# INTRODUCING ROLLING AXIS INTO MOTION CONTROLLED GAMEPLAY AS A NEW DEGREE OF FREEDOM USING MICROSOFT KINECT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EVREN C. BOZGEYİKLİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF MODELING AND SIMULATION

AUGUST 2012

# INTRODUCING ROLLING AXIS INTO MOTION CONTROLLED GAMEPLAY AS A NEW DEGREE OF FREEDOM USING MICROSOFT KINECT

Submitted by **EVREN C. BOZGEYİKLİ** in partial fulfillment of the requirements for the degree of **Master of Science in Game Technologies, Middle East Technical University** by,

Prof. Dr. Nazife Baykal
Director, **Informatics Institute**

Assist. Prof. Dr. Hüseyin Hacıhabiboğlu
Head of Department, **Modeling and Simulation**

Assoc. Prof. Dr. Veysi İşler
Supervisor, **Computer Engineering**

**Examining Committee Members:**

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineeering, METU

Assoc. Prof. Dr. Veysi İşler
Supervisor, Computer Engineering, METU

Dr. Umut Durak
Roketsan

Assist. Prof. Dr. Hüseyin Hacıhabiboğlu
Modeling and Simulation, METU

Dr. Erdal Yılmaz
Sobee Studios

**Date:** August 29, 2012

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.**

**Name, Last name : Evren C. Bozgeyikli**

**Signature :** ______________________

# ABSTRACT

## INTRODUCING ROLLING AXIS INTO MOTION CONTROLLED GAMEPLAY AS A NEW DEGREE OF FREEDOM USING MICROSOFT KINECT

Bozgeyikli, Evren C.
M.Sc., Department of Modeling and Simulation
Supervisor: Assoc. Prof. Dr. Veysi İşler

August 2012, 145 pages

Motion controlling is a rapidly improving area of game technologies. In the last few years, motion sensing devices for video games such as Nintendo Wii, Microsoft Kinect for Xbox 360 and Sony PlayStation Move have gained popularity among players with many compatible motion controlled games. Microsoft Kinect for Xbox 360 provides a controller free interaction system in which the player controls games by using only body movements. Although Kinect provides a natural way of interaction, rolling action of body joints are not recognized within the standard motion sensing scope of the tool. Aim of this thesis is to provide an improved gameplay system with an increased degree of freedom by introducing rolling axis of movement using Microsoft Kinect for Xbox 360 for motion sensing. This improved gameplay system provides the players a more natural and accurate way of motion controlled interaction, eliminating unnatural gestures that are needed to be memorized to compensate for lacking of the roll movement recognition.

Keywords: Touchless, Gesture, Motion Tracking, Kinect, Motion Controlled Video Games

# ÖZ

## YUVARLANMA EKSENİNİN YENİ BİR SERBESTLİK DERECESİ OLARAK HAREKETLİ OYUN KONTROLÜ KAPSAMINA MİCROSOFT KINECT KULLANILARAK DAHİL EDİLMESİ

Bozgeyikli, Evren C.
Yüksek Lisans, Modelleme ve Simülasyon Bölümü
Tez Yöneticisi: Yard. Doç. Dr. Veysi İşler

Ağustos 2012, 145 sayfa

Hareket algılama, oyun teknolojilerinin hızla gelişmekte olan alanlarından biridir. Geçtiğimiz yıllarda Nintendo Wii, Microsoft Kinect for Xbox 360 ve Sony PlayStation Move gibi hareket algılayan oyun konsolları, hareket ile kontrol edilen pekçok uyumlu oyunla birlikte kullanıcılar arasında oldukça popüler olmuştur. Microsoft Kinect kullanıcılara herhangi bir fiziksel kumanda cihazı kullanmaksızın yalnızca vücut hareketlerini kullanarak oyunları kontrol edebilecekleri bir etkileşim olanağı sağlamaktadır. Kinect kullanıcılara doğal bir etkileşim yöntemi sağlasa da, vücut eklemlerinin yuvarlanma eksenindeki hareketleri, cihazın mevcut hareket algılama kapsamı dahilinde algılanmamaktadır. Bu tezin amacı, yuvarlanma eksenindeki hareketlerin etkileşime dahil edilmesiyle serbestlik derecesinin arttırılarak, Microsoft Kinect ile iyileştirilmiş bir oynanış sistemi geliştirmektir. Geliştirilen etkileşim sistemi, kullanıcılara daha doğal ve hassas hareket kontrolü olanağı sağlamakta ve daha önce yuvarlanma eksenindeki hareketlerin algılanmamasından dolayı ortaya çıkan boşluğu doldurmak için tasarlanan, kullanıcların öğrenmesi gereken kontrol hareketlerini ortadan kaldırmaktadır.

Anahtar Kelimeler: Temassız, Hareket, Hareket Algılama, Kinect, Hareket Kontrollü Video Oyunları

To the meaning of my life, my wonderful girl, my wife; Gamze...

# TABLE OF CONTENTS

CHAPTER

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **2D** | Two Dimensional |
| **3D** | Three Dimensional |
| **API** | Application Programming Interface |
| **CEGEQ** | Core Elements of the Gaming Experience Questionnaire |
| $d_{Hn,1-Hn,2}$ | Distance Between $H_{n,1}$ and $H_{n,2}$ |
| $DJ_{RH}$ | Position of Right Hand Joint in Depth Coordinates |
| $d_o$ | Direction of the Object |
| **DOF** | Degree of Freedom |
| $d_{RH}$ | Depth of the Right Hand Joint |
| $DT_o$ | Position of Tip Point of the Object in Depth Coordinates |
| $E_1$ | First Edge Point of the Object |
| $E_2$ | Second Edge Point of the Object |
| **FPS** | Frames per Second |
| **H** | Edge Point of the Hand |
| **IR** | Infrared |
| **LED** | Light Emitting Diode |
| $M_o$ | Midpoint of the Object |
| **RGB** | Red-Green-Blue |
| **SDK** | Software Development Kit |
| **SMEQ** | Subjective Mental Effort Questionnaire |
| $T_o$ | Position of Tip Point of the Object in World Coordinates |
| $W_{E1}$ | Position of Object's First Edge Point in World Coordinates |
| $W_{E2}$ | Position of Object's Second Edge Point in World Coordinates |
| $W_{E1-E2}$ | 3D Vector Passing through $W_{E1}$ and $W_{E2}$ |
| $WJ_{RH}$ | Position of Right Hand Joint in 3D World Coordinates |
| $WJ_{RH-To}$ | 3D Vector Passing through $WJ_{RH}$ and $T_o$ |
| $\delta$ | Orientation of the Object |

# CHAPTER 1

# INTRODUCTION

Recent advancements in the field of game technologies enabled a wide variety of new areas to be explored by the researchers. Motion sensing technology for games is one of these intriguing areas that emerged with the release of Nintendo Wii [1] which offered the players a new way of interaction via performing movements with the handheld motion controller followed by Sony PlayStation Move [2] that enabled a similar way of interaction also performed with a handheld motion controller, and peaked with the release of Microsoft Kinect for Xbox 360 [3] which enabled the players control games with natural body movements without using any handheld physical controllers.

Sales figures indicate that Kinect attracted a lot of attention among players with its unique gameplay style. Giles point out the large number of sales as "Over a million people bought one of Microsoft's Kinect game controllers in the 10 days after its US launch on 4 November" [4]. The device also earned a Guinness World Record in the fastest selling gaming peripheral category with an average of 133,333 units sold per day in the first two months period after its release [5].

As the focus is turned to game development, full body motion controlling is a new way of interaction that can be considered to be inchoate currently and may benefit from research that will help building informative foundations as Moen states [6]. This young way of interaction which enables the players act as controllers with their body movements offers possible research opportunities in new areas to be improved and explored, towards providing a better gameplay experience for players.

Motion tracking algorithm of Microsoft Kinect for Xbox 360 is limited in understanding rolling axis of rotation. No velocity or rotation tracking is supported by the skeleton tracking pipeline of the device as it is stated in its official documentation [7]. Hence, rolling action of body joints are not considered in the standard motion sensing scope of the tool, limiting the variety of gestures used in gameplay and introducing some new gestures to be used rather than natural rolling movements in some cases, creating a possible problem of difficulty in learning for the players. People usually play video games to enjoy themselves, so requiring the player learn and memorize various unnatural command gestures is considered to be a repulsive way of game design. LaViola and Keefe state that gestural interaction is an effectual way of controlling, especially in video games with their endless potential uses [8]. The authors emphasize the downside of the approach as follows: "However, one problem with gestural interaction is that the user needs to learn all the gestures." stating additionally that the average user cannot remember more than seven gestures normally due to the memory limitations. They argue that applications' requiring large variety of gestures may create important problems, especially for the inexperienced users.

It is important to keep the interaction design as simple and user friendly as possible, which can be achieved by using systems that have accurate motion sensing capabilities, with minimum learning requirements imposed to the players offering natural ways of communication they already know how to perform instinctively rather than assigning them with new unfamiliar and unnatural gestures to be memorized.

In this thesis, it is aimed to provide an improved motion controlled gameplay system using Microsoft Kinect by introducing turns in rolling axis to the range of recognized movements, hence increasing the accuracy of motion recognition and offering the players a more natural way of interaction. Both bare hand and handheld object rolling rotation is introduced into motion controlled gameplay with developed algorithms. Two experimental games are developed each having two different controlling versions to evaluate the outcomes of the developed motion controlling methods and a user study is performed. Results of the user study are statistically analyzed and it is revealed that developed motion controlling methods provided the players a more natural and realistic experience, eliminating the additional gestures that were required for compensating lack of rolling rotation detection. Developed algorithms provided more accurate motion controlling which increased sense of being in control of the game, providing the player more enjoyment.

Outline of the thesis study is provided below:

- Chapter 2 includes background information and relevant work. First, motion tracking video game devices are presented. Then, previous works on gestural interaction are provided. Finally, motion controlled game examples from academia and industry are presented.
- Chapter 3 includes information on proposed approach. First, problem statement is stated that is followed by the detailed discussion of developed algorithms for both bare hand rolling axis of rotation detection and object rotation detection. Finally, limitations of the developed algorithms and implementation are presented.
- Chapter 4 includes information on evaluation of the developed algorithms. First, games that are designed for evaluation are explained with their important features. Then, information of the participants is presented that is followed by the experiment procedure. Finally, results of the user study and their discussion is presented.
- Chapter 5 includes conclusion of the study and implications of future work.

# CHAPTER 2

# BACKGROUND AND LITERATURE REVIEW

Although full body motion control is a recently becoming popular area of game technologies, motion controlling has a wide area of applications in computer science. Below, previous works that are related to motion controlling whose results will be used later in this thesis study to construct an improved motion controlling method are mentioned in three main groups: Motion Tracking Video Game Devices, Gestural Interaction and Motion Controlled Games.

## 2.1. Motion Tracking Video Game Devices

This subsection presents brief fundamental information on commercial motion controlling video game systems: Microsoft Kinect, Nintendo Wii and Sony PlayStation Move.

### 2.1.1. Microsoft® Kinect™

Kinect for Xbox 360 is a motion sensing accessory of Xbox 360 video game console that is released by Microsoft in 2010 [3]. The device can be seen below in Figure 2.1.

Figure 2.1: Microsoft Kinect for Xbox 360

Kinect creates an impressive synergy of motion sensing and gameplay with the unique way of interaction it offers to players: full body motion controlling without using any tangible controllers. Kinect altered the way that players interact with game consoles and offered the players a far more active way of interaction in comparison with pushing controller buttons. The company encapsulates the device emphasizing its unique way of interaction as follows: “Because with Kinect, there are no controllers. You are the controller.” [3]. The device is stated to provide the users a natural way of touch free interaction which the user already knows how to perform from their prior real-life interaction experiences [9].

Kinect is a powerful device running at interactive rates, yet affordable with a state of the art technology underneath. The device is able to detect spatial movements of the players and interpret them as game inputs. Kinect includes a red-green-blue (RGB) color camera, three dimensional (3D) depth sensors and four microphones that allow acoustic source localization inside [8]. The device also involves an internal motor that enables tilting in pitch axis of rotation. Both RGB and depth cameras are at the resolution of 640x480 pixels running at 30 frames per second (FPS).

Working mechanism of the device can be summarized as follows: The depth sensor is composed of an infrared (IR) projector that casts out IR rays and a complementary metal oxide semiconductor sensor that recognizes these rays and gives 3D depth data of each pixel [8]. IR rays that are cast out by Kinect to interpret depth information can be seen in the photo presented in Figure 2.2, which is taken with a digital camera at night vision setting. Depth data Kinect provides by interpreting cast out IR rays, which is visualized as an image stream can be seen in Figure 2.3. Left side of the figure shows an RGB image obtained with Kinect’s color camera and right side of the figure shows a depth image stream obtained with Kinect's depth camera and then converted into image stream.

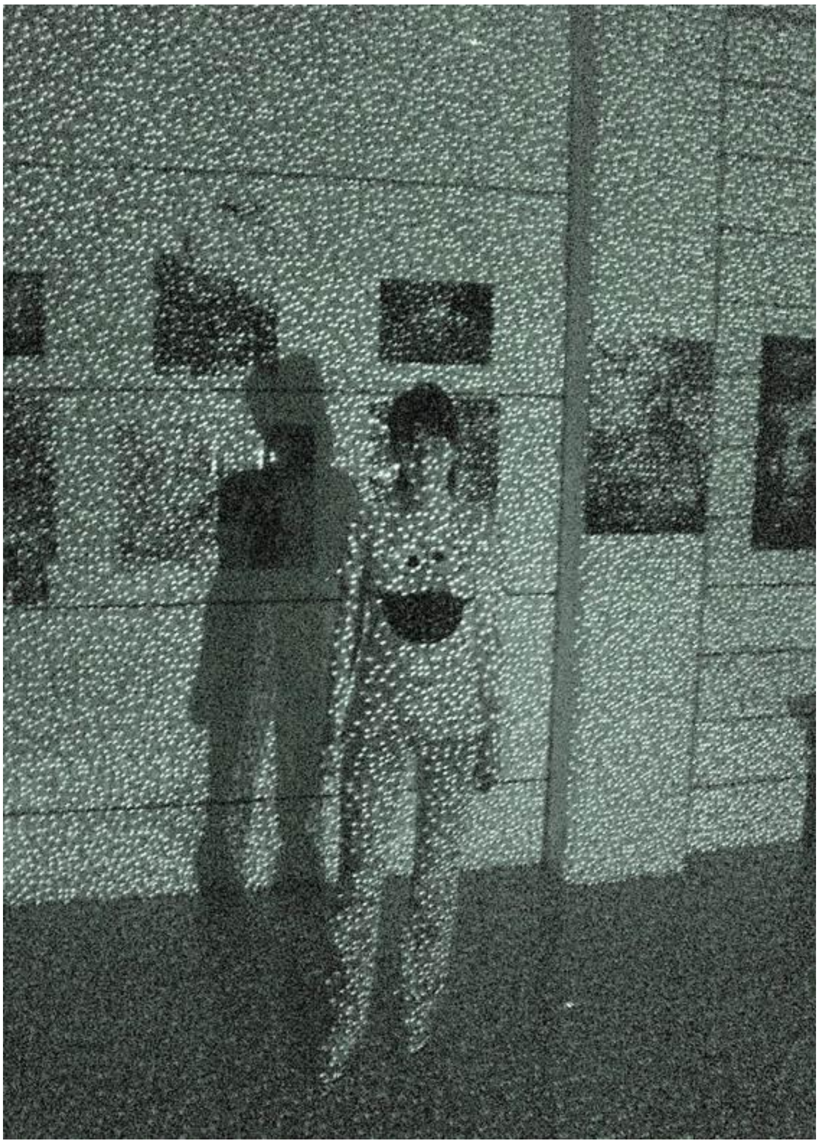


Figure 2.2: IR Rays Cast Out by Kinect to Detect Depth Information

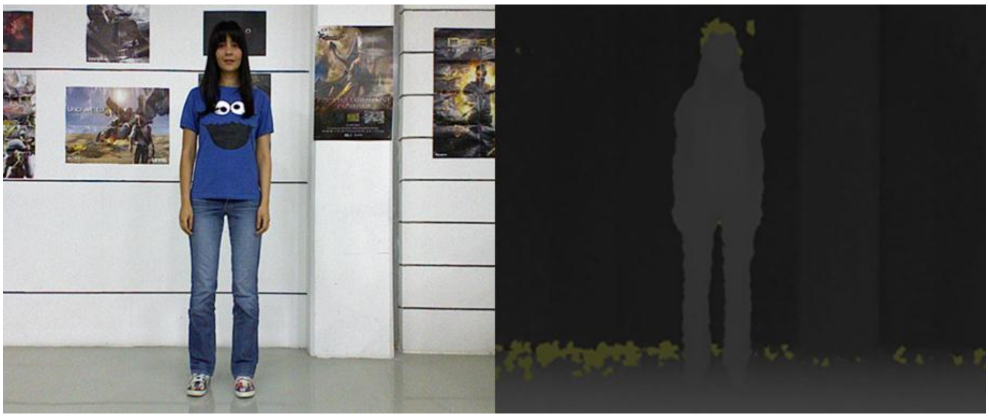


Figure 2.3: Depth Data Stream of Kinect Left: RGB Image Captured with Kinect's Color Camera Right: Depth Image Stream Captured with Kinect's Depth Camera

Understanding the depth information with IR rays makes the device independent of lighting conditions and able to operate in dark but on the other hand, the device is not capable of working in sunlight due to the IR confusion [8]. Kinect is also color and texture invariant which means that environmental color and texture does not affect the tracking capability of the device.

