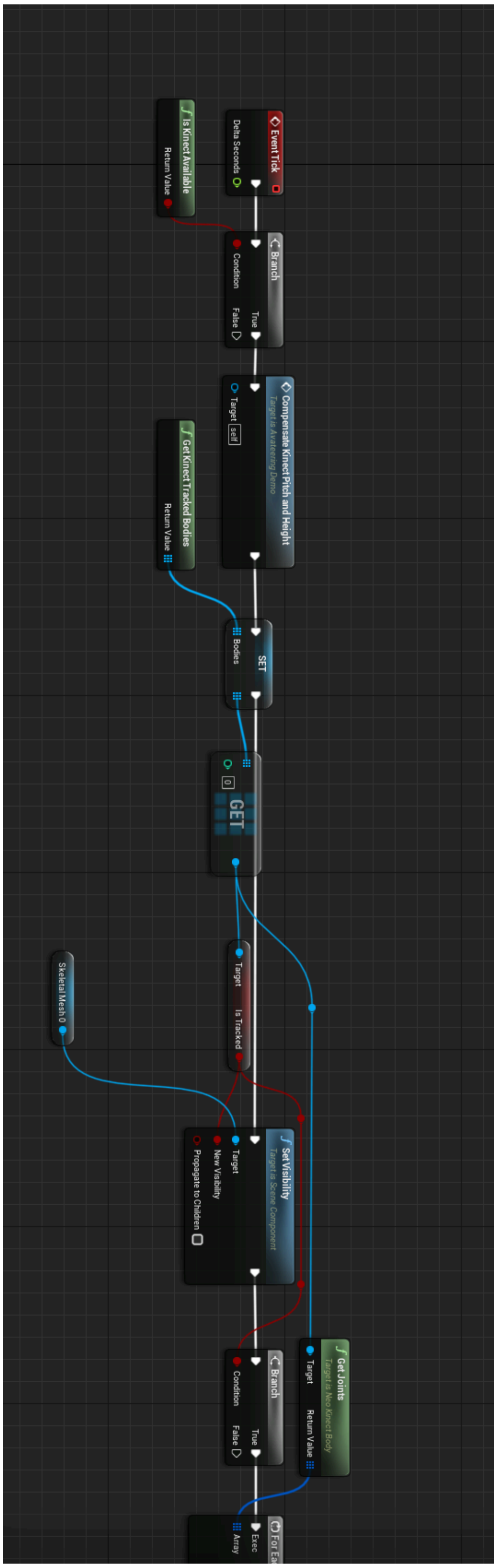


Fig 3.1.1 Motion Capture Event Begin



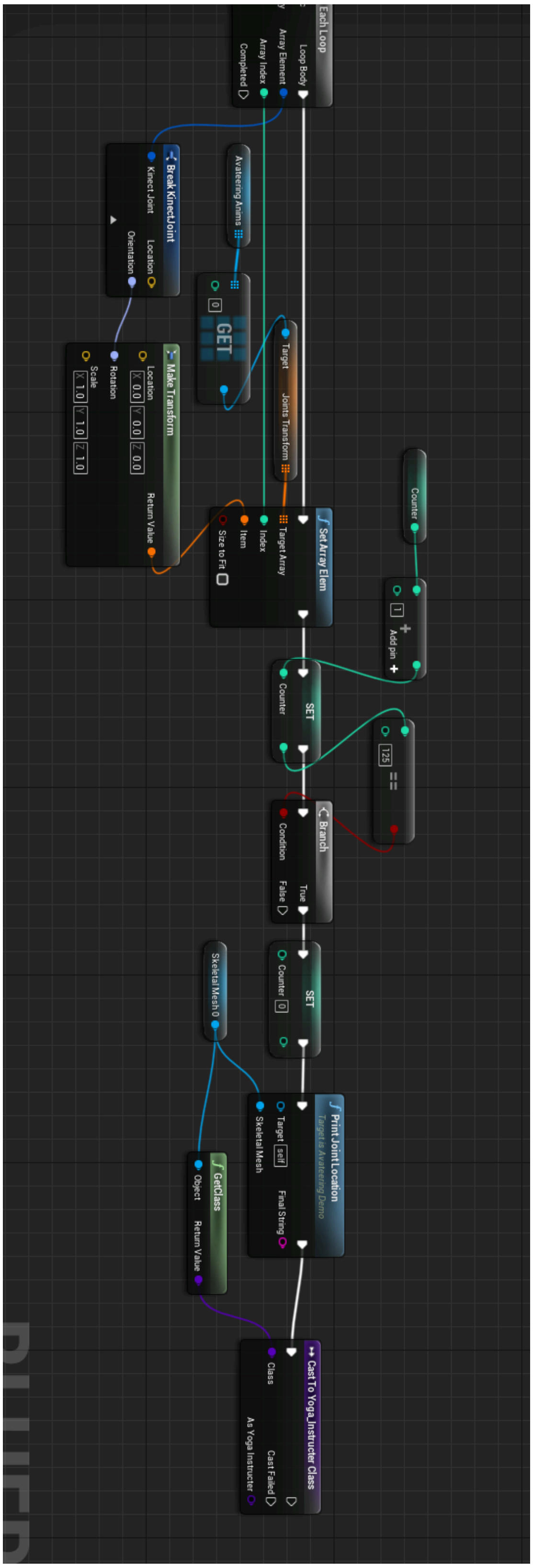
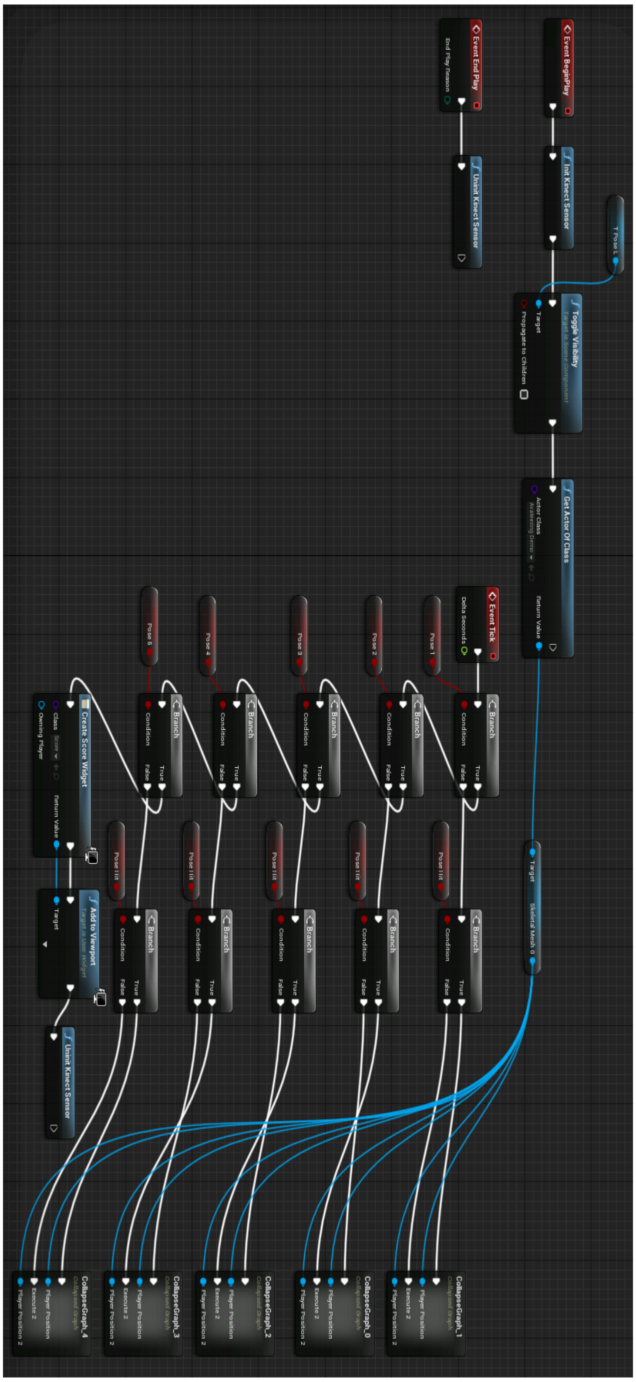