Kinect is able to track skeletons of up to two users standing in front by interpreting data acquired from the depth sensor, which enables the device to understand user's spatial movements and use them as controlling inputs. A tracked user skeleton is presented in Figure 2.4 which is captured using Kinect Explorer application of Microsoft Kinect SDK [10]. Left side of the figure shows an RGB image obtained with Kinect's color camera with the tracked skeleton drawn on it and right side of the figure shows a depth image stream on which tracked skeleton is also drawn.

Figure 2.4: User Skeleton Recognized by Kinect Left: RGB Image Showing Tracked Skeleton Right: Depth Image Showing Tracked Skeleton

Kinect interprets the user skeleton with 20 body joints which are illustrated below in Figure 2.5. In the figure, body joints recognized by Kinect are presented with their body labeling information compatible with Kinect's official SDK [10].

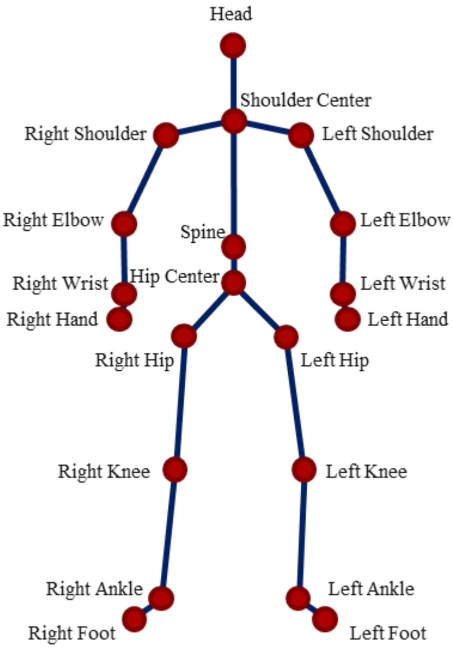


Figure 2.5: Body Joints Interpreted by Kinect in Skeleton Tracking

The device has a physical range of 57.5 degrees horizontal and 43.5 degrees vertical, having a tilt capability of ±27 degrees upwards and downwards [9]. Although Kinect has a sight of 0.8 to 4 meters, practically, the device works best between 1.2 to 3.5 meters since it detects and tracks user skeleton which is distinguishable by the device between these ranges optimally and giving unreliable skeleton tracking results outside of this range. Motion tracking range of Kinect is illustrated in Figure 2.6, presenting both physical limits under which the device is able to sense the environment theoretically and practical limits under which the skeleton tracking works reliably.

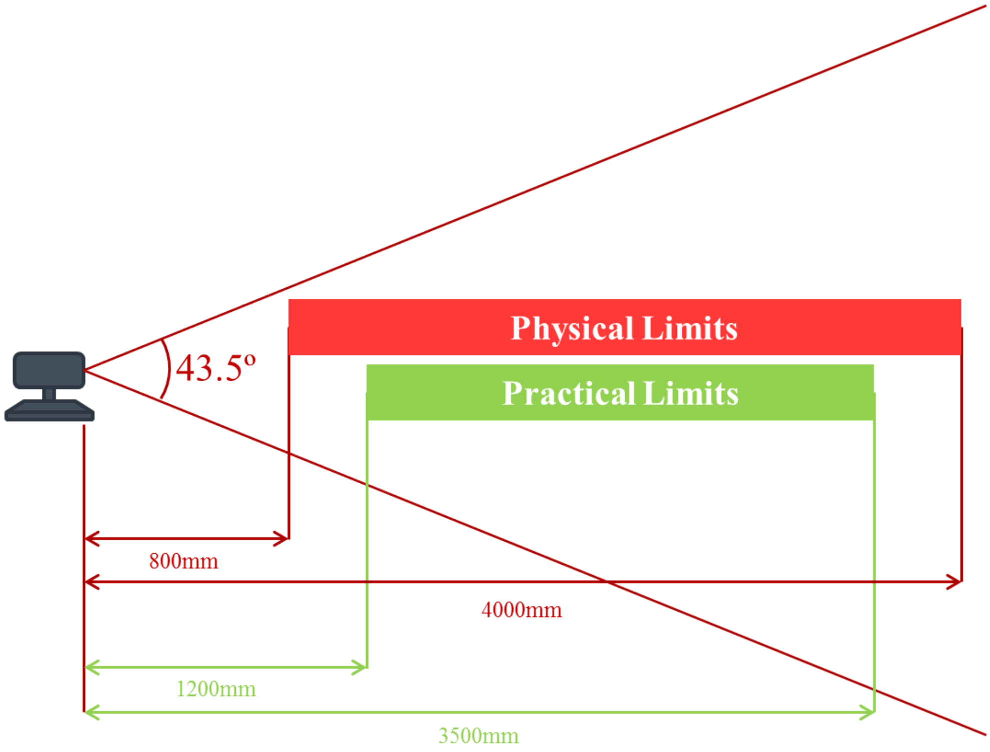


Figure 2.6: Physical and Practical Motion Tracking Limits of Kinect

To understand how Kinect interprets depth data and achieves markerless skeleton tracking, the work of Shotton et al. provides an explanatory basis for the underlying working principle of the device [11]. The authors proposed a novel method of predicting three dimensional body joint positions from a depth image without using any time related information meaning that body joint positions can be determined instantaneously without using any information from the previous image data. The method runs interactive at 200 frames per second on Xbox 360 game console, which is presented below in Figure 2.7. The authors developed a framework in which pixels are labeled as body parts with related probabilities and joint locations are determined by considering all pixels of the body parts allocated. For assigning the pixels to the body parts, pre-built decision trees are used.

Figure 2.7: Microsoft Xbox 360 Video Game Console

For a clear understanding of the device that is used for motion tracking in the scope of this thesis, working principle of Kinect can be summarized sequentially as follows [8]:

1. The user stands in front of Kinect.
2. Point cloud of the user is generated with the depth data.
3. The device makes a guess for the user skeleton.
4. User's body labels are guessed with different level of confidences.
5. Closest skeleton to the user's posture is assigned and drawn to the screen.

Microsoft enables the programmers to develop games and applications using full features of Kinect via the non-commercial software development kit (SDK) released under the name of "Kinect for Windows SDK" [10]. The SDK provides the user with raw data streams from depth sensor which gives depth information of the pixels, color camera sensor which gives color information of the pixels and microphone array which gives information of the voice coming from surrounding with its direction, and enables skeleton tracking that is achieved with the internal algorithm of the device. SDK also involves application programming interfaces (API) such as Natural User Interface API and Audio API. Natural User Interface API involves general functions such as activating and deactivating sensors, and controlling the tilting motor of Kinect. NUI API involves sub APIs such as NUI Image Camera API that provides the developer access to the cameras including depth and image information and NUI Skeleton API which provides the developer with skeleton tracking functions. Audio API provides the developer with access to microphone array information.

### 2.1.2. Nintendo® Wii™

Nintendo's motion sensing video game console Wii, which can be seen in Figure 2.8, offers the players a different way of motion controlling with a handheld input device [1].

Figure 2.8: Nintendo Wii Video Game Console

The video game system involves a sensor bar, a wireless controller called Wii-mote that communicates with the sensor bar via Bluetooth, and an accompanying controller called Nunchuck that is plugged into Wii-mote via a short cord. Motion controller Wii-mote is presented in Figure 2.9. There exists a simple infrared camera in front of the Wii-mote that recognizes the light emitting from the sensor bar, which is usually located on top of the television, to measure the distance between sensor bar and itself.

Figure 2.9: Wii-mote Motion Controller of Nintendo Wii

Sensor bar of Nintendo Wii can be seen in Figure 2.10 which presents two shots of the bar, above involving normal appearance of sensor bar to human eye which is taken at regular settings of a digital camera and below involving appearance of the light cast out by light emitting diodes (LED) at both sides of the bar which is taken at night vision setting of a digital camera.

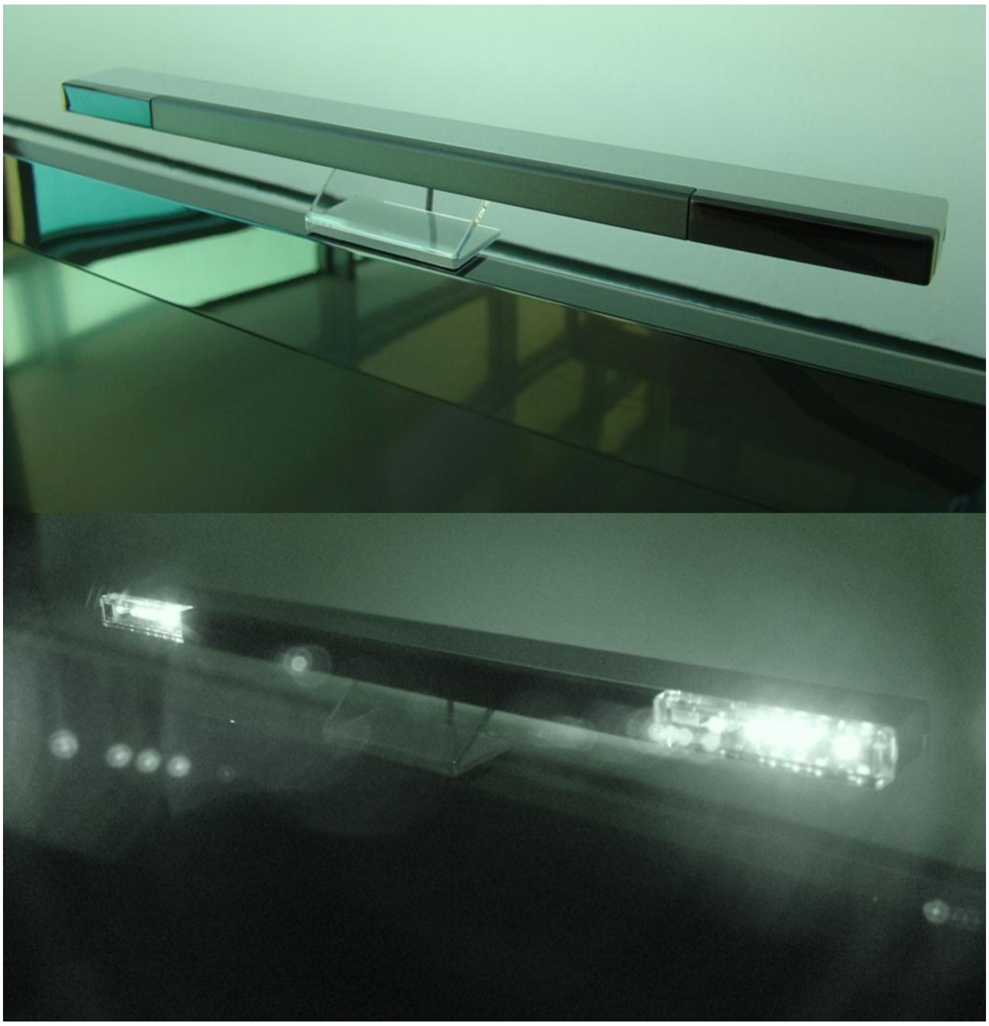


Figure 2.10: Sensor Bar of Nintendo Wii on Top of a Television Above: Sensor Bar Shot at Normal Settings Below: Sensor Bar Shot at Night Vision

Button pressing like input transmission from the Wii-mote to the console is achieved via Bluetooth [12, 13]. Wii-mote also includes accelerometers that sense linear acceleration in three axes of x, y, z and this enables the system to track the remote's motion accurately. Wii-mote also hosts a speaker for audial output, a vibrator for tactile output and four light emitting diodes for visual output. The accompanying controller Nunchuck that is used for motion based interaction is presented in Figure 2.11. On the Nunchuck, an analog stick is located at the front and the controller also involves an accelerometer that sense acceleration in three axes as in Wii-mote [14].

Figure 2.11: Nunchuck Motion Controller of Nintendo Wii

An expansion to the Wii-mote called Wii Motion Plus is released in 2009. Wii Motion Plus can be attached to the bottom of a standard Wii-mote and includes a gyroscope that provides rotational motion data in three axes, providing a more accurate and richer motion controlled gameplay experience to players [8].

There are some non-official SDKs that provide access to limited data of Wii-mote after connecting the controller to a computer via Bluetooth enabling the developers to use the controller's motion tracking features to develop custom games and applications [14].

### 2.1.3. Sony® PlayStation Move™

PlayStation Move is the motion tracking system that is released by Sony Computer Entertainment in 2010 for PlayStation 3 video game console [2]. In the system, a digital camera which is called Play Station eye is used for color and motion tracking [8]. The device also involves an internal microphone array that enables voice tracking. Motion input is provided via a wireless handheld motion controller that is presented in Figure 2.12 at the left side, which involves an orb at the front that contains RGB LEDs inside and is illuminated in various colors by the system to recognize the 3D position with color tracking. Color of the orb is specifically selected by the system to be distinguishable from the environment. Internal light tracking procedure makes the system invariant to other environmental lighting conditions. Up to four controllers that are illuminated in different colors can be recognized by the system simultaneously. The motion controller also involves a gyroscope that provides 3D angular data and an accelerometer that provides 3D acceleration data. The system also involves an accompanying controller called navigation controller that can be seen in Figure 2.12 at the right side, which is used synchronously with the motion controller in the other hand. The navigation controller hosts an analog stick and some buttons on it.

Figure 2.12: PlayStation Move Motion Controller and Navigation Controller

Sony enables the programmers to develop motion controlled games and applications using the features of PlayStation Move system via the software development kit released under the name “Move.Me” [8]. SDK provides the user three dimensional orientation, velocity and acceleration data of the handheld motion controller.

## 2.2. Gestural Interaction

In this subsection, previous works that are related to gestural interaction are presented from different point of views. First, a definition of gesture is provided to be used consistently throughout the thesis, followed by a brief history of gestural interaction for video games. Subsequently, gestural interaction is reviewed from the points of expressiveness, naturalness, learning requirements and motion projection.

### 2.2.1. Definition of Gesture

There are many definitions of what a gesture is in the literature. Mitra and Acharya define gesture to be expressive and meaningful movements of human body that are performed for either transmitting information of having a meaning or environmental interaction [15]. Grandhi et al. refer gesture in their work as “movements that communicate information from shared common ground” [16].

A gesture, in the scope of this thesis means any physical action performed by the player that the motion tracking device can interpret as an input such as waving, pointing, twisting, and snapping.

### 2.2.2. A Brief History of Gestural Video Game Interaction

In his book “Designing Gestural Interfaces”, Saffer shares valuable information and insight about gestures [17]. It is known that the earliest gestural input system for video games, which is called “Essential Reality P5 Glove”, is published by Lionhead Studios, an English game development studio founded in 1997, in the form of a specially designed wired glove that transmits hand gestures of the player as inputs for the simulation genre video game “Black & White”. As a sequent progress, Konami, a Japanese game development studio, published a motion capture boxing game for the video arcades in 2001, which is played by wearing physical boxing gloves that transmit hand motions of the players via infrared.

The progression of gestural interaction for video games continued till recent years that witnessed an evolution of gesture controlled interaction for gaming with the release of motion tracking game consoles Wii, PlayStation Move and Kinect.

Gestural interaction endeavors are not limited to commercial video game systems, there are various computer science related academic works in the search of advanced gestural interaction systems such as the study of Wang and Popović, who proposed a real time 3D hand tracking framework that is achieved with the help of a camera by using a specially designed multi colored glove of having unique color pattern that is worn on the hand and enables tracking the motions of the hand in three degrees of freedom via color tracking [18]. In the work, a database of 18,000 different hand poses that cover commonly used hand gestures is constructed and the hand pose is determined by finding the nearest neighbor from the poses in the database. The proposed system enables successful hand tracking and pose recognition but with the limitation of the special cloth equipment that needs to be worn on hand for the tracking to be performed.