Fig 3.1.2 Motion Capture Event TIck

Fig 3.1.3 Motion Capture Event Tick Followed



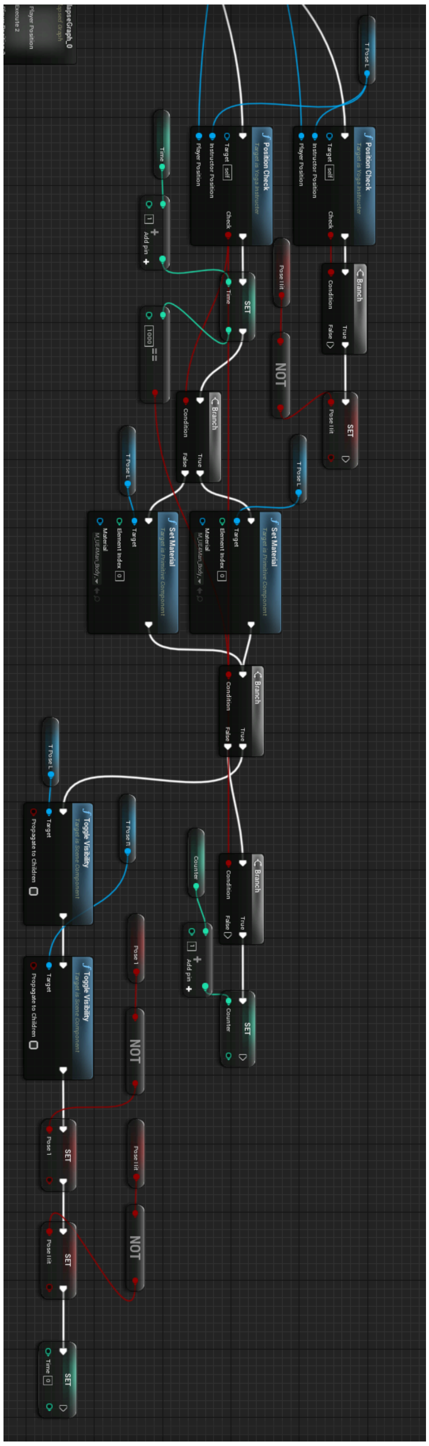


Fig 4.1.0 Game 1 Outline of Script

Fig 4.1.1 Game 1 In Depth Function Review

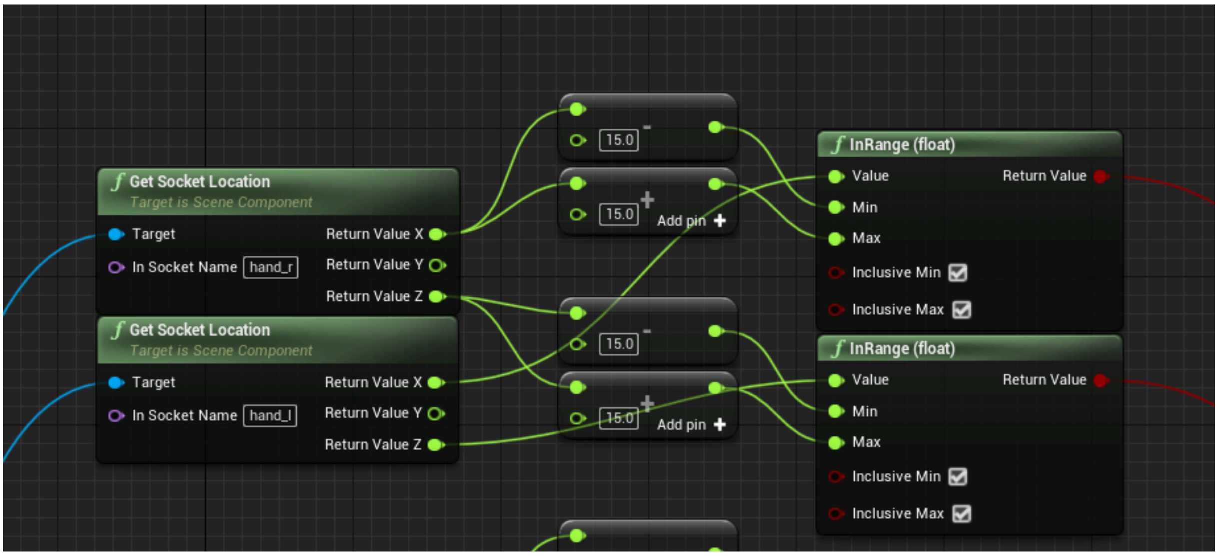


Fig 4.1.2 Game 1 Position Validation

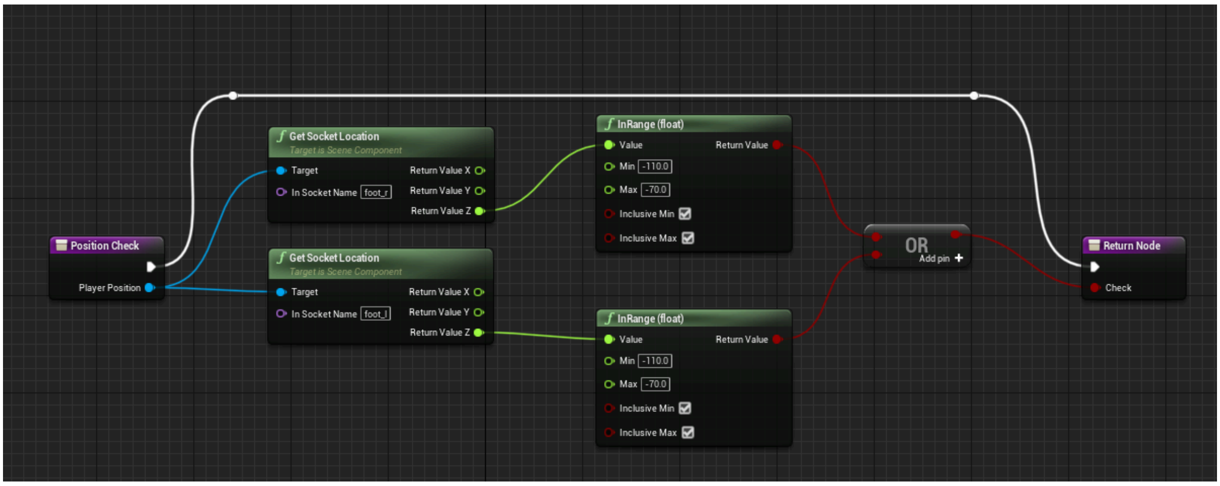


Fig 4.2.0 Game 2 Position Validation

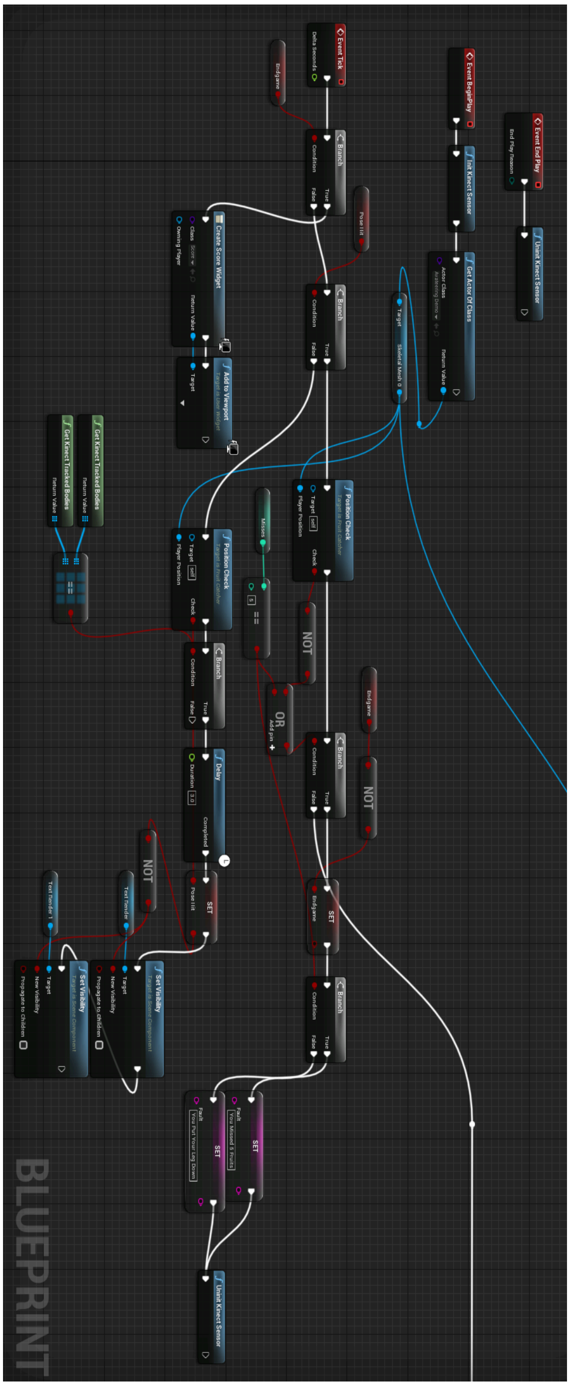


Fig 4.2.1 Game 2 Initial Game Setup

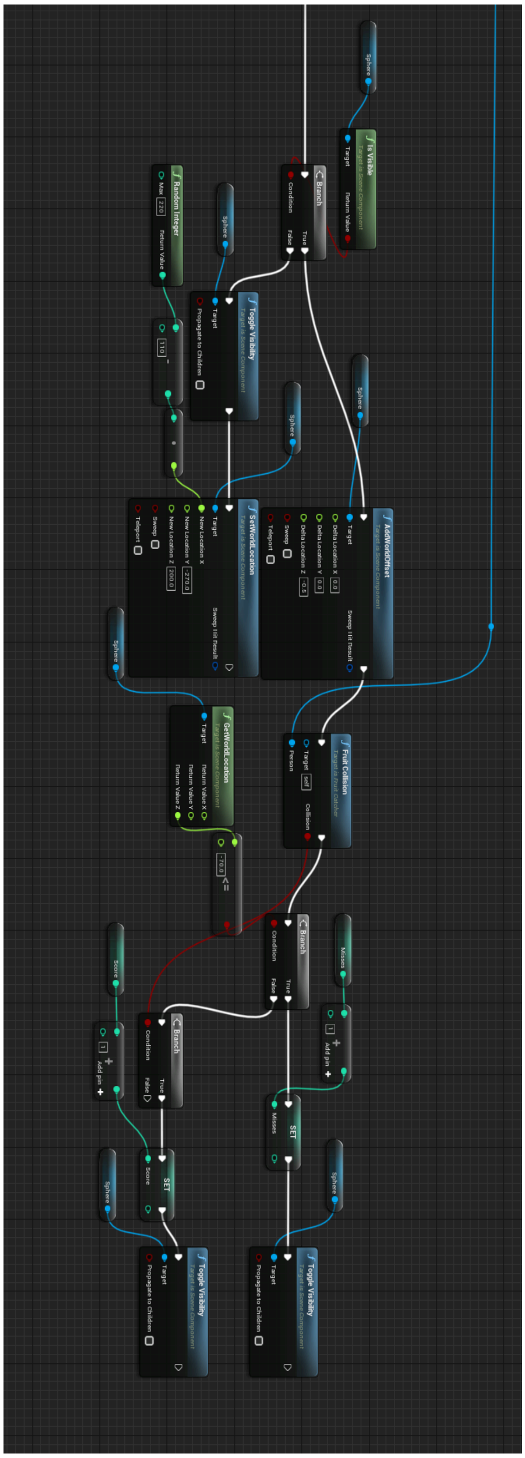


Fig 4.2.2 Game 2 Fruit Movement

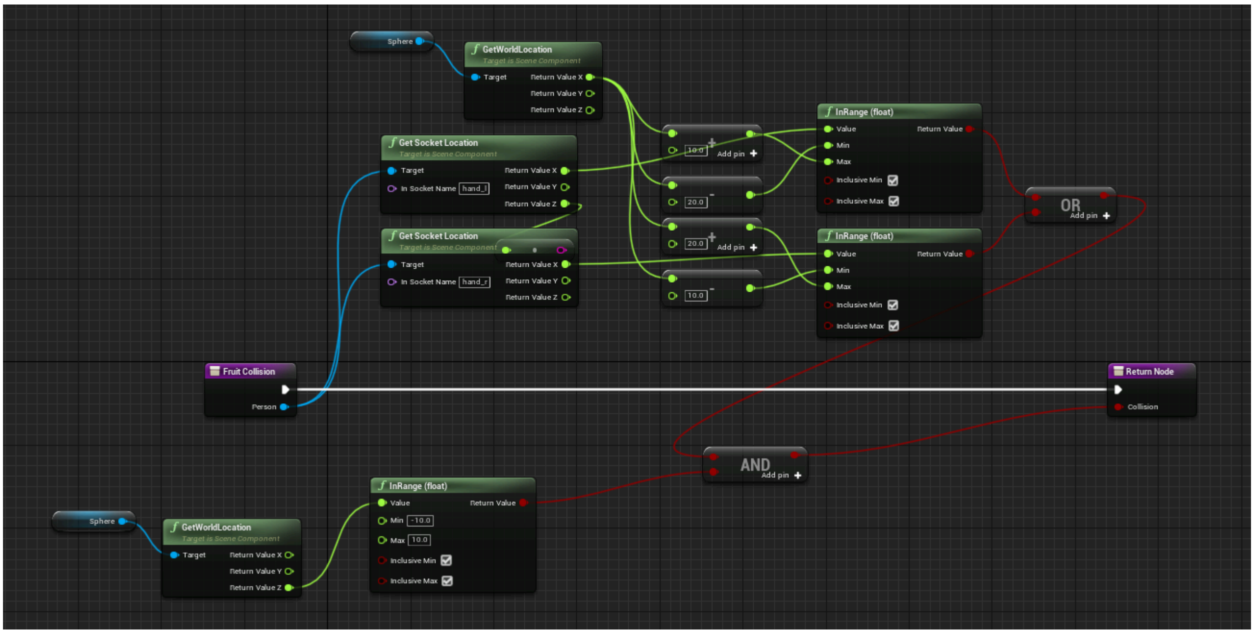