### 2.2.3. Expressiveness of Gestures

Wachs et al. state that gestures serve as a source of expressive guide for the players in video games [19]. Some gestures are self-expressive such as a hand pointing forward resembling movement in that direction, and contrarily an open stationary palm expressing stillness. So, if designed with this consideration, gestures may give the player ease of correlating movements to their aimed actions with their explanatory nature.

Kinect's official interface guideline provides valuable information on gesture design principles [9]. The authors state that meaningful gestures should be assigned for actions, which the users can easily interpret such as pointing left to go in the left direction. Abstract body movements, having no logical relationship with the action such as moving the hand up to go in the left direction, that are needed to be learned by the users are advised to be avoided, since they will be difficult to remember for the users during gameplay, requiring a lot of mental effort.

### 2.2.4. Naturalness of Gestures

In his book, Saffer states that gestural interaction is a natural form of communication as follows: "Interactive gestures allow users to interact naturally with digital objects in a physical way, like we do with physical objects." [17]. He also states that best interaction designs are those that the users naturally know how to use. "The best, most natural designs, then, are those that match the behavior of the system to the gesture humans might already do to enable that behavior." The author emphasizes that gestural design should be seamless, mimicking natural behaviors of human being that the users already know how to perform intrinsically, without spending too much effort to learn them, reminding that interactive gestural systems are called to be natural user interfaces due to their being a more natural method of interaction which is advised to be kept in mind during gestural design.

Roccetti et al. state that recent trends show a correlation between a game's degree of realism in interaction and its success in the market [20]. The authors assert that players do not get entertained with using classical controllers such as keyboards and mice any longer, but seek for performing with realistic natural interactions which involve real movement.

Authors of Kinect's official interface guideline state that in order to provide a good gaming experience for the players without frustrating them, gestures should be designed to be as intuitive and natural as possible, having easy mental mappings, making users comfortable while performing them [9]. The authors state that the games that are built for Kinect will be most enjoyable if the interaction is as natural as possible, making the user feel like they already know how to perform movements. In the work, innate gestures are described to be "the ones that the user intuitively knows or that make sense based on the users' understanding of the world" and gestures are advised to be designed as innate as possible to provide the users a more natural way of interaction.

### 2.2.5. Learned Gestures

Grandhi et al. state that most of the gestures that are used in applications are chosen due to their being easy to implement but this incurs the users burden of memorizing unnatural and non-intuitive actions that they may have difficulty with relating them to the actions in real world [16]. Authors proposed a more natural approach of gestural design for applications. In the work, a user study is performed by showing the participants two sequential pictures, showing a before and after situation of commonly used easy computer related tasks such as cutting and opening, that are represented on the pictures as everyday non computer related tasks they resemble visually. Results of the user study revealed that using gestures to interact with a system gets easier when they are designed to be in a familiar form for the users as in the case of being inherited from commonly used daily natural tasks.

Wachs et al. state that there exists strong evidence showing that future human computer interfaces will allow more natural communication via gestures, in a form that is not much different than human to human natural communication [19]. Users who prefer using classical input devices such as mouse are expressed to be disinclined to adapt to this new technology of touchless interaction. A possible solution to encourage them for this new type of interaction is proposed to be the ease of adaptation that may be provided with requiring shorter periods of learning with natural gestures employed. The authors emphasize that gestures are one of the most fundamental and expressive communication means of human being. In the work, it is also criticized that many gestural interface systems lack the naturalness of human gestural communication that can be observed beginning from the early childhood. It is also stated that unnatural gestures that are needed to be memorized incur mental load to the users, hence gestures should better be designed as being natural and intuitive as possible, resembling everyday sign language to prevent users from trying to memorize and remember complicated action sets. Gesture controlled games generally require a training tutorial for the players to learn all the gesture configurations required to play the game and this may be a discouraging factor for the player as compared to a natural, already known gesture set.

In his work, Lee states that the term natural user interface resembles an intuitive interaction system that is not visible to the user while the act of interaction, like the way we communicate with people in everyday life [21]. In the work, communicative gestures are stated to be classified into three as mimetic gestures that imitate the shape or behavior of an object, deictic gestures that express information and arbitrary gestures that are stated to be learned motions which belong to specific communication mediums. The author states that arbitrary gestures are widely chosen to be used in human computer interaction interfaces due to their being easily distinguishable, but these unnatural gestures require the user to learn them via training and it would be better to use mimetic and deictic gestures due to the user's acquaintance to these natural means of communication without the need of learning new, unfamiliar sets of movements.

In the Kinect's official interface guideline, it is stated as a rule of thumb that number of gestures should be kept as small as possible while developing motion controlled games so that users will not have difficulty in learning and remembering them during gameplay [9]. It is also stated that keeping the gestures simple, easy to learn and easy to master is important for making the users comfortable and confident, encouraging them to play the developed game.

The authors point out that the more the gestures are easy to learn and recall, the less the mental work is imposed to the users which may engage them more with the game. It is also indicated that innate gestures are better be chosen to provide the user less learning requirement, rather than unnatural ones requiring mental effort to be learned and remembered.

### 2.2.6. Motion Projection

Simon states that the better the correspondence of the player's motions are into the game, the more enjoyable the gameplay becomes for them [22]. With a system understanding even advanced movements of the player, in other words sensing the gestural excess and incorporating it into the game, giving visual real time representation of their movements and showing the effects of their performance on gameplay provides a more immersive gameplay. The author indicates that it is a more immersive and attractive way of gameplay in which the player actually performs realistic actions and observes that his/her own actions affect the outcomes of the game rather than the player's pretending to do some actions that have no effect on the outcomes of the game.

In their work, Lok et al. perform an experiment in which users are instructed to make a pinch gesture to pick up a virtual element in the application but given visual feedback with the virtual avatar making a grasping move [23]. It is stated by the authors that 25% of the participants were distracted by this exaggerated action projection and inclined to make a grasping gesture resulting in the experimenter's reminding of the correct gesture of pinching required to be done to grasp the virtual blocks. The authors state that kinematic fidelity of a virtual avatar, which enables the users see their actions projected into virtual world analogously, provides the user more sense of presence than visual fidelity of a virtual avatar, which provides the users see a virtual avatar that resembles them physically in the virtual world. The participants of the experiment are reported by the authors to state that they felt more sense of presence as the avatar moved corresponding to their moves, rather than seeing an avatar which visually looks like them. It can be inferred from the study that accurate motion projection may help to increase sense of presence of the players, resulting in improved engagement and immersion.

Frasca states that players enjoy games more as they are allowed to affect the system outcomes with their performances, which can be observed in real world from players' making exaggerated moves even if they do not affect the system outcome such as a person's rolling a dice after making weird swinging moves of the hand, believing in that his/her performance would affect the result of a totally probabilistic event [24]. The author points out that players get more engaged as they discover that their movements affect the system outcomes, feeling more in control.

As all of these previous works emphasize, natural gestures are the key for a successful interaction system. Although full body motion tracking provided by Kinect is an incomparably more natural way of interaction than using gamepads by pressing buttons, the limitation in the recognition of rolling axis degrades the naturalness of the interaction, limiting the degrees of freedom supported by the range of  recognized movements. As Buxton states, “The richness of interaction is highly related to the richness/numbers of degrees of freedom (DOF), and in particular, continuous degrees of freedom, supported by the technology.” [25].

## 2.3. Motion Controlled Games

Motion controlled video games constitute an attractive area for both academic researchers and commercial game development companies. There are many examples some of which that are relevant to the scope of this thesis are mentioned below in the following subsections under two categories: academic game studies and commercial games.

### 2.3.1. Academic Game Studies

There are several experimental games that are developed by academic community for research purposes, which employ motion sensing technology to achieve better gameplay by incorporating players into games more actively with body movements and gestures.

Tang et al. developed a dancing game in which a virtual partner accompanies the player by responding his/her moves synchronously with the usage of an optical motion capture system and employment of a progressive block matching algorithm [26]. The work brings novelty to the classic motion controlled dance games, by enabling the player a virtual partner that dances in response to his/her movements rapidly interpreting the motion capture data and using pre-recorded dance move templates, providing a more interactive way of gameplay.

Rocetti et al. developed an instructive motion controlled video game of preparing tortellino, an Italian stuffed pasta, in which the player repeats after the movements of a virtual chef [27]. In the work, movements of the players are recorded via a camera that is placed on top, and visualized to the players in digital animated representation forms of their actions, by first delivering the data to a software to be assessed in terms of correctness of the movements, then giving feedback to the user. Gestures that the player performs are recognized according to the beginning and ending positions of the player's hands and the trajectory they follow. Although the gestures that are recognized by the developed system are limited with the scope of the movements that are needed to prepare tortellino in real-life, the work constitutes an effective implementation of interactive gestural systems in a serious game which teaches the players preparation of a popular culinary recipe.

Another example can be given as the work of Hamalainen et al. that introduces a gestural controlled martial arts game in which the player fights with virtual opponents using realistic moves such as kicks and punches for the sake of training and self-progression [28]. Motion tracking is achieved with the usage of a web camera followed by image processing with OpenCV library. The work constitutes a good example of the usage of motion controlled games for skill development and training, where accuracy of the motion tracking plays an important role.

These academic endeavors provide examples for the effective employment of motion tracking systems to control interactive games.

### 2.3.2. Commercial Games

In addition to academic experimental games, there are also various commercial Kinect games that use full body motions of the player as controlling inputs, which may have been benefited the richness of gestures provided by the addition of rolling axis turn, considering their themes and gameplay styles.

An example is the action adventure video game "Harry Potter and the Deathly Hallows™ - Part 1", in which the protagonist uses his wand in a virtual magical world [29]. The imaginary wand that is held at hand is directed to in game objects to be used and this type of interaction may hypothetically benefit from rolling axis motion of the hands which would provide more literal and richer motion control, eliminating unnatural gestures to be learnt that are assigned in replacement of the rolling movement.

“Kinect Star Wars” is another example which may hypothetically benefit from tracked rolling motion of the hands [30]. In one part of the game, the player controls an airplane with two hands that are directed to Kinect. With the recognition of rolling movement, the airplane might have been controlled with one hand alternatively, if had been found to be fitting into the design of the game by the developers.

Sword battling in “Deca Sports Freedom” Kinect game may also hypothetically benefit from the recognition of the roll axis turn of the hands that are directed to the sensor [31]. The sword might have been swung to the sides according to the rolling movement of the hand, providing a more realistic and immersive sword battle with a more natural interaction.

Another example that may be enriched in scope of the motion controlled interaction can be given as table tennis in “Kinect Sports” video game [32]. Table tennis is a sport that utilizes wrist turn mostly. In the game, players are required to swing their arms to perform hits with the imaginary paddle. To add spin to the ball, which is accomplished by also turning the paddle in the roll axis of hand in real-life, required gestures are moving the hand upwards and downwards, since the system does not recognize rolling movements. Table tennis is another example that may benefit from rolling motion recognition due to its nature of being played with turned wrists.

A final example can be given as the game "Dance Central 3", which may hypothetically benefit from recognized rolling turns of body limbs while the players mimic real life dance figures, providing a more natural and accurate interaction style [33].

On the contrary, a few commercial game examples that include rolling axis of rotation in their scope of motion interaction with the help of specially equipped motion controllers can be given as "Smooth Moves" [34] and "Kororinpa" [35]. "Smooth Moves" is a Nintendo Wii game that is mainly focused on the advanced motion sensing capabilities that are added to the gameplay by Wii Motion Plus, especially rolling action, which is achieved via utilizing the gyroscope inside the controller. "Kororinpa" is a puzzle genre Nintendo Wii game that is controlled by tilting the Wii-mote with Motion Plus in its rolling and pitching axes.

Each type of interaction comes with its advantages and disadvantages, and do not fit into all gameplay designs. Although it may not be suitable for all motion controlled game designs to use the rolling movement as a controlling input, since it is a natural human movement to perform many games may benefit from the addition of the movement in scope of the motion recognition with numerous possible new gestural and gameplay designs.

# CHAPTER 3

# PROPOSED APPROACH

This chapter presents the proposed approach that is implemented and evaluated in the study to achieve an improved gameplay by introducing rolling motion of rotation into motion controlling. First, problem statement is made which is followed by the algorithm explanations. Then, limitations of the developed algorithms are described and in the final subsection, implementations of the algorithms are provided with supplementary code excerpts. In the following chapter, evaluation of these developed algorithms will be presented.

## 3.1. Problem Statement

As discussed in the previous sections, although Kinect provides a natural way of motion controlled interaction, rolling axis of motion is not recognized by current motion tracking scope of the tool, as no rotational information is supported by the skeletal tracking provided, creating a limitation in the variety of possible gestures that may be used for motion controlling.

Rolling axis of motion is a natural degree of freedom (DOF) that people use in their everyday movements. As the previous works in gestural interaction indicate, the interaction should be as natural as possible to engage the player into the game. Inclusion of the rolling action into gameplay may provide a more natural way of interaction, eliminating new unnatural gestures that are assigned in place of the rolling movement that could give rise to learning difficulties for the player.

This thesis aims at removing some of the motion tracking limitations using Kinect, in the search of an improved gameplay. In the Kinect's official interface guideline, skeleton data is stated to be very unreliable as hands and arms are kept in front of the user body while performing gestures, which imposes a limitation on the gestural design and variety [9]. In the document, it is also stated that the key gesture of Kinect, which is waving, has to be performed by the player with rotating the elbow for the system to recognize the motion, waves that are performed by turning only the wrist are not recognized by the system, due to the same rotational limitation. This study aims at providing a more natural way of motion controlled gameplay using Kinect, removing rolling axis limitation.

Problem definition can then be stated as follows: Increasing degree of freedom for an improved more natural motion controlled gameplay using Kinect, which recognizes rolling axis hand rotations of the player.

## 3.2. Algorithms

In this thesis, two rolling axis of motion recognition algorithms are developed and implemented for both bare hand rotation recognition and handheld object rotation recognition. In the following subsections, these algorithms are presented and discussed in detail.

### 3.2.1. Bare Hand Rolling Axis of Rotation Detection Algorithm

As discussed in the previous sections, rotations of body limbs are not recognized by the current skeleton tracking pipeline of Kinect, indicating that rotation of hands performed from wrists cannot be tracked. To give a better understanding, aforementioned rotation is illustrated in Figure 3.1 using a representation of the standard skeleton tracked by Kinect. Hand's rolling axis of rotation can be described as the rotation in the direction of rounded arrows that are shown in the figure. Global axis configuration is also presented at the bottom of the figure. In the coordinate system used, global x axis is pointing towards right of the player, global y axis is pointing towards above the player and global z axis is pointing towards front of the player. The algorithm that is developed is able to recognize rolling rotation of a hand that is directed through the global z axis. Rolling axis corresponds to global +z and -z axes in this configuration.

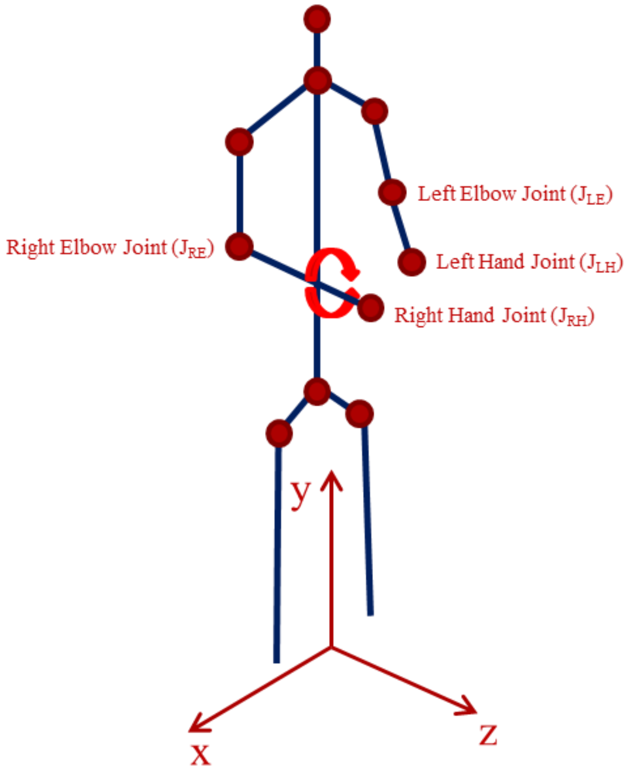


Figure 3.1: Rolling Axis of Rotation Illustration of Right Hand Joint

To provide a clear understanding of the rolling movement that is incorporated into motion tracking with the proposed algorithm, three consequent image sets each consisting of an RGB image above and a depth image below that are captured using Kinect Explorer application of Microsoft Kinect SDK [10] are presented in Figure 3.2. In the figure, left image set shows the player's extended hand rotated to her left side in global z axis, middle image set shows the player keeping her extended hand horizontal and right image set shows the player's extended hand rotated to her right side in global z axis.