Fig 4.2.3 Game 2 Fruit Collision

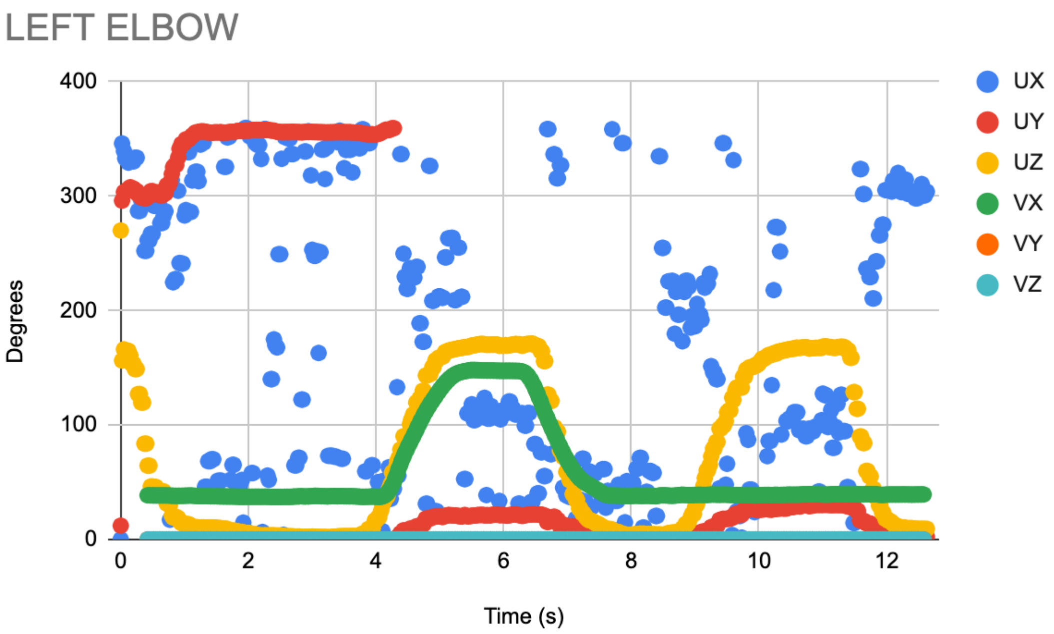


Fig 5.1 Left elbow angle against Time (s)

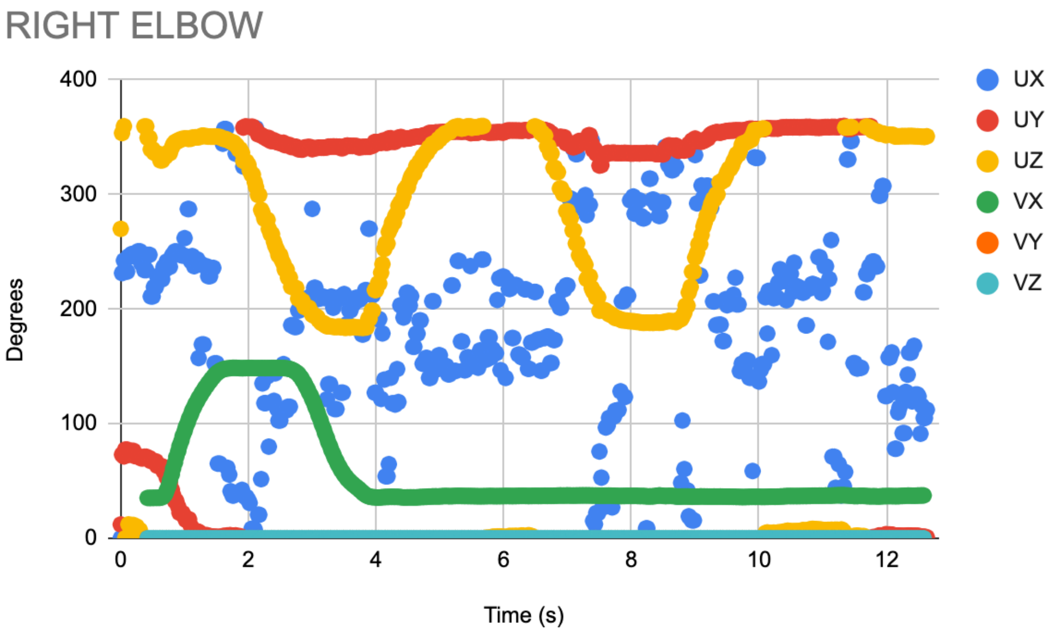


Fig 5.2 Right elbow angle against Time (s)

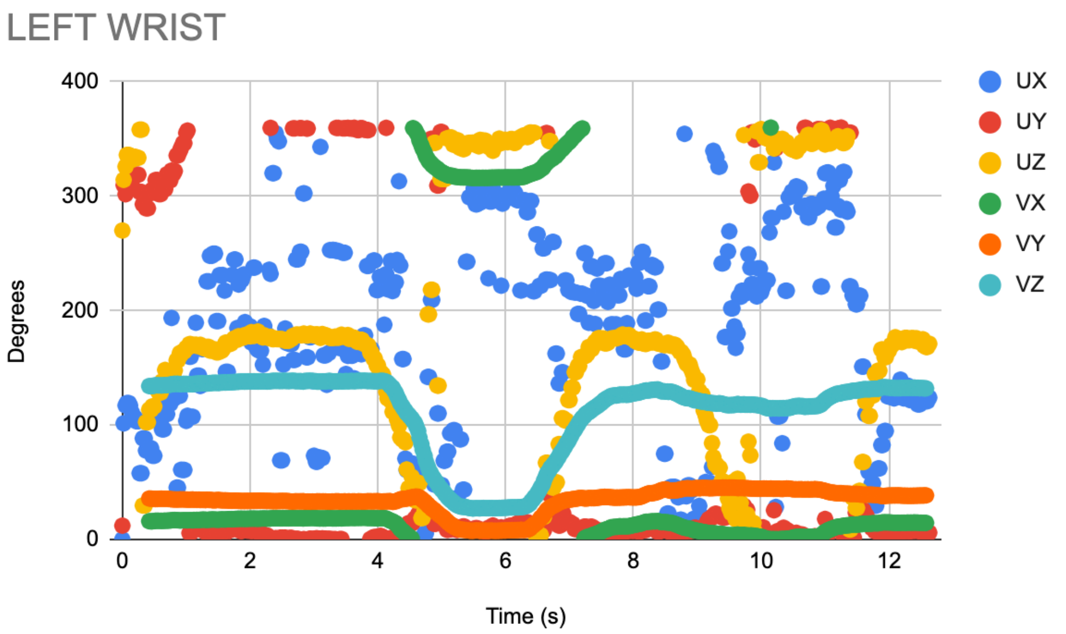


Fig 5.3 Left wrist angle against Time (s)

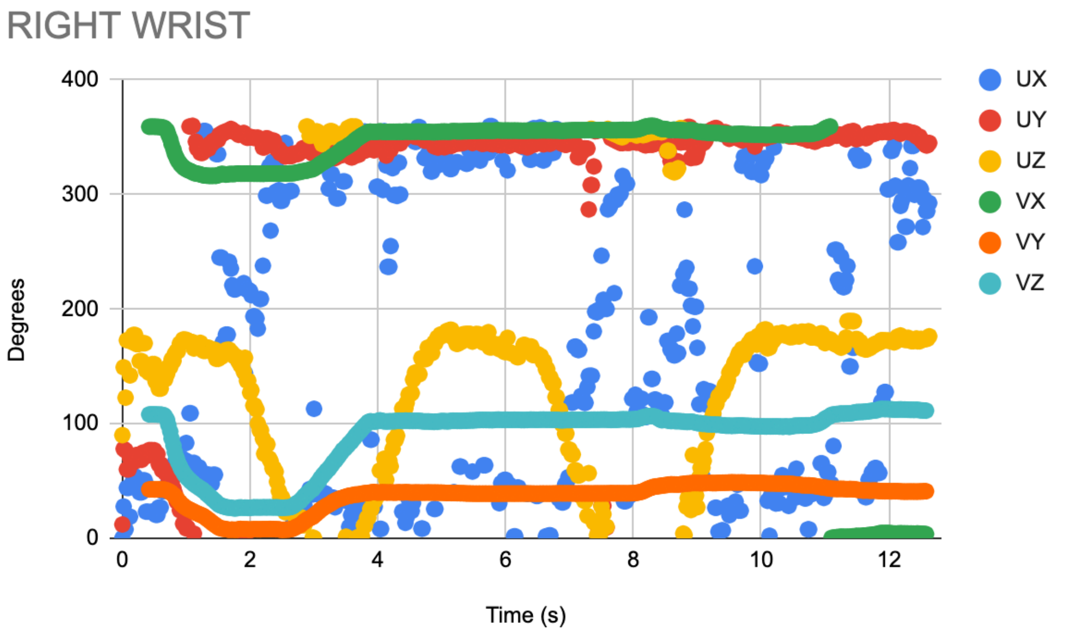


Fig 5.4 Right wrist angle against Time (s)

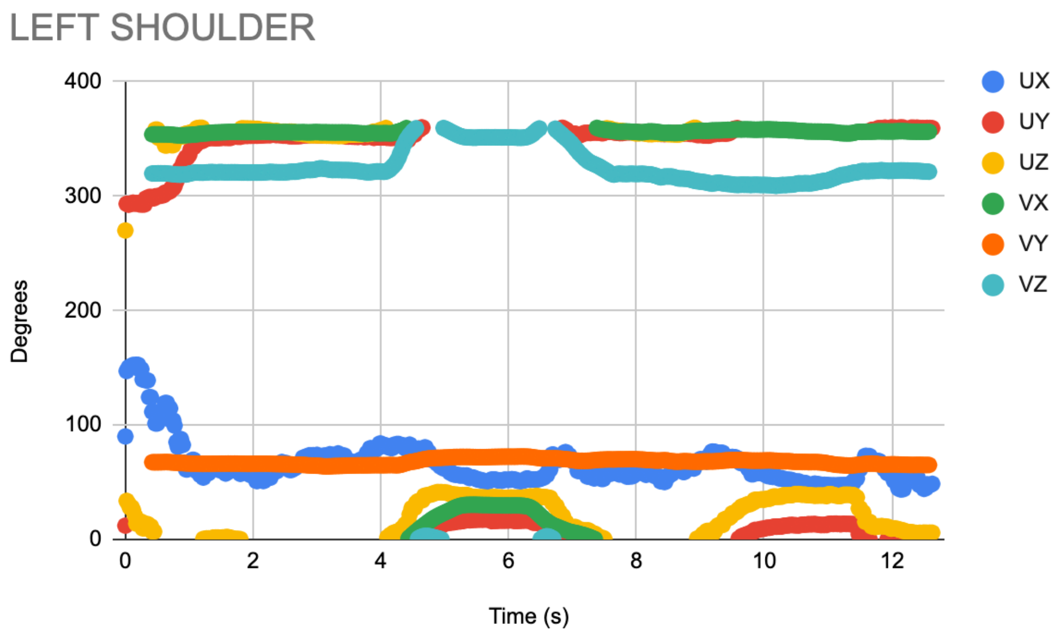


Fig 5.5 Left shoulder angle against Time (s)

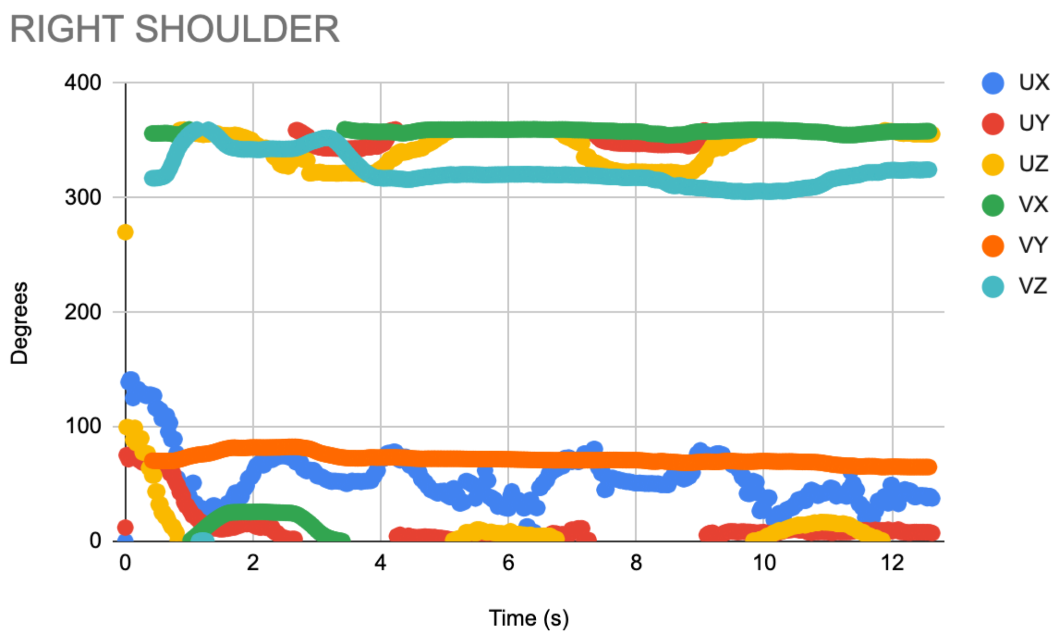


Fig 5.6 Right shoulder angle against Time (s)