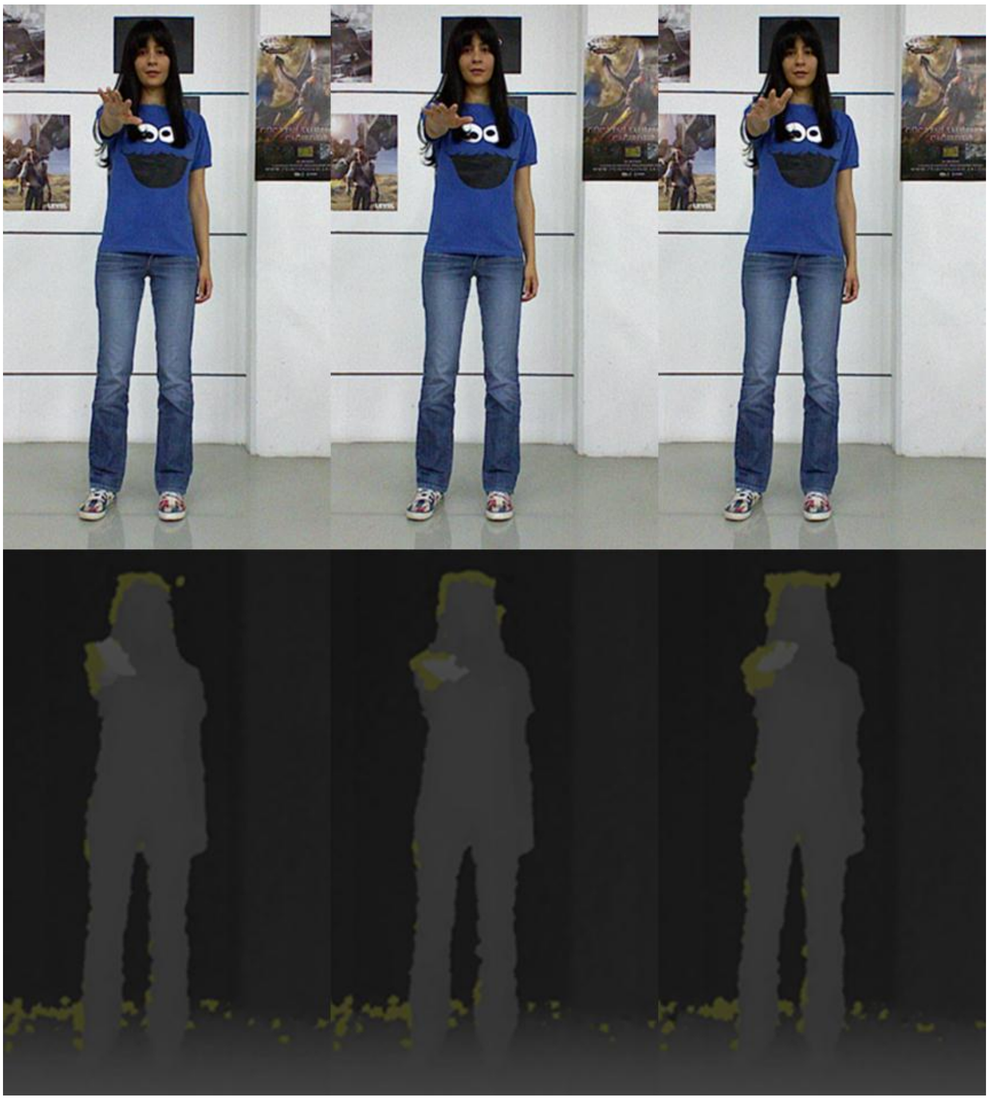


Figure 3.2: Image Sets Showing the Player's Hand Rotated in Rolling Axis Left: The Player Rolls Her Hand to Her Left Middle: The Player Keeps Her Hand Horizontal Right: The Player Rolls Her Hand to Her Right

The proposed algorithm to incorporate rolling movement of the extended hand in global z axis is discussed below in eight steps. The algorithm is presented here for the rolling movement recognition of right hand, to apply the algorithm for the left hand; all data that are related to right hand should be changed with the data of left hand. To recognize motions of both hands simultaneously, the algorithm should be iterated through for each hand separately.

**Step 1:** Position of the right hand joint in 3D world coordinates ($WJ_{RH}$) is gathered from the skeletal data stream that is provided by Microsoft Kinect for Windows SDK [10].

**Step 2:** $WJ_{RH}$ is converted to two dimensional (2D) depth coordinates $DJ_{RH}$ from 3D world coordinates for being able to work with the 2D depth data stream gathered from Kinect depth sensor thereafter.

**Step 3:** Depth data of the right hand joint ($d_{RH}$) is gathered from the depth data stream of Microsoft Kinect for Windows SDK [10] by finding out the 2D depth value which corresponds to $DJ_{RH}$.

**Step 4:** A linear search is performed starting from $DJ_{RH}$ in the search direction $ds_n$ where n is the angle from horizontal, ranging from 0 degrees to 180 degrees, with a step size of 1 degree. Iterative search directions are visualized in Figure 3.3. In the figure, a hand that is directed to the sensor is illustrated and search directions from 0 to 180 degrees are presented with two sided dashed orange lines.

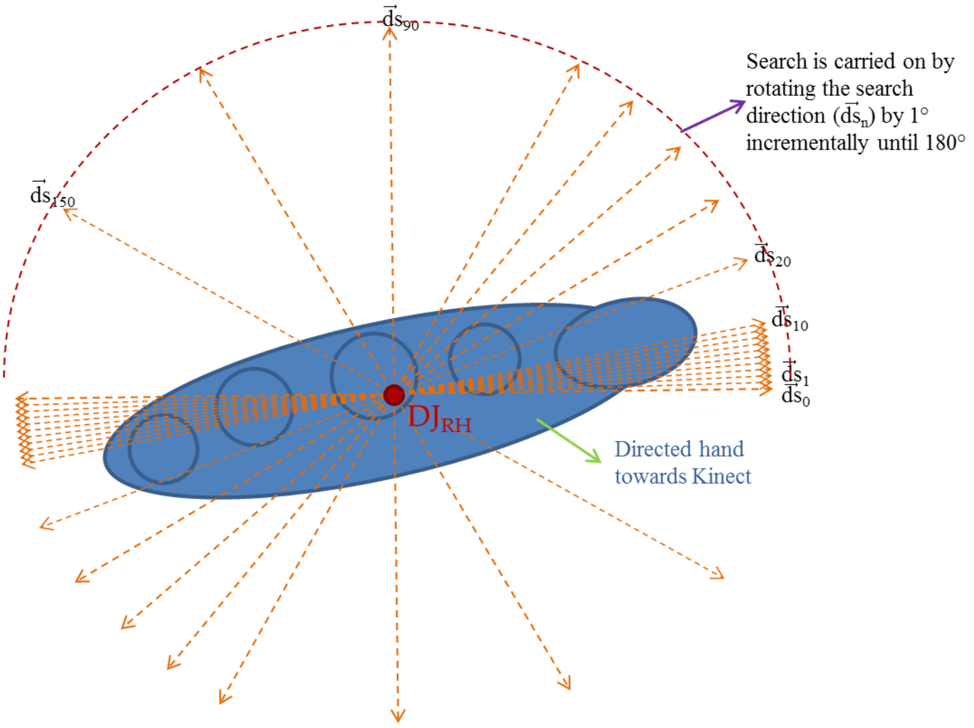


Figure 3.3: Linear Search Performed Incrementally Rotating the Search Direction by 1 Degree until 180 Degrees

**Step 5:** For each $ds_n$, edge points ($H_{n,1}$, $H_{n,2}$) are detected by comparing depth of each iterated pixel $d_{x,y}$ with the previous one to detect a considerable decrease, meaning that the hand has ended and edge point of the considered side is reached. The iteration continues as long as $d_{RH}-\varepsilon < d_{x,y} < d_{RH}+\varepsilon$, where $\varepsilon$ is a coefficient which is determined to be 150 for the scope of this thesis assuming that the maximum hand finger length of human being is approximately 150 millimeters and can be tweaked accordingly for the other usages of this algorithm than hand rotation detection.

The search is performed two ways, starting from the $DJ_{RH}$ and progressing in the positive and negative $ds_n$ sequentially. After one of the edge points is detected in either positive or negative $ds_n$, the search is applied in the opposite direction beginning from the same starting point of $DJ_{RH}$ to find the complementary edge point.

**Step 6:** Distance between $H_{n,1}$ and $H_{n,2}$ ($d_{Hn,1-Hn,2}$) is calculated and the angle n is stored in a variable which is updated at each iteration, always replaced with the angle of new maximum distance compared to the previous iteration, to keep the angle of the direction giving the maximum distance between two edge points.

**Step 7:** After the search is performed for all $ds_n$, last stored value of n gives the rotation of the extended hand in rolling axis.

**Step 8:** This algorithm is performed at 30 FPS, synchronous with Kinect's frame rate and the rolling axis of rotation of the hand is incorporated into the motion controlled gameplay using the rotation information gathered in Step 7.

To provide a clear understanding, algorithm is illustrated in Figure 3.4, presenting a visual summary of the steps that are discussed above. In the figure, edge points that are found in search iterations are presented with yellow circles, distances between two edge points are presented with purple lines and the maximum distance between two edge points is presented with a thick red line.

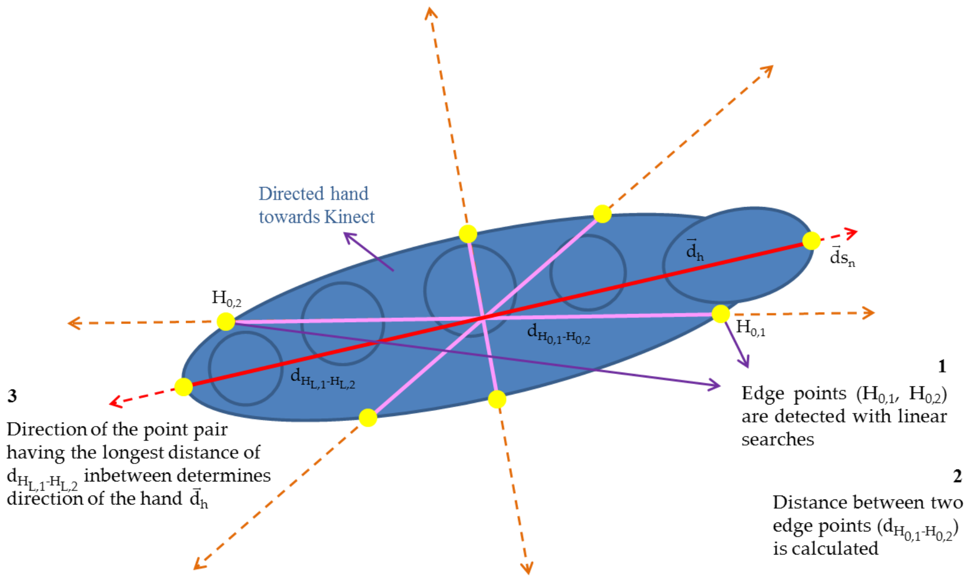


Figure 3.4: Visual Summary of the Hand's Rolling Axis of Rotation Recognition Algorithm

A visual display of the algorithm is presented in Figure 3.5, with the player standing in front of Kinect and extending his right hand forward in global z axis. On the left side of the figure, the player is visualized with the RGB image involving the standard skeleton that is tracked by Kinect and on the right side, depth image of the player is presented, involving rotational direction of his right hand that is visualized with a red line, in addition to the tracked skeleton present.

Figure 3.5: Detection of the Hand Rotation from Depth Data Left: RGB Image of the Player Directing His Right Hand towards Kinect Right: Depth Image of the Player with Right Hand Rotational Direction Detected

Although visualization is done using an open hand, the algorithm is able to detect fists either since it measures the maximum distance of the line that passes through the origin of the hand meaning that as long as the hand will not be in a perfect round shape and have a ratio of width to height other than 1, the algorithm will be able to detect hand rotation, no matter in which form it is.

### 3.2.2. Object Rolling Axis of Rotation Detection Algorithm

[36] presents a study on introducing handheld tangible objects into motion controlled gameplay using Kinect. Rolling axis of rotation recognition of the

handheld tangible object may provide more degree of freedom to the user, yielding to more accurate motion projection. Considering these improvements, an algorithm that recognizes rotation of the handheld objects in rolling axis is presented in this subsection. To give a better understanding of the handheld object's rolling rotation, an illustration is presented in Figure 3.6. In the illustration, the player carries an object in the right hand. The object's rolling axis of rotation is the rotation in the direction of rounded red arrows that are shown in the figure. Local axis configuration of the handheld object is also presented in the figure, represented with blue arrows located on the object. In the coordinate system used, local x axis is pointing towards tip of the object, local y axis is pointing towards palm of the right hand and local z axis is pointing towards bottom part of the fingers of right hand in grasp form. The developed algorithm is able to recognize rolling rotation of an object in its local +x and -x axes.

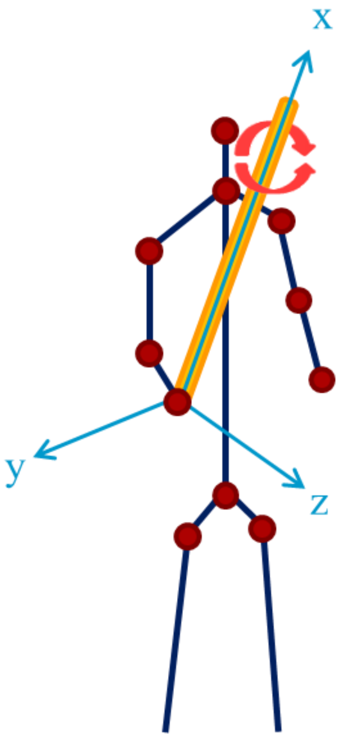


Figure 3.6: Rolling Axis of Rotation Illustration of a Handheld Object

The proposed algorithm to incorporate rolling movement of a handheld object is discussed below in nine steps. The algorithm is presented here for the rolling movement recognition of the object that is held at the right hand, to apply the algorithm for the left hand; all data that are related to right hand should be changed with the data of left hand. To recognize motions of the handheld objects at both hands simultaneously, the algorithm should be iterated through for each hand separately.

**Step1:** Position information of the tip point of the grabbed object ($T_o$) in world coordinates is found by using the proposed approach in [36]. Position of the right hand joint in world coordinates ($WJ_{RH}$) is gathered from the skeletal data stream of Microsoft Kinect for Windows SDK [10].

**Step 2:** $T_o$ and $WJ_{RH}$ is converted from 3D world coordinates to 2D depth coordinates ($DT_o$ and $DJ_{RH}$ respectively) for being able to work with the 2D depth data provided by Microsoft Kinect for Windows SDK [10] from now on.

**Step 3:** Midpoint of the object ($M_o$) and direction of the object ($\vec{d_o}$) are found using $DT_o$ and $DJ_{RH}$ with Equation 3.1 and Equation 3.2 that are presented below.

$$M_o = ( DT_o + DJ_{RH} ) / 2 \qquad \text{(Equation 3.1)}$$

where

$M_o$ = Midpoint of the object in depth coordinates

$DT_o$ = Position of the grabbed object's tip point in depth coordinates

$DJ_{RH}$ = Position of the right hand joint in depth coordinates

Slope of the Object = $( DT_o.y - DJ_{RH}.y ) / ( DT_o.x - DJ_{RH}.x )$ (Equation 3.2)

where

$DT_o.y$ = y component of the position of the grabbed object's tip point in depth coordinates

$DJ_{RH}.y$ = y component of the position of the right hand joint in depth coordinates

$DT_o.x$ = x component of the position of the grabbed object's tip point in depth coordinates

$DJ_{RH}.x$ = x component of the position of the right hand joint in depth coordinates

Since $\vec{d_o}$ is perpendicular to the slope of the object, it can be found by using Equation 3.3 that is presented below.