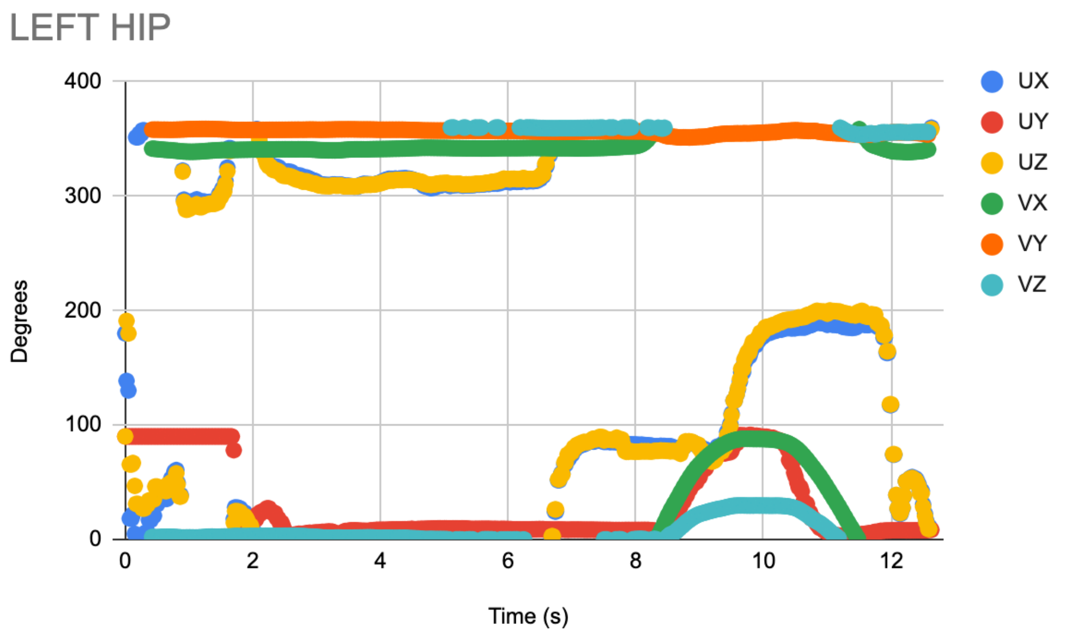


Fig 5.7 Left hip angle against Time (s)

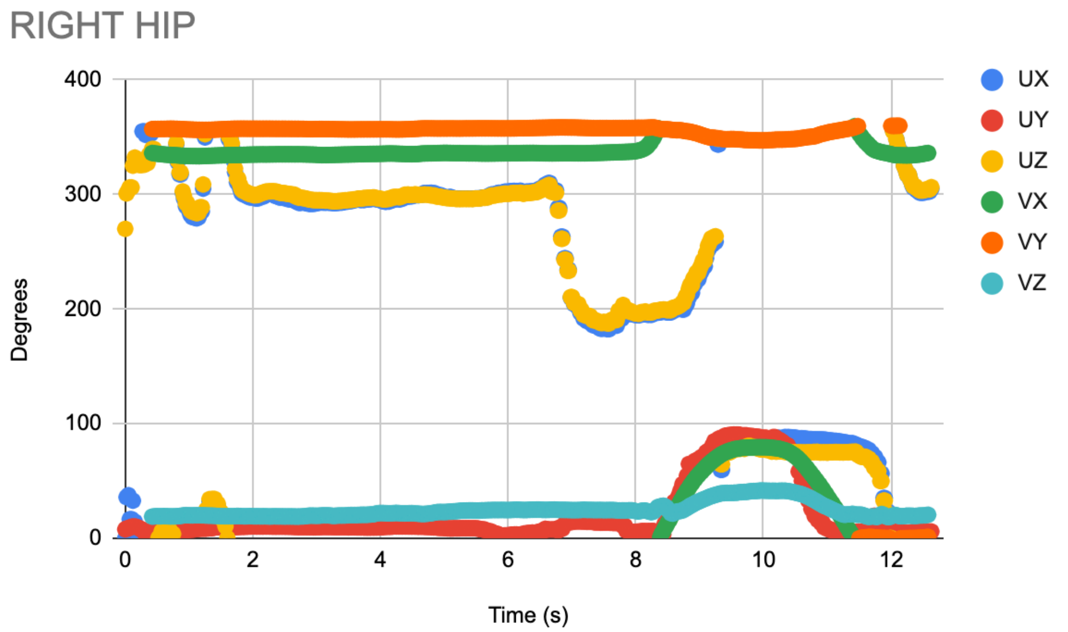


Fig 5.8 Right hip angle against Time (s)

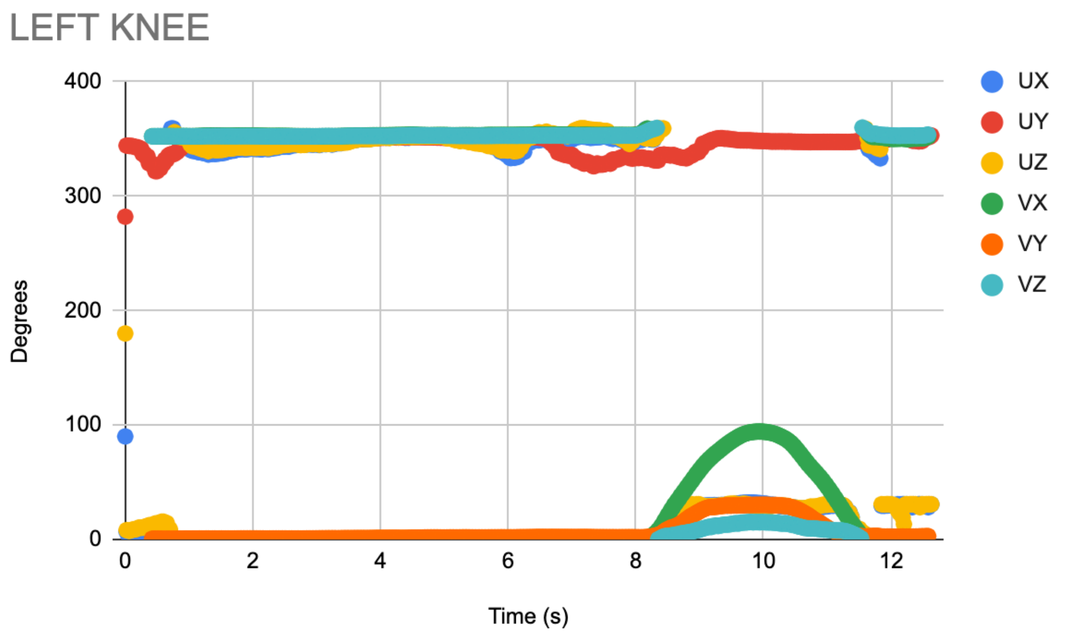


Fig 5.9 Left knee angle against Time (s)

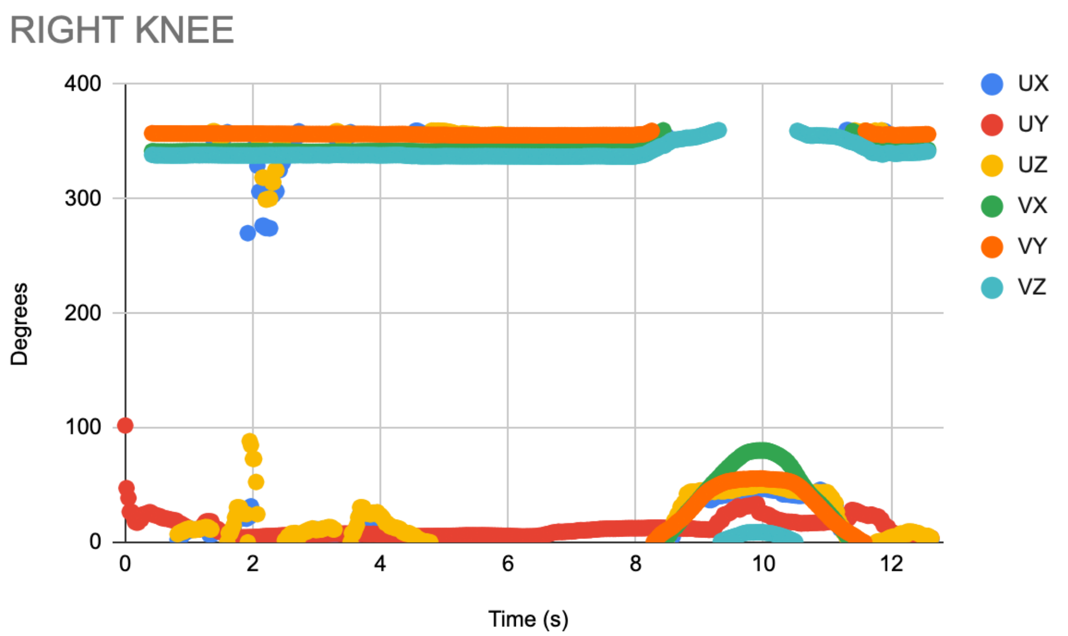


Fig 5.10 Right knee angle against Time (s)

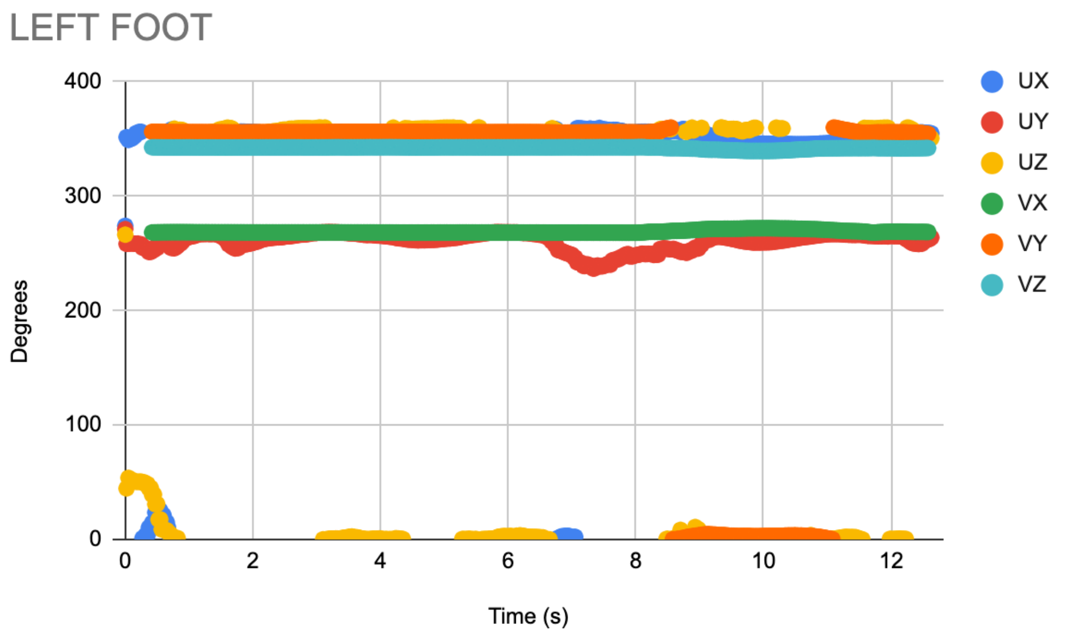


Fig 5.11 Left foot angle against Time (s)

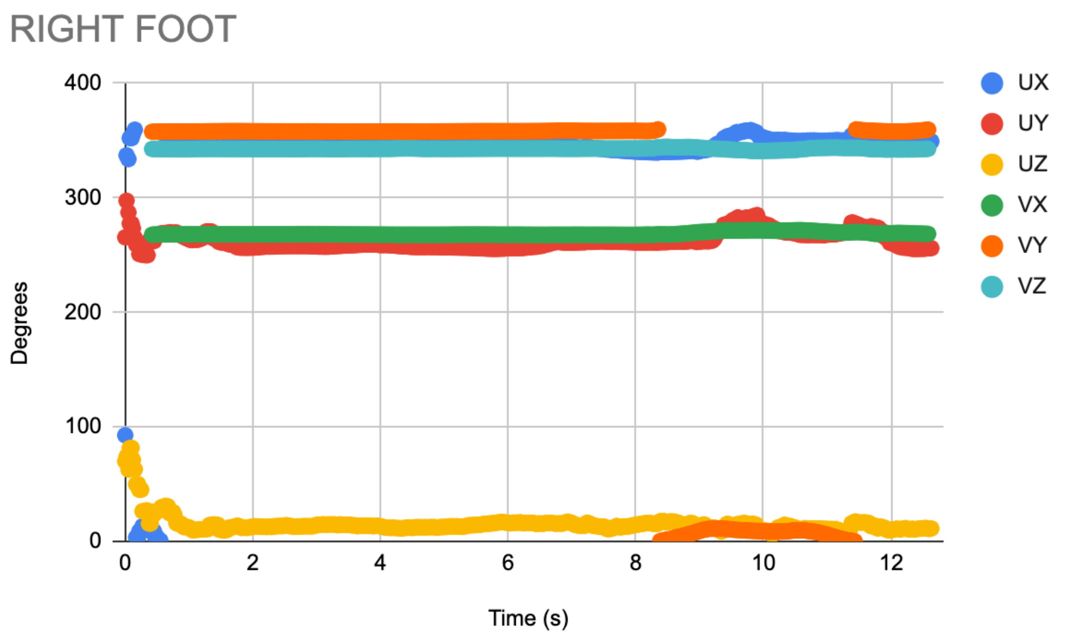


Fig 5.12 Right foot angle against Time (s)

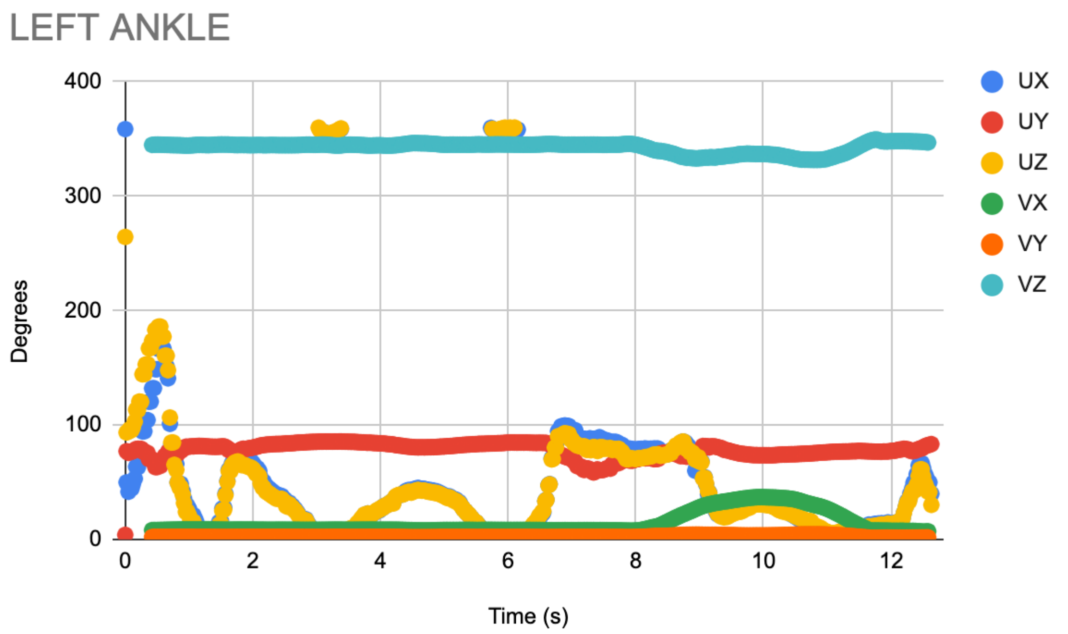


Fig 5.13 Left ankle angle against Time (s)