$\vec{d_o} = - ( DT_o.x - DJ_{RH}.x ) / ( DT_o.y - DJ_{RH}.y )$ (Equation 3.3)

where

$\vec{d_o}$ = Direction of the hand held object

$DT_o.x$ = x component of the position of the grabbed object's tip point in depth coordinates

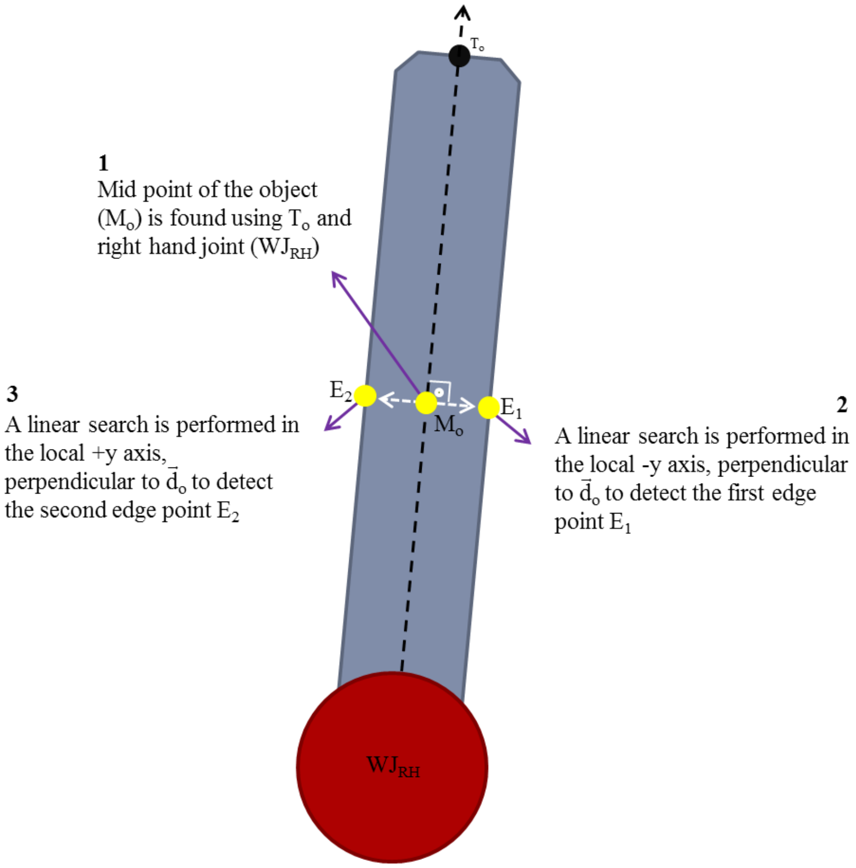
$DJ_{RH}.x$ = x component of the position of the right hand joint in depth coordinates

$DT_o.y$ = y component of the position of the grabbed object's tip point in depth coordinates

$DJ_{RH}.y$ = y component of the position of the right hand joint in depth coordinates

**Step 4:** A linear search is performed in local -y axis, perpendicular to $\overrightarrow{d_o}$ to detect the first edge point ($E_1$) and then another linear search is performed in the local +y axis, perpendicular to $\overrightarrow{d_o}$ to detect the second edge point ($E_2$).

To provide a clear understanding, Step 3 and Step 4 are illustrated in Figure 3.7 in which midpoint of the handheld object and two edge points are found respectively. In the figure, a handheld sword is visualized on which two edge points and the midpoint are illustrated with yellow circles. Tip point of the object that was found in [36] is also illustrated in the upper part of the figure with a black circle located at the tip of the object. Handheld object direction is represented with a dashed black line in the figure.



1
Mid point of the object ($M_o$) is found using $T_o$ and right hand joint ($WJ_{RH}$)

2
A linear search is performed in the local -y axis, perpendicular to $\vec{d}_o$ to detect the first edge point $E_1$

3
A linear search is performed in the local +y axis, perpendicular to $\vec{d}_o$ to detect the second edge point $E_2$

Figure 3.7: Illustration of the Algorithm Step 3 and Step 4 in which Midpoint of the Object and Two Edge Points are Found Respectively

**Step 5:** $E_1$ and $E_2$ are converted into 3D world coordinates ($W_{E1}$ and $W_{E2}$ respectively) to be able to find the 3D vector passing through these points.

**Step 6:** A 3D vector ($W_{E1\text{-}E2}$) passing through $W_{E1}$ and $W_{E2}$, having a direction from $W_{E1}$ to $W_{E2}$ is calculated using Equation 3.4 that is presented below.

$$W_{E1\text{-}E2} = W_{E2} - W_{E1} \qquad \text{(Equation 3.4)}$$

where

$W_{E1-E2}$ = 3D vector from $W_{E1}$ to $W_{E2}$

$W_{E2}$ = Second edge point of the object in 3D world coordinates

$W_{E1}$ = First edge point of the object in 3D world coordinates

To provide a better understanding, Step 5 and Step 6 of the algorithm in which edge points $E_1$ and $E_2$ are converted into world coordinates and the 3D vector passing through $W_{E1}$ and $W_{E2}$ are found respectively are illustrated in Figure 3.8. In the figure, 3D vector passing through two edge points in world coordinates are represented with a blue line arrow pointing left.

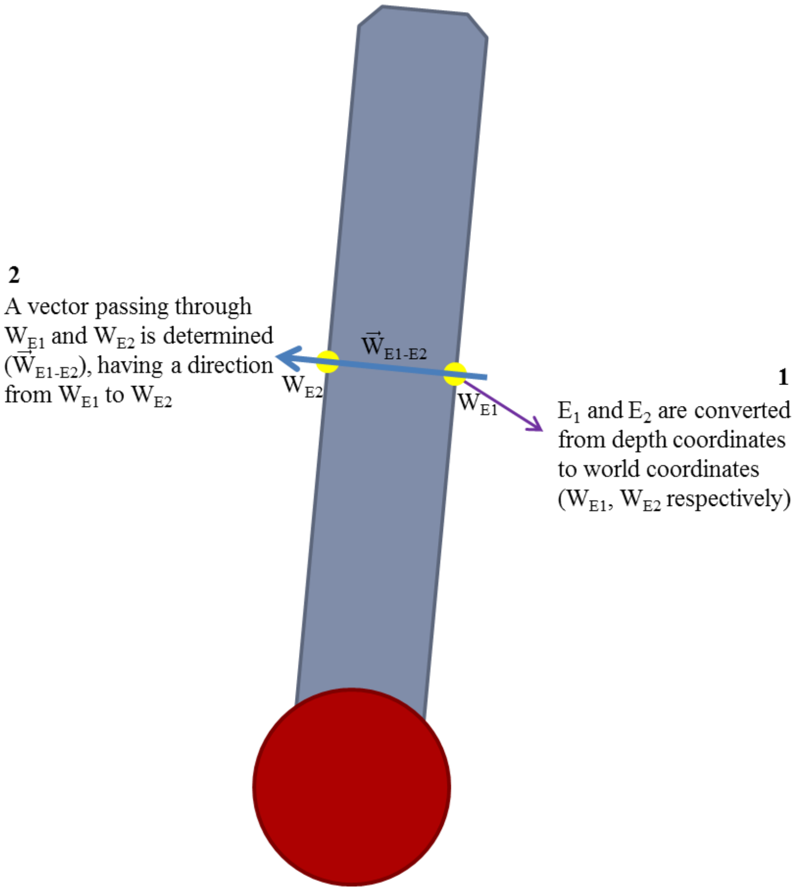


Figure 3.8: Illustration of the Algorithm Step 5 and Step 6 in which Edge Points $E_1$ and $E_2$ are Converted into World Coordinates and Vector Passing Through $W_{E1}$ and $W_{E2}$ are Found Respectively

**Step 7:** 3D Vector ($WJ_{RH\text{-}To}$) passing through $WJ_{RH}$ and $T_o$, having a direction from $WJ_{RH}$ to $T_o$ is calculated using Equation 3.5 that is presented below.

$$WJ_{RH\text{-}To} = T_o - WJ_{RH} \qquad \text{(Equation 3.5)}$$

where

$WJ_{RH\text{-}To}$ = 3D Vector from $WJ_{RH}$ to $T_o$

$T_o$ = Position of the object's tip point in 3D world coordinates

$WJ_{RH}$ = Position of the right hand joint in 3D world coordinates

**Step 8:** Orientation of the object (δ) is then found by taking the cross product of two vectors $WJ_{RH\text{-}To}$ and $W_{E1\text{-}E2}$, as presented in Equation 3.6.

$$\delta = WJ_{RH\text{-}To} \times W_{E1\text{-}E2} \qquad \text{(Equation 3.6)}$$

where

δ = Orientation of the object

$WJ_{RH\text{-}To}$ = 3D Vector from $WJ_{RH}$ to $T_o$

$W_{E1\text{-}E2}$ = 3D vector from $W_{E1}$ to $W_{E2}$

To provide a better understanding, Step 7 and Step 8 of the algorithm in which vector passing through $WJ_{RH}$ and $T_o$, and cross product of the vectors $WJ_{RH\text{-}To}$ and $W_{E1\text{-}E2}$ are found respectively are illustrated in Figure 3.9. In the figure, two vectors are represented with navy blue lines and the cross product of these two vectors is represented with a red dashed line.

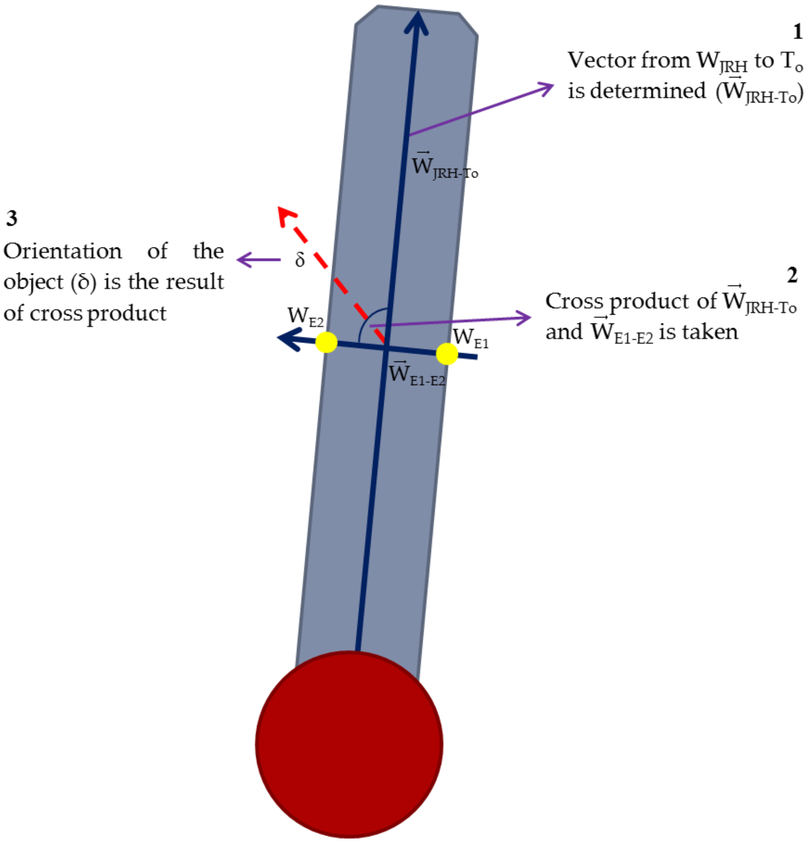


Figure 3.9: Illustration of the Algorithm Step 7 and Step 8 in which Vector Passing through $WJ_{RH}$ and $T_o$ and Cross Product of Vectors $WJ_{RH-To}$ and $W_{E1-E2}$ are Found Respectively

**Step 9:** This algorithm is performed at 30 FPS, synchronous with Kinect's frame rate and the object is incorporated into gameplay by placing the virtual sword to the hand joint of the virtual avatar in the detected direction and rotation.

To provide visual expression, Figure 3.10 is presented in which the player rotates the handheld sword in rolling axis of motion. In the figure, depth images, which are shown below their companion RGB images, involve two spheres of one blue and one red that are attached to the right and left sides of the sword respectively to show the object rotation in its local x axis. As the player rotates the sword to her left side in its local x axis as shown in the left of the figure, blue sphere is seen to be in front of the red one, indicating a rolling movement in the player's right direction.

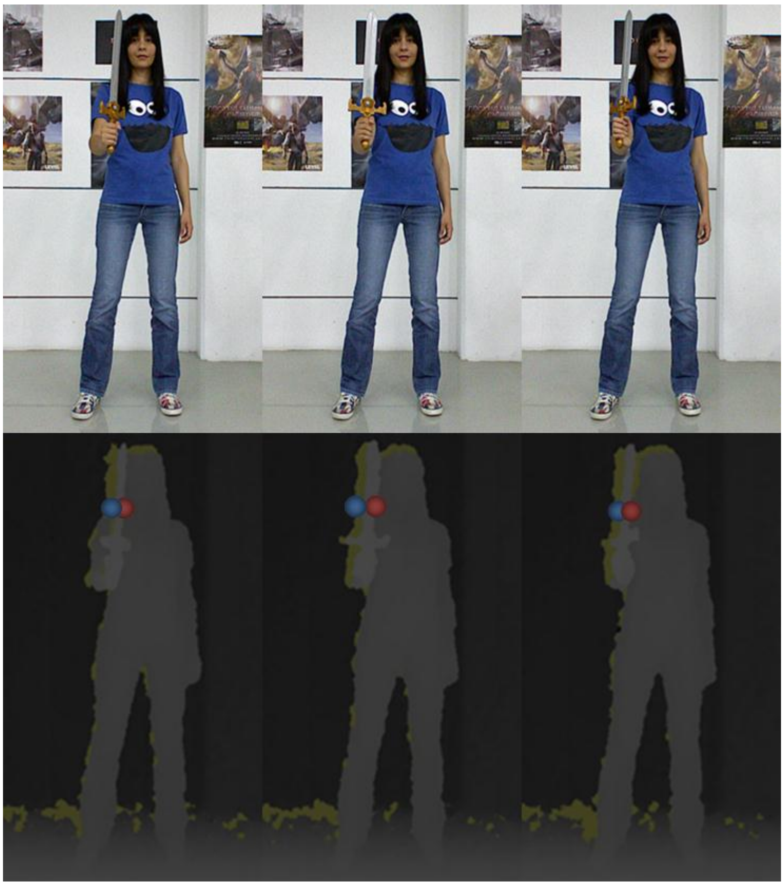


Figure 3.10: RGB and Depth Image Sets Showing the Object Rotated in Rolling Axis Left: Player Rolls the Object to Her Left Middle: Player Keeps the Object Straight Right: Player Rolls the Object to Her Right

The first algorithm that is developed for the recognition of bare hand rolling movement had the limitation of understanding only the rotation of the hand that is directed through the global z axis. The second algorithm that is developed for the recognition of handheld object rolling movement does not have this limitation. Rolling rotation of the object can be understood in all three global axes.

In Figure 3.11, the player leans the sword in the global z axis and rolling rotation of the leaned object in its local x axis can still be detected with the developed algorithm. In the figure, RGB and companion depth images are presented above and below respectively and the red and blue spheres showing rotation of the object in its local x axis are included in depth images. At the left side of the figure, the player rolls the leaned sword towards outside in its local x axis. At the middle of the figure, the player keeps the leaned sword straight. And at the right side of the figure, the player rolls the leaned sword towards inside in its local x axis.