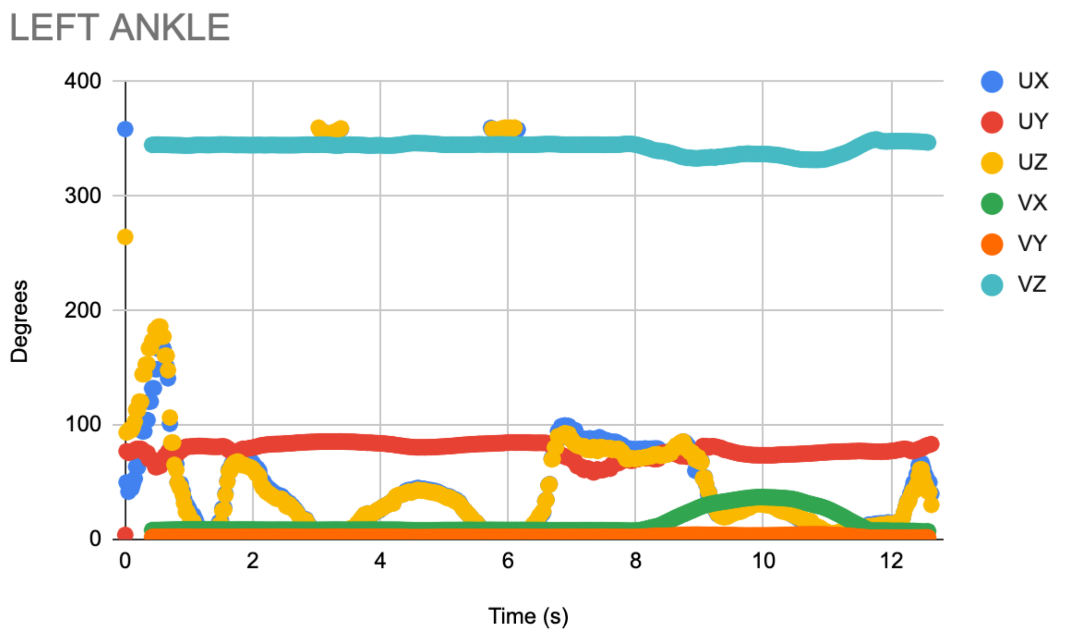


Fig 5.14 Right ankle angle against Time (s)

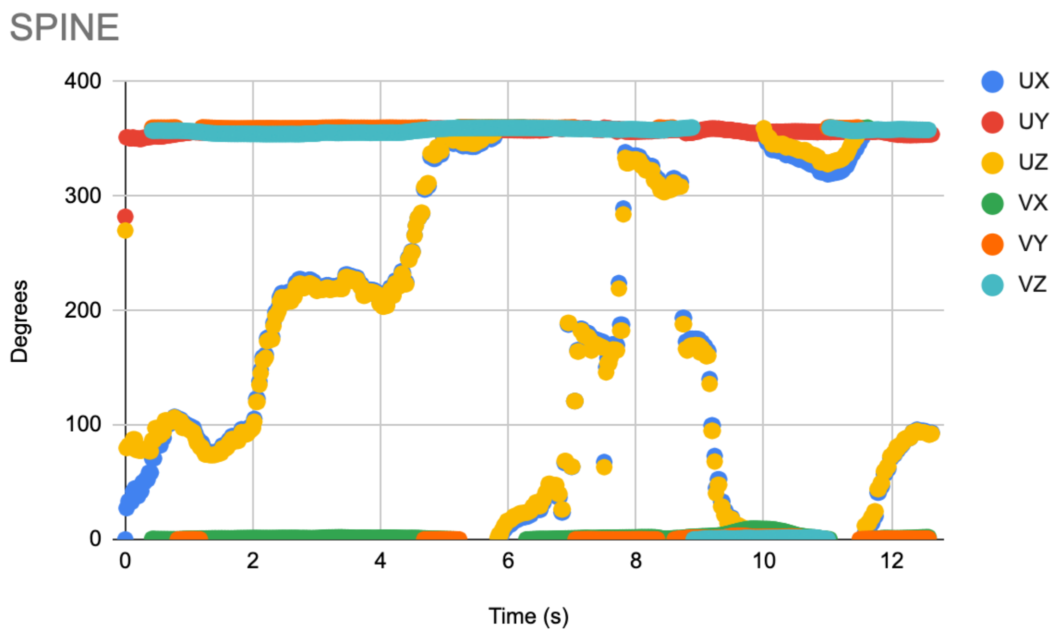


Fig 5.15 Spine angle against Time (s)

Person	Question 1	Question 2	Question 3	Question 4
1	I thought it was relatively enjoyable and fun to use	Very engaging but I wouldn't do this often	Yes they were very easy to understand	Mostly, with the exception of when squatting during one of the exercises the game character had funny movements
2	Not enjoyable	Very engaging	They were simple to understand	For the most part however some positions i was in didn't respond well in game
3	Enjoyable for the most part	Very engaging	Very simple	Somewhat well with the exception of a few game plays
4	Very enjoyable	Pretty engaging and entertaining	Very easy to understand	Very well
5	Quite enjoyable	Not engaging	The games were super easy to understand	Yes my motions were captured fine
6	Somewhat enjoyable but found the games to be repetitive	Mostly engaging	Yes very simple	For the most part
7	I found it very enjoyable	Mostly engaging	Very simple and easy even without instructions	Mostly yes
8	Pretty enjoyable	I was mostly engaged	The games were easy to understand	For the most part yes
9	Very enjoyable	I was engaged for the most part with a few days lacking motivation to play	Very easy to understand	For the most part yes
10	I found it pretty enjoyable, the games were interesting	I found it very engaging	The games were very simple to understand, I had no problems with them	For the most part my motion was captured well, however certain actions like squating showed problems

Fig 6.2.0 Questionnaire answers for questions one to four

Person	Question 5	Question 6	Question 7	Question 8
1	I think it’s probably unchanged since starting the game practice	I could probably include this in a routine	Yes most likely I could	Nothing
2	No	It’s unlikely I would	Probably not	Not really, maybe more mini games
3	Not at all	I would consider incorporating it into a routine	Most likely yes	More mini games could make it more interesting
4	Yes a little bit	I would definitely incorporate it	Yes most likely	More mini games and better motion capture during certain tasks
5	Not really	I would consider it but unlikely	Most likely but unsure	More mini games
6	Not at all	Only if I knew it would benefit me	Probably not	More mini games
7	A little bit	I would definitely incorporate this in a balance routine	Yes definitely	More mini games
8	Maybe somewhat but not drastically	I would consider using it in a balance training routine	Maybe when the products finished	More mini games
9	Not that I can tell	I would definitely incorporate this into a routine	Yes pretty likely	More mini game options
10	No, I’m confidence is unchanged	I would be inclined to try incorporating this into a daily balance routine	Yes, I think £45 is very reasonable	More fun minigames

Fig 6.2.1 Questionnaire answers for questions five to eight