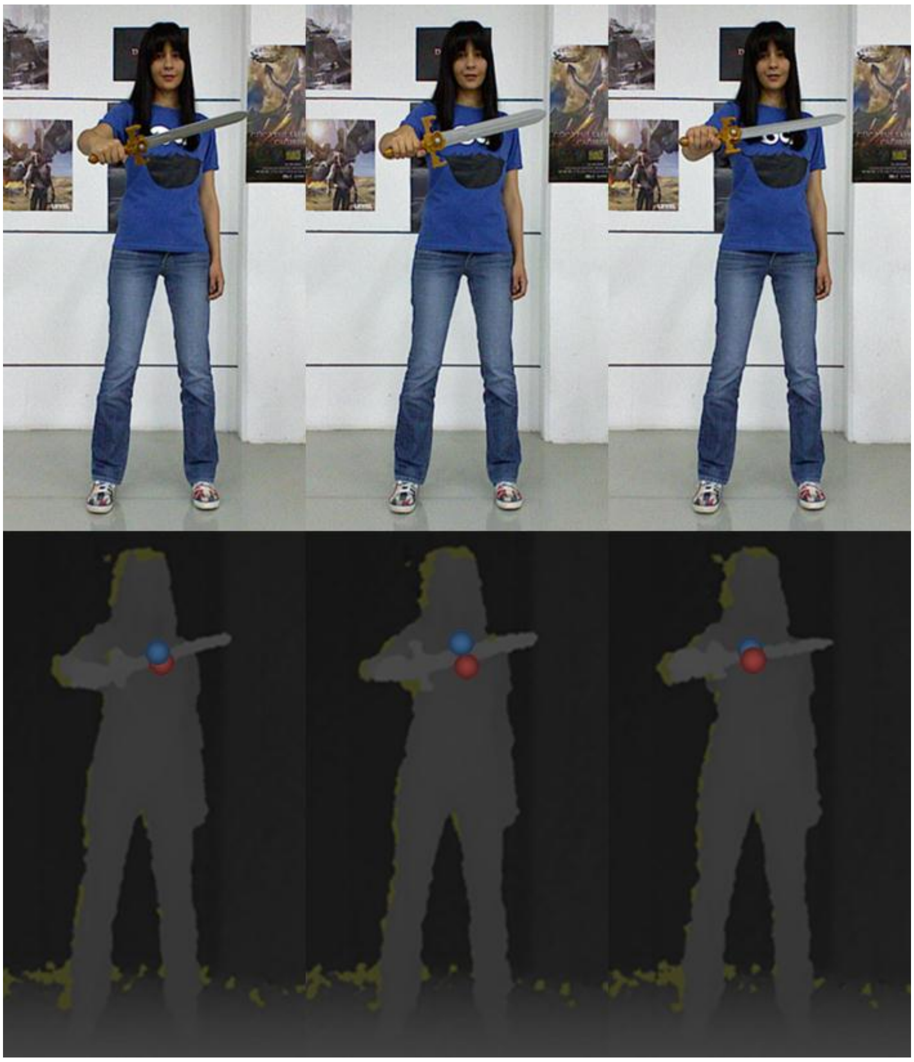


Figure 3.11: RGB and Depth Image Sets Showing the Leaned Object Rotation in Rolling Axis Left: Player Rolls the Leaned Object Outside Middle: Player Keeps the Leaned Object Straight Right: Player Rolls the Object Inside

### 3.3. Limitations

The developed algorithms have following limitations:

- Hand rotation is only recognized as the user directs her hand towards the sensor in global z axis, hand rotation is not recognized by the algorithm in other axes.
- Object rotation is limited to 30 degrees at most in local +x axis and –x axis since as the sword rotates more than 30 degrees, its width becomes too thin for Kinect to cast depth rays that fall onto it and detect its presence.
- Rotation of the objects that have width to depth ratio bigger than 3.0 can be recognized successfully by the algorithm, and the algorithm gives more reliable results as the ratio becomes bigger. Rotational motions of the objects that are in a form close to cylindrical cannot be recognized by the algorithm.
- The sword that will be used should be opaque since Kinect will not detect the presence of a transparent object.

### 3.4. Implementation

Two designed games are implemented using Unity3 [37] as game engine, Microsoft Kinect for Windows SDK [10] to gather data from the sensors and Zigfu wrapper [38] as a binder between Unity3 and Kinect to provide communication between them. The games run at 60 FPS on a Toshiba Satellite L750 Laptop Computer having Intel Core i7-2670QM CPU @ 2.20 GHz, 4 GB RAM and NVIDIA GeForce GT 525M GPU.

In the following subsections, implementation of the algorithms on two designed games is explained with provided code excerpts that are written in C# scripting language.

### 3.4.1. Hand Rotation Detection Algorithm Implementation

In this subsection, implementation of the hand rotation detection algorithm that is discussed previously in Section 3.2.1. is presented with code excerpts of the operations that are considered to be critical to the algorithm.

Code excerpt presented in Figure 3.12 performs Step 4 of the hand rotation recognition algorithm which initializes a directional search from 0 to 180 degrees as discussed previously in Section 3.2.1. Variable initializations and conversion of degree to radian is also performed here.

```
// Search is performed on directions having 0 to 180 degrees horizontal angle.
for (int angle = 0; angle < 180; angle ++)
{
// Degree is converted into radian.
angle_radian = (float)angle * Mathf.PI / 180.0f;
length = 0;
nextpixel = RightHandPixel;
```

Figure 3.12: Directional Search and Variable Initializations

Code excerpt presented in Figure 3.13 performs Step 5 of the hand rotation recognition algorithm that is discussed previously in Section 3.2.1. Linear depth detection search in the direction that is determined in Figure 3.12 is performed and two edge points of the hand in the search direction are found.

```
/* A linear search in the direction having a slope of angle degrees is performed
beginning from the center of the hand. */
for (int i = 0; i < max_pixel_number_to_search; i ++)
{
RightHandPixelIndex = (((int)Depth.yres - (int)nextpixel.y) * Depth.xres) +
                                ((int)Depth.xres - (int)nextpixel.x);
/* If depth of the candidate pixel is between a threshold of the hand depth,
search continues with the next pixel candidate. */
if ((Depth.data[RightHandPixelIndex] < RightHandDepth + depth_variant)
&& (Depth.data[RightHandPixelIndex] > RightHandDepth - depth_variant))
{
length ++;
nextpixel.x = RightHandPixel.x + (Mathf.Cos(angle_radian) * 2.0f * i);
nextpixel.y = RightHandPixel.y + (Mathf.Sin(angle_radian) * 2.0f * i);
}
else
{
/* First edge point of the hand is found and the search in the direction having
a slope of angle degrees is ended. */
i = max_pixel_number_to_search + 1;
} }
/* A linear search in the direction having a slope of -angle degrees is
performed beginning from the center of the hand. */
nextpixel = RightHandPixel;
for (int i = 0; i < max_pixel_number_to_serch; i ++)
{
RightHandPixelIndex = (((int)Depth.yres - (int)nextpixel.y) * Depth.xres) +
                                ((int)Depth.xres - (int)nextpixel.x);
/* If depth of the candidate pixel is between a threshold of the hand depth,
search continues with the next pixel candidate. */
if ((Depth.data[RightHandPixelIndex] < RightHandDepth + depth_variant)
&& (Depth.data[RightHandPixelIndex] > RightHandDepth - depth_variant))
{
length ++;
nextpixel.x = RightHandPixel.x + (Mathf.Cos(angle_radian+Mathf.PI) * 2.0f * i);
nextpixel.y = RightHandPixel.y + (Mathf.Sin(angle_radian+Mathf.PI) * 2.0f * i);
}
else
{
/* Second edge point of the hand is found and the search in the direction
having a slope of -angle degrees is ended. */
i = max_pixel_number_to_search + 1;
} }
```

Figure 3.13: Two Edge Points of the Hand in the Search Direction are Found

Code excerpt presented below in Figure 3.14 performs Step 6 of the hand rotation recognition algorithm that is discussed previously in Section 3.2.1. Distance between two edge points of the hand that are found in Figure 3.13 is calculated and stored to be as the maximum distance if it is bigger than the previous maximum distance.

```
/* Distance between two edge points is calculated and compared to the
previously stored maximum length. */
if (length > max_lenght)
{
/* If the distance is greater than the previously stored maximum distance; it is
assigned to be the new maximum distance. */
max_lenght = lenght;
```

Figure 3.14: Distance between Two Edge Points of the Hand is Found and Maximum Distance is Updated

Code excerpt presented in Figure 3.15 performs Step 7 of the hand rotation recognition algorithm that is discussed previously in Section 3.2.1. Rotation of the hand is assigned to be having the direction of the two edge points having maximum in-between distance found in Figure 3.14.

```
/* Rotation of the hand is assigned to be in the direction having a slope of
max_angle_radian. */
max_angle_radian = angle_radian;
// Radian is converted into degrees.
max_angle = (int)(max_angle_radian * 180.0 / Mathf.PI);
} }
```

Figure 3.15: Rotation of the Hand is Determined

### 3.4.2. Object Rotation Detection Algorithm Implementation

In this subsection, implementation of the object rotation detection algorithm that is discussed previously in Section 3.2.2. is presented with code excerpts of the operations that are considered to be critical to the algorithm developed.

Code excerpt presented in Figure 3.16 performs Step 3 of the object rotation recognition algorithm that is discussed in Section 3.2.2. Midpoint of the handheld object is found in depth coordinates by using 2D hand position data that is gathered from Microsoft Kinect for Windows SDK and the tip point of the object.

```
/* Midpoint of the object is found by using 2D hand position data and tip point
of the object. */
CenterPixel = ( RightHandPixel + TipPixel ) / 2.0f ;
// Slope of the line that is perpendicular to the object in depth image is found
slope = Mathf.Atan2((-(TipPixel.x - RightHandPixel.x)) , (TipPixel.y -
          RightHandPixel.y));
// Depth array representation of the midpoint is found to gather its depth value
CenterPixelIndex = ((Depth.yres-CenterPixel.y) * Depth.xres) + (Depth.xres -
                         CenterPixel.x);
// Depth value of the center point is found.
CenterDepth = Depth.data[CenterPixelIndex];
```

Figure 3.16: Midpoint of the Handheld Object in Depth Coordinates is Found

Code excerpt presented in Figure 3.17 performs Step 4 of the object rotation recognition algorithm that is discussed previously in Section 3.2.2. Linear searches from the midpoint in the positive and negative axes that is perpendicular to the direction of the object are performed to detect two edge points of the object.

```
// A search is iterated in local positive y axis in the direction having slope.
i = 1;
nextpixelindex = CenterPixelIndex;
/* As long as depth of the candidate pixel is between a threshold of the
midpoint’s depth, iteration continues. */
while ( (Depth.data[nextpixelindex] < CenterDepth + depth_variant) &&
(Depth.data[nextpixelindex] > CenterDepth - depth_variant))
{
endpoint1_depth = Depth.data[nextpixelindex];
nextpixel.x = CenterPixel.x + (Mathf.Cos(slope) * i);
nextpixel.y = CenterPixel.y + (Mathf.Sin(slope) * i);
nextpixelindex = (((int)Depth.yres - (int)nextpixel.y) * Depth.xres) +
((int)Depth.xres - (int)nextpixel.x);
i ++;
endpoint1_world = ConvertImageToWorldSpace ( new Vector3(nextpixel.x,
                          nextpixel.y, endpoint1_depth));
}
// A search is iterated in local negative y axis in the direction having slope.
i = 1;
nextpixelindex = CenterPixelIndex;
/* As long as depth of the candidate pixel is between a threshold of the
midpoint’s depth, iteration continues. */
while ( (Depth.data[nextpixelindex] < CenterDepth + depth_variant) &&
(Depth.data[nextpixelindex] > CenterDepth - depth_variant))
{
endpoint2_depth = Depth.data[nextpixelindex];
nextpixel.x = CenterPixel.x + (Mathf.Cos(slope + Mathf.PI) * i);
nextpixel.y = CenterPixel.y + (Mathf.Sin(slope + Mathf.PI) * i);
nextpixelindex = (((int)Depth.yres - (int)nextpixel.y) * Depth.xres) +
                      ((int)Depth.xres - (int)nextpixel.x);
i ++;
endpoint2_world = ConvertImageToWorldSpace ( new Vector3(nextpixel.x,
                          nextpixel.y, endpoint2_depth));
}
```

Figure 3.17: Two Edge Points of the Object are Detected

Code excerpt presented in Figure 3.18 performs Step 6 and Step 7 of the object rotation recognition algorithm that is discussed in Section 3.2.2. 3D vector from first edge point to second edge point and 3D vector from right hand joint to tip point of the object are calculated.

```
// 3D Vector from endpoint_1 to endpoint_2 is found.
EndPointsVector = endpoint2_world - endpoint1_world;

// 3D direction vector of the object from hand joint to tip point is found.
sword_direction = TipWorld - RightHandWorld;
```

Figure 3.18: 3D Vector from First Edge Point to Second Edge Point and 3D Vector from Hand Joint to Tip Point of the Object are Found

Code excerpt presented below in Figure 3.19 performs Step 8 of the object rotation recognition algorithm that is discussed in Section 3.2.2. Upward vector of the object is calculated by taking the cross product of the previously found two vectors in Figure 3.18. Orientation of the object is assigned as a quaternion ready for being incorporated into gameplay.

```
/* Upward vector of the object is found by taking cross product of the previously
found two vectors EndPointsVector and sword_direction. */
Vector3 upwards = Vector3.Cross (EndPointsVector, sword_direction);

/* Orientation of the object is assigned to a Quaternion to be incorporated into
gameplay. */
Quaternion current_rotation = Quaternion.LookRotation (sword_direction, upwards);
```

Figure 3.19: Upward Vector of the Object is Found Stored as a Quaternion

# CHAPTER 4

# EVALUATION

In this chapter, evaluation of the developed algorithms to provide the players an improved gameplay experience that are mentioned in the previous chapter is presented. First, designs of the experimental video games that are created to assess effects of the developed algorithms on player experience are presented. Following, information of the participants that attended to the user study are shared. Then, experiment procedure that is followed throughout the study is discussed. Results of the user study are presented afterwards. Finally, discussion of the results is provided at the end of the chapter.

## 4.1. Game Design

To measure effects of the two developed algorithms on gameplay, two games are designed and developed both having two versions of the same game that are controlled in two different ways. These games are presented and explained in the following subsections.

### 4.1.1. Hand Rotation Experimental Game Design

A game that incorporates the hand motions actively is desired to be designed to employ the developed algorithm in gameplay effectively. The game is designed to be having low number of variables, letting the player concentrate on the interaction without getting disturbed by complex interaction mechanics. Keeping all in mind, to find out whether incorporating hand's rolling axis of rotation into motion controlled gameplay provides an improved user experience or not, an experimental game called "Space Shape" is designed in which the player directs his/her right hand to Kinect and controls a spaceship by just rotating his/her hand in the rolling axis with the aim of passing through the obstacles having holes of different rotations. A screen capture of the game can be seen in Figure 4.1. In the figure, space ship is passing through an obstacle through its hole rotated to left in counter clockwise direction. Through the hole, upcoming obstacle can also be seen having a hole oppositely rotated to right in clockwise direction.

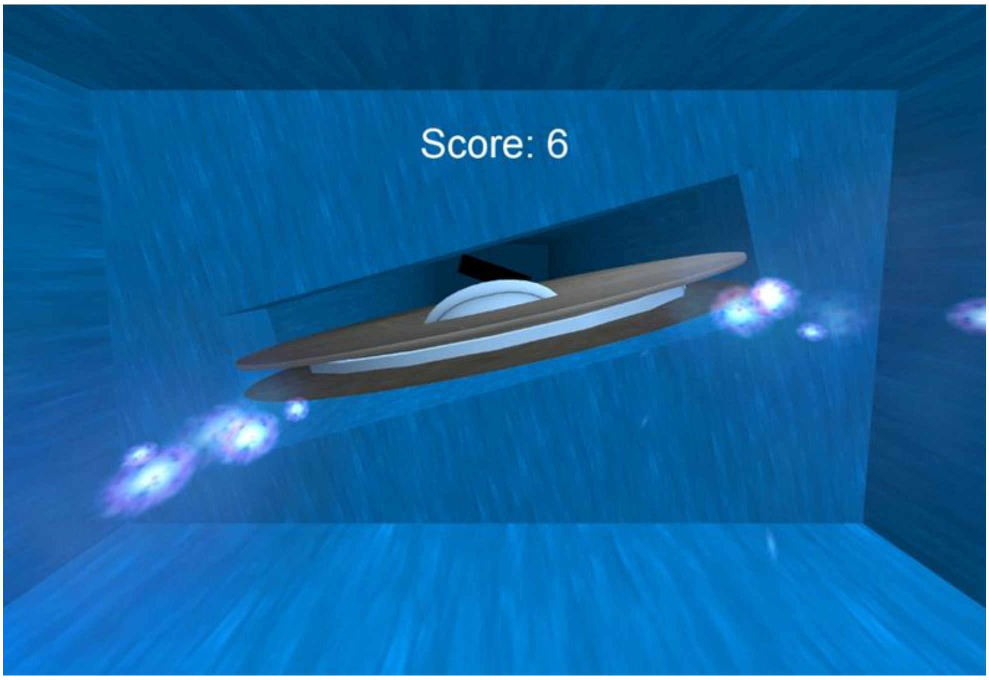
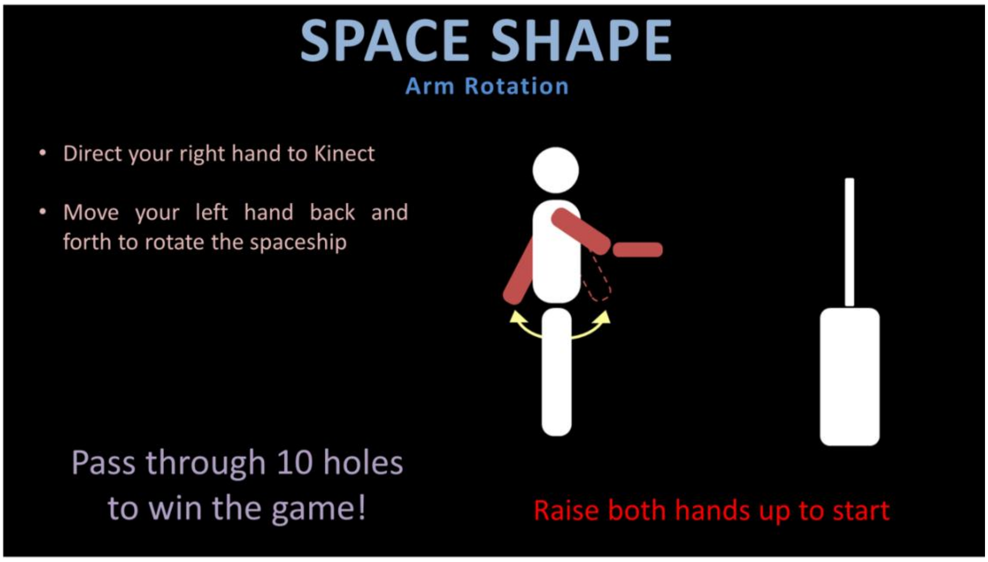


Figure 4.1: Screen Capture of the Experimental Game "Space Shape"

In the game, a simple HUD is used that informs the player on score. The HUD is intentionally kept plain, to avoid from distracting the user. Win condition of the game is determined to be passing 10 obstacles successfully, rotating the spaceship conveniently to pass through the hole without touching anywhere, to provide the player enough time to experience the interaction yet not to get him/her tired so that it would affect the second game version tested negatively. If the player cannot pass through an obstacle successfully, it does not cost any penalty points but the score is not increased either. So, to score 10 points the player must pass through 10 obstacles successfully, no matter how many obstacles he/she passes with failure.

To measure effects of the proposed controlling style on gameplay, two versions of the game are designed first one as the control version and second one as the experimental version. Control version of the game is designed to be motion controlled using the existing skeleton tracking provided by Kinect. To make the spaceship move forward, player should extend his right hand towards the sensor and to rotate the space ship in the global z axis, the player should move his/her left arm back and forth. To overcome any possible bias that may arise from unnaturalness of the designed gesture for the control version, interaction is designed to be easy in movement projection. As the player moves his/her left arm forth, the spaceship rotates to the right in clockwise direction, an instinctively expected move pretending the rotation of an imaginary line between the player's left arm and his/her body. As the player moves his/her left arm back, the spaceship rotates to the left oppositely in the counter clockwise direction. Instructions screen of the control version that is displayed at the beginning of the game is shown in Figure 4.2. In the figure, name of the game, controlling instructions supported with a visual illustration and win condition statement can be seen.



SPACE SHAPE

Arm Rotation

- Direct your right hand to Kinect
- Move your left hand back and forth to rotate the spaceship

Pass through 10 holes
to win the game!

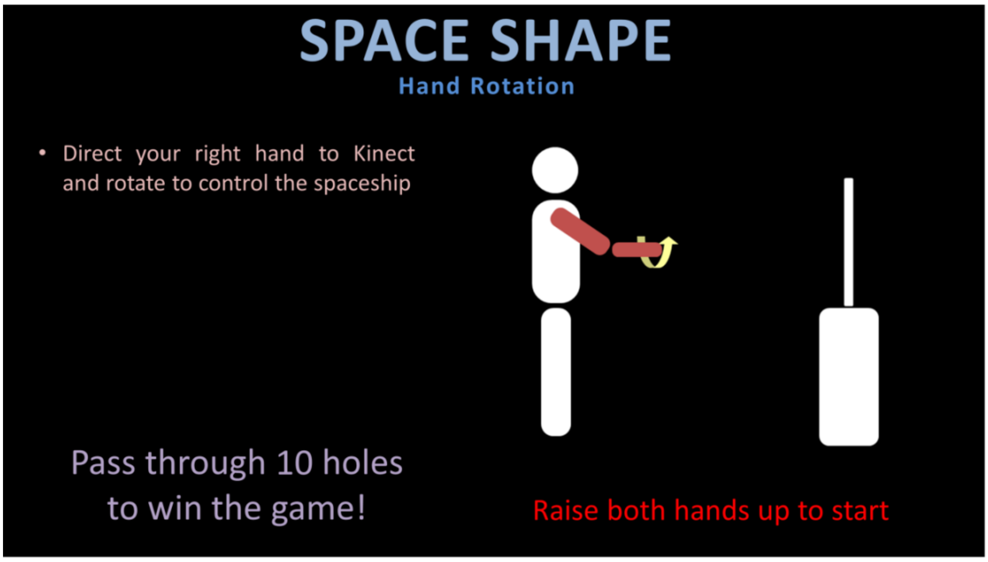
Raise both hands up to start

Figure 4.2: Instructions Screen of Space Shape's Control Version

An image showing the player controlling the game is presented in Figure 4.3. At the left side of the figure, the player keeps her left hand forth to rotate the spaceship right in clockwise direction. At the right side of the figure, the player keeps her left hand back to rotate the spaceship left in counter clockwise direction. She keeps her right hand extended towards Kinect to make the spaceship move forward and rotates the spaceship with her left arm.

Figure 4.3: A User Playing the Control Version of "Space Shape" that is Controlled with Two Arms

Experiment version of the game is designed to be controlled with the improved motion tracking proposed, using only one arm. To make the spaceship move forward, player should extend his/her right hand towards the sensor and to make the space ship rotate in the z axis, the player should just rotate his/her extended hand in z axis. Since aim of the improvement is to provide a more natural gameplay, the interaction is designed with this consideration in mind. As the player turns his right hand to the right, the spaceship also rotates to the right in clockwise direction, and controversially as the player turns his right hand to the left, the spaceship rotates to the left in counter clockwise direction, mimicking the rotation of player's hand. Instructions screen of the experiment version that is displayed at the beginning of the game is shown in Figure 4.4. In the figure, name of the game, controlling instructions supported with a visual illustration and win condition statement are presented.



SPACE SHAPE

Hand Rotation

- Direct your right hand to Kinect and rotate to control the spaceship

Pass through 10 holes to win the game!

Raise both hands up to start

Figure 4.4: Instructions Screen of Space Shape’s Experiment Version

To provide a better comprehension of the game and interaction design, a player playing the experiment version of the game which is controlled with one arm is presented in Figure 4.5. The player keeps her right hand extended towards Kinect to make the spaceship move forward. At the left side of the figure, she rolls her right hand that is directed to Kinect to the left to rotate the spaceship left in counter clockwise direction. At the right side of the figure, the player rolls her right hand to the right to rotate the spaceship right in clockwise direction.

Figure 4.5: A User Playing the Experiment Version of "Space Shape" that is Controlled with One Arm

### 4.1.2. Object Rotation Experimental Game Design

To find out whether incorporating an object's rolling axis of rotation into motion controlled gameplay provides an improved user experience or not, a modified version of the game "Hoopla" that is designed in [36] is developed in which the player rotates the sword in his/her hand to catch the disks that are thrown at different rotations. The game includes all features of "Hoopla" developed in [36] with the addition of rolling action recognition meaning that the user is able to rotate the handheld sword in rolling axis which is projected directly to the on screen virtual sword. To utilize the developed rolling recognition, disks are thrown at different rotations meaning that the player has to rotate the handheld sword conveniently in the rolling axis to be able to catch them. A screen capture of the designed game which is named "Rotate Hoopla" can be seen in Figure 4.6. In the figure, red disk seen on the upper side has white strips which all disks thrown in the game have on them to indicate their rotational directions to the player. It can be observed in the lower side of figure that the sword is rotated to the right in its local x axis. Shadow of the sword on the ground also provides a good indicator in detecting rotational direction of the object.

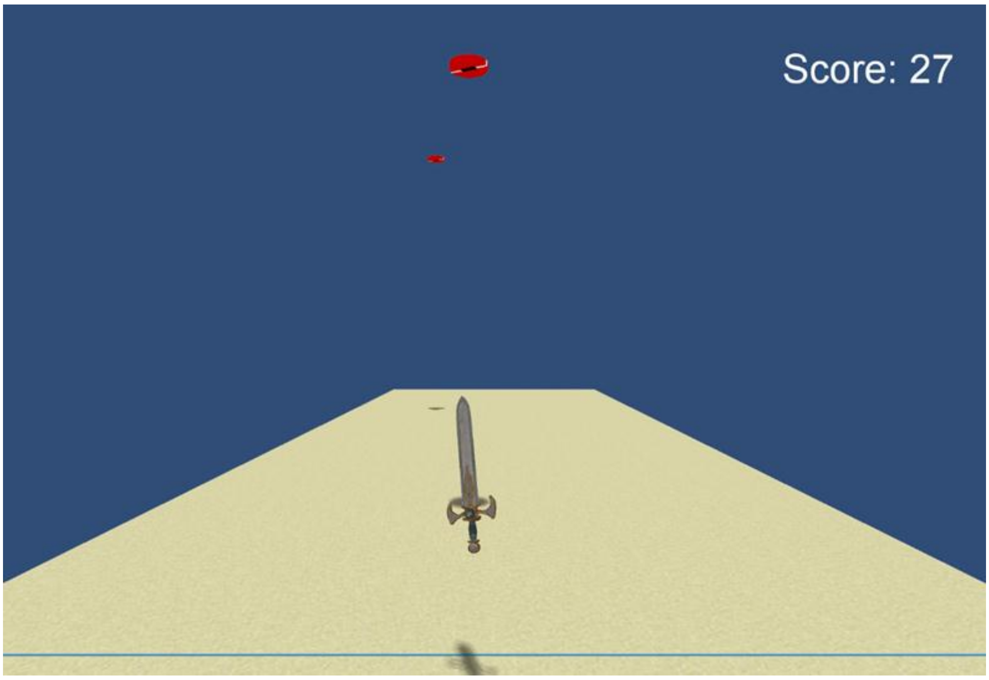
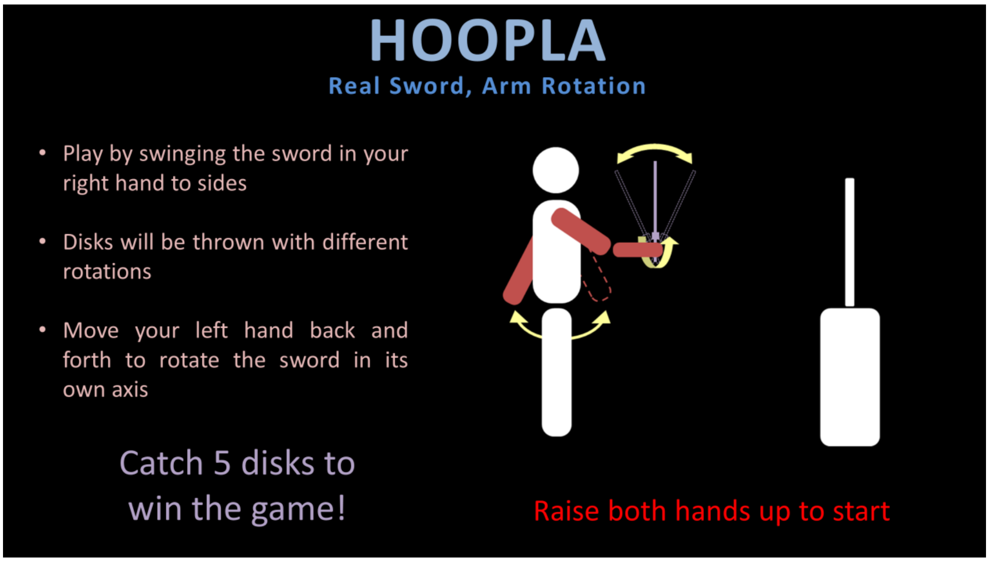


Figure 4.6: Screen Capture of the Experimental Game "Rotate Hoopla"

To measure effects of the proposed controlling style on gameplay, two versions of the game "Rotate Hoopla" are designed first one as control version and second one as experimental version. Control version of the game is designed to be controlled with the developed motion tracking algorithm of [36] and existing skeleton tracking provided by Kinect. In this version, to rotate the sword in their hands in its local x axis, the player should move his/her left arm back and forth. To overcome any possible bias that may arise from unnaturalness of the designed gesture for the control version, the interaction is designed to be easy in movement projection. As the player moves his/her left arm forth, the sword rotates to the right in clockwise direction, which is an instinctively expected move pretending the rotation of an imaginary line between the player's left arm and his/her body. As the player moves his/her left arm back, the sword rotates to the left oppositely in counter clockwise direction. Instructions screen that is displayed at the beginning of the game is shown below in Figure 4.7. In the figure, name of the game, controlling instructions supported with a visual illustration and win condition statement are presented.



**HOOPLA**

Real Sword, Arm Rotation

- Play by swinging the sword in your right hand to sides
- Disks will be thrown with different rotations
- Move your left hand back and forth to rotate the sword in its own axis

Catch 5 disks to win the game!

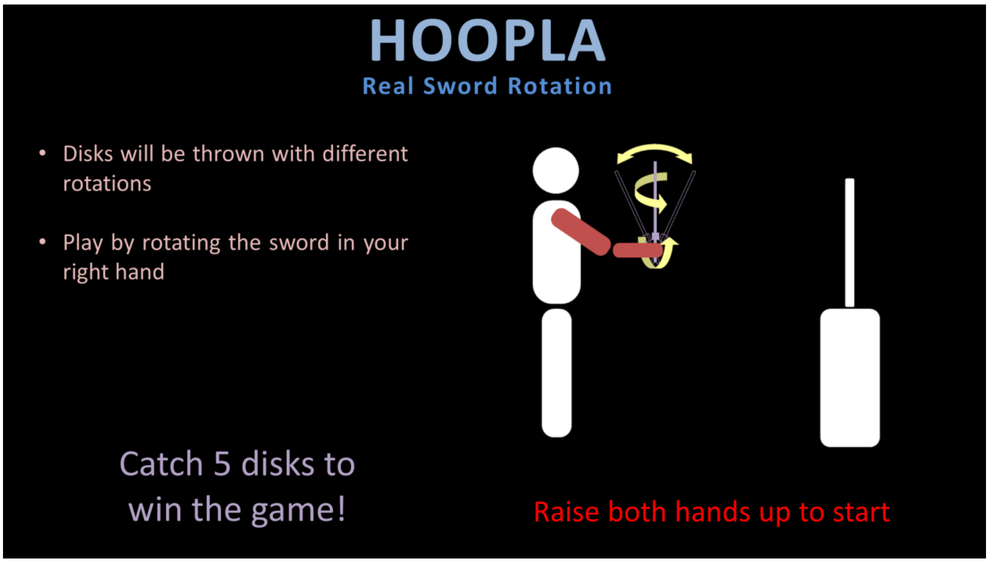
Raise both hands up to start

Figure 4.7: Instruction Screen of the Experimental Game Rotate Hoopla's Control Version

To provide a better understanding of the interaction design, a player playing the control version of the game is presented in Figure 4.8. The player carries a sword in her right hand and provides rotational motion to the virtual sword by moving her left arm forth and back. In the figure, player keeps her left arm to her back, rotating the virtual sword to the left in counter clockwise direction.

Figure 4.8: A User Playing the Control Version of the Game "Rotate Hoopla" that is Controlled with Two Arms

Experiment version of the game is designed to be controlled with the improved motion tracking proposed, recognizing roll axis of rotation of the handheld sword and enabling one handed more accurate and natural control. To rotate the sword in its local x axis, the player should rotate the sword in his/her hand in x axis. Since aim of the improvement is to provide a more natural gameplay, the interaction is designed with this consideration in mind. As the player turns the sword to the right, the virtual sword on the screen also rotates to the right, and controversially as the player turns the sword to the left, the virtual sword on the screen also rotates to the left, mimicking the rotation of the sword in the player’s hand identically. Instructions screen of the experiment version that is displayed at the beginning of the game is shown below in Figure 4.9. In the figure, name of the game, controlling instructions supported with a visual illustration and win condition statement are presented.



HOOPLA

Real Sword Rotation

- Disks will be thrown with different rotations
- Play by rotating the sword in your right hand

Catch 5 disks to win the game!

Raise both hands up to start

Figure 4.9: Instruction Screen of the Experimental Game Rotate Hoopla’s Experiment Version

A player playing the experiment version of the game is presented in Figure 4.10. The motion recognition system understands tangible sword’s rolling rotation; hence the player is able to control the game using only the arm that is holding the sword. The player provides rotational motion to the virtual sword by just rotating the sword in her hand and the rotational motion is mirrored into the virtual sword identically. In the figure, the player rotates the sword to her outside in clockwise direction and inside in counter clockwise direction from left to right respectively.

Figure 4.10: A User Playing the Experiment Version of the Game "Rotate Hoopla" that is Controlled with One Arm

### 4.2. Participant Information

A user study is performed to evaluate the outcomes of the developed system with 16 participants who played each of the developed games' control and experiment versions. Participant's ages ranged from 18 to 29, having a mean of 23 and standard deviation of 3.27. Gender distribution is 50% as 8 males and 8 females. All participants are right handed. Occupation of all participants is student, distributed as 1 high school, 11 undergraduate and 4 graduate students.

As they are asked on how often they play video games, 4 participants stated that they play video games less than once a week and 12 participants stated that they play video games more than once a week. As the participants are asked about the hours they spend playing video games, their answers ranged from 1 hour to 50 hours with a mean of 20.06 and standard deviation of 15.08. As the participants are asked about their prior experience of playing Kinect, 7 participants stated that they have prior game playing experience with Kinect and 9 participants stated that they have no Kinect experience before.

## 4.3. Experiment Procedure

Procedure of the experiment can be summarized as follows: Users are informed about the study and requested to fill out a form consisting of demographic information that's results are shared above. Then they are directed to front of the experiment set up consisting of a TV and Kinect. The participants are shown an instruction screen involving required information on how to play the game and win condition. After finishing the game, the participant is requested to fill out a questionnaire and three forms that assess his/her experience with the interaction, following a procedure similar to [39]. Detailed information on the content of questionnaire and forms are provided in Section 4.4 under relevant subsections.

As an important experimental condition, games and their controlling versions are assigned to the players randomly as follows: First, a game is determined randomly among “Rotate Hoopla” and “Space Shape”. Then a version of the selected game is assigned randomly among control version of two arms controlling and experiment version of one arm controlling. As the player finishes playing the assigned version of the assigned game, other version of the same game is presented to the player.

In the user study, version 1 refers to the control version that is controlled with two arms and version 2 refers to experiment version that is controlled with one arm achieved employing the developed algorithms. Win conditions of the games that are determined by in house gameplay tests are as follows: Rotate Hoopla requires 5 disks to be successfully caught for the player to win and Space Shape requires 10 obstacles to be successfully overcome for the player to win.

## 4.4. Results

Results of the user study are presented in the following subsections with their statistical analyses for the two games “Rotate Hoopla” and “Space Shape”.

### 4.4.1. Results of the “Space Shape” Experiment

Results of the experiment involving two different controlling styles of the game “Space Shape” are discussed in the following subsections under relevant focus groups presented with their statistical analyses.

#### 4.4.1.1. Questionnaire Results

Questions in the questionnaire conducted are designed to measure user experience in terms of enjoyment, frustration, gameplay, control, learning requirement, naturalness of the interaction, immersion, performance and effort. Some questions are adopted from the Core Elements of the Gaming Experience Questionnaire (CEGEQ) and other questions are designed to assess user experience with a focus on interaction following guidelines presented in [40].

Results of the questionnaire that is conducted to the participants are presented below in relevant groups of enjoyment, frustration, gameplay, control, learning requirement, naturalness, immersion, performance and effort. T tests are conducted to validate reliability of the questionnaire results with an alpha of 0.05.

**Enjoyment:** To assess the effect of interaction on player enjoyment, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the enjoyment score for version 1 (M = 3.646, SD = 0.785) with mean of the enjoyment score for version 2 (M = 4.208, SD = 0.898) with an alpha level 0.05 and the test was found statistically significant with a t value of -4.132 and p value of 0.0001 meaning that version 2 provided the players more enjoyment.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on enjoyment closely. As the mean of the question "I enjoyed playing the game." is compared for version 1 (M = 3.625, SD = 0.719) and version 2 (M = 4.188, SD = 0.981) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.058 and p value of 0.0287 meaning that players enjoyed playing the game while using the interaction of version 2 more.

As the mean of the question "I liked the game." is compared for version 1 (M = 3.938, SD = 0.574) and version 2 (M = 4.500, SD = 0.633) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.093 and p value of 0.0037 meaning that players liked the game more as it is controlled with the interaction of version 2.

As the mean of the question “I would play this game again.” is compared for version 1 (M = 3.375, SD = 0.957) and version 2 (M = 3.938, SD = 0.998) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.183 and p value of 0.0227 meaning that players were more likely to play the game again with interaction version 2.

**Frustration:** To assess the effect of interaction on player frustration, one question is asked to the players. As the mean of the question “I was frustrated while playing the game.” is compared for version 1 (M = 1.938, SD = 0.574) and version 2 (M = 1.688, SD = 0.479) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 1.464 and p value of 0.0819, meaning that interaction style had no effect on the frustration of players during gameplay.

**Gameplay:** To assess the effect of interaction on gameplay, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the gameplay score for version 1 (M = 3.958, SD = 0.898) with mean of the gameplay score for version 2 (M = 4.625, SD = 0.489) with an alpha level 0.05 and the test was found statistically significant with a t value of -4.283 and p value of 0.0000 meaning that version 2 provided the players a better gameplay.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on gameplay closely. As the mean of the question “It was easy to score.” is compared for version 1 (M = 3.938, SD = 0.998) and version 2 (M = 4.563, SD = 0.512) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.076 and p value of 0.0277 meaning that players scored easier with the interaction of version 2.

As the mean of the question “I understood how to play the game easily.” is compared for version 1 (M = 4.063, SD = 0.854) and version 2 (M = 4.813, SD = 0.403) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.223 and p value of 0.0028 meaning that players understood how to play the game easier while using the interaction of version 2.

As the mean of the question “The game was easy.” is compared for version 1 (M = 3.875, SD = 0.885) and version 2 (M = 4.500, SD = 0.516) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.179 and p value of 0.0229 meaning that players perceived the game to be easier with the interaction of version 2.

**Control:** To assess the effect of interaction on control, five questions are asked to the players. As these five questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 3.938, SD = 0.847) with mean of the gameplay score for version 2 (M = 4.525, SD =

0.675) with an alpha level 0.05 and the test was found statistically significant with a t value of -5.379 and p value of 0.0000 meaning that version 2 provided the players better control.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on gameplay closely. As the mean of the question "I interacted with the game easily." is compared for version 1 (M = 4.063, SD = 0.929) and version 2 (M = 4.625, SD = 0.806) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.764 and p value of 0.0072 meaning that players interacted with the game easier with the interaction of version 2.

As the mean of the question "The game was easy to control." is compared for version 1 (M = 3.938, SD = 0.772) and version 2 (M = 4.625, SD = 0.619) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.711 and p value of 0.0081 meaning that players found the interaction of version 2 easier to control.

As the mean of the question "I felt that the control was on me during the game." is compared for version 1 (M = 3.875, SD = 0.957) and version 2 (M = 4.625, SD = 0.500) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.818 and p value of

0.0065 meaning that players felt the control on them more with the interaction of version 2.

As the mean of the question “I felt free while playing the game.” is compared for version 1 (M = 3.500, SD = 0.817) and version 2 (M = 4.188, SD = 0.834) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.112 and p value of 0.0259 meaning that players felt more freedom while using the interaction of version 2.

As the mean of the question “I remember the gestures that are used for controlling the game.” is compared for version 1 (M = 4.313, SD = 0.602) and version 2 (M = 4.563, SD = 0.512) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of -1.732 and p value of 0.0519 meaning that different interaction style in two versions does not have an effect on the players’ remembering the gestures that are used in the game afterwards.

**Learning Requirement:** To assess the effect of interaction on learning requirement, two questions are asked to the players. As these two questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 2.344, SD = 0.701) with mean of the gameplay score for version 2 (M = 1.188, SD = 0.397) with an alpha level 0.05 and the test was found statistically significant with a t value of 7.400 and p value of 0.0000 meaning that version 2 imposed the players less learning requirement.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on learning requirement closely. As the mean of the question "It was difficult to learn the gestures that are required to control the game." is compared for version 1 (M = 2.250, SD = 0.856) and version 2 (M = 1.188, SD = 0.403) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 4.259 and p value of 0.0003 meaning that players had more difficulty in learning the gestures that are required for version 1.

As mean of the question "It was difficult to remember which movement to perform to do the required actions during the game." is compared for version 1 (M = 2.438, SD = 0.512) and version 2 (M = 1.188, SD = 0.403) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 6.455 and p value of 0.0000 meaning that players had more difficulty in remembering the required actions during gameplay while using interaction of version 1.

**Naturalness:** To assess the effect of interaction on naturalness, four questions are asked to the players. As these four questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1

(M = 3.734, SD = 0.821) with mean of the gameplay score for version 2 (M = 4.547, SD = 0.733) with an alpha level 0.05 and the test was found statistically significant with a t value of -6.789 and p value of 0.0000 meaning that version 2 provided the players a more natural way of interaction.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on naturalness closely. As the mean of the question "I think the interaction was natural." is compared for version 1 (M = 3.813, SD = 0.981) and version 2 (M = 4.563, SD = 0.727) using a related samples t test with an alpha level 0.05, the test was found to be statistically significant with a t value of -2.818 and p value of 0.0065 meaning that players found interaction of version 2 more natural.

As the mean of the question "Controls were intuitive." is compared for version 1 (M = 3.625, SD = 0.619) and version 2 (M = 4.563, SD = 0.727) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -4.392 and p value of 0.0003 meaning that players found controlling of version 2 more intuitive.

As the mean of the question "It was familiar for me to control the game." is compared for version 1 (M = 3.938, SD = 0.929) and version 2 (M = 4.625, SD = 0.619) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.711 and p value of 0.0081 meaning that players found controlling version 2 more familiar.

As the mean of the question "I felt that I already knew how to play the game." is compared for version 1 (M = 3.563, SD = 0.727) and version 2 (M = 4.438, SD = 0.892) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.656 and p value of 0.0012 meaning that players felt more like they priorly knew how to play the game with version 2.

**Immersion:** To assess the effect of interaction on immersion, one question is asked to the players. As the mean of the question "I forgot everything around me during gameplay." is compared for version 1 (M = 2.313, SD = 0.704) and version 2 (M = 2.625, SD = 0.806) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -1.775 and p value of 0.0481, meaning that players got more immersed into game with interaction version 2.

**Performance:** To assess the effect of interaction on performance, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 3.792, SD = 0.849) with mean of the gameplay score for version 2 (M = 4.542, SD = 0.582) with an alpha level 0.05 and the t-test was found to be statistically significant with a t value of -4.988 and p value of 0.0000 meaning that version 2 provided the players better performance.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on performance closely. As the mean of the question "I felt what was happening in the game was my own doing." is compared for version 1 (M = 4.063, SD = 0.772) and version 2 (M = 4.625, SD = 0.500) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.522 and p value of 0.0117 meaning that players felt more like they effected the outcome of the game with their actions in version 2.

As the mean of the question "I think I performed well on the game." is compared for version 1 (M = 4.063, SD = 0.854) and version 2 (M = 4.563, SD = 0.512) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -1.826 and p value of 0.0439 meaning that players felt more like they performed well on the game in version 2.

As the mean of the question "I felt like I really performed the actions in the game." is compared for version 1 (M = 3.250, SD = 0.683) and version 2 (M = 4.438, SD = 0.727) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -4.538 and p value of

0.0002 meaning that players felt more like they really performed the controlling actions in version 2.

**Effort:** To assess the effect of interaction on effort, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 2.688, SD = 0.926) with mean of the gameplay score for version 2 (M = 2.313, SD = 1.014) with an alpha level 0.05 and the test was found statistically significant with a t value of 3.186 and p value of 0.0013 meaning that version 1 required the players to spend more effort.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on effort closely. As the mean of the question "My right hand got tired during gameplay." is compared for version 1 (M = 3.250, SD = 1.126) and version 2 (M = 3.125, SD = 1.148) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 0.565 and p value of 0.2902 meaning that different interaction versions did not have an effect on the tiredness of the right hand.

As the mean of the question "My left hand got tired during gameplay." is compared for version 1 (M = 2.438, SD = 0.629) and version 2 (M = 1.563, SD = 0.512) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 5.653 and p value of 0.0000 meaning that left hand of the player got more tired in version 1.

As the mean of the question “I got tired during playing the game.” is compared for version 1 (M = 2.375, SD = 0.719) and version 2 (M = 2.250, SD = 0.578) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 0.696 and p value of 0.2487 meaning that different interaction versions did not have an effect on the overall tiredness of the player.

To make sure that participants’ answers have not affected by a possible bias of different versions motion recognition accuracy, a question regarding this issue is asked as “Motion recognition was sensitive.” As the mean of the answers is compared for version 1 (M = 4.188, SD = 0.403) and version 2 (M = 4.125, SD = 0.806) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 0.251 and p value of 0.4028 meaning that no difference is detected by the players in the accuracy of motion recognition between the two versions.

### 4.4.1.2. Mental Effort Results

At the end of each version, users are requested to fill out Subjective Mental Effort Questionnaire (SMEQ) chart, which is presented in Figure 4.11. The chart aims at measuring subjective mental effort the participant spends while performing an activity. Users are requested to point out the level which they think is most representative of the mental effort they spent during playing the version.

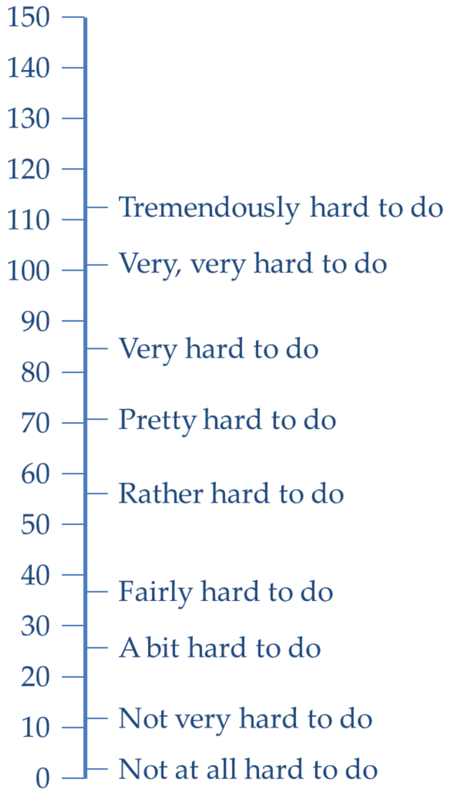


Figure 4.11: Subjective Mental Effort Questionnaire Chart

SMEQ score distribution of participants is presented in Figure 4.12 below for version 1 and version 2. Mean SMEQ scores with standard deviations are presented in Figure 4.13 for both versions of the game. Players stated the mental effort they spent for version 1 with a score having mean 34.63 and standard deviation 12.409 whereas they stated the mental effort they spent for version 2 with a score having mean 22.25 and standard deviation 10.096. The scores indicate that players spent more mental effort for version 1, which is expected since the gesture required for rotation with left arm in version 1 is not as natural as mimicking the onscreen motions of the game object in version 2.

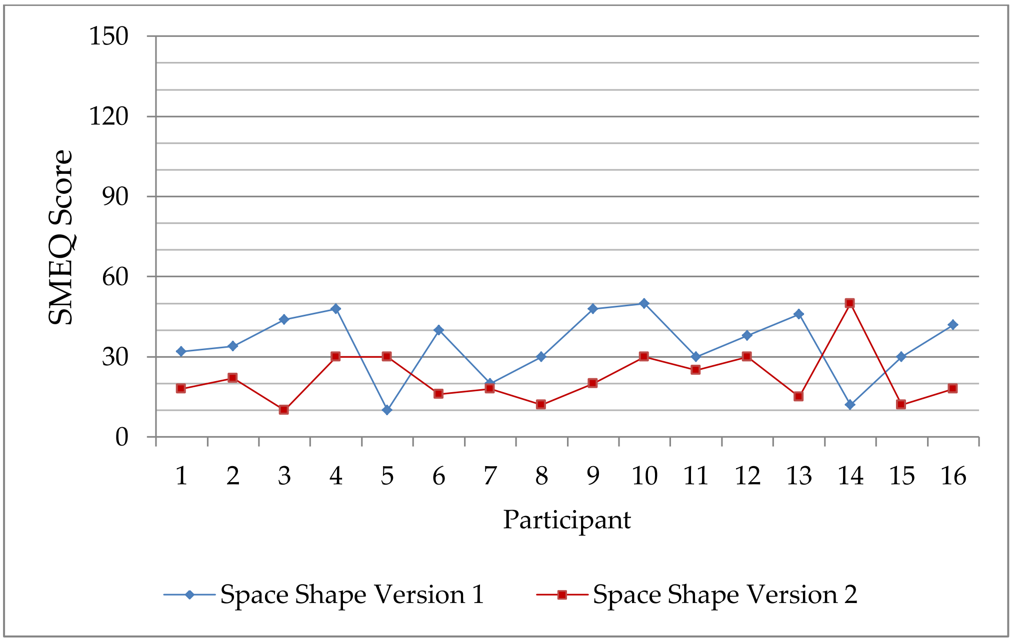


Figure 4.12: SMEQ Score Distribution for Version 1 and Version 2 of "Space Shape"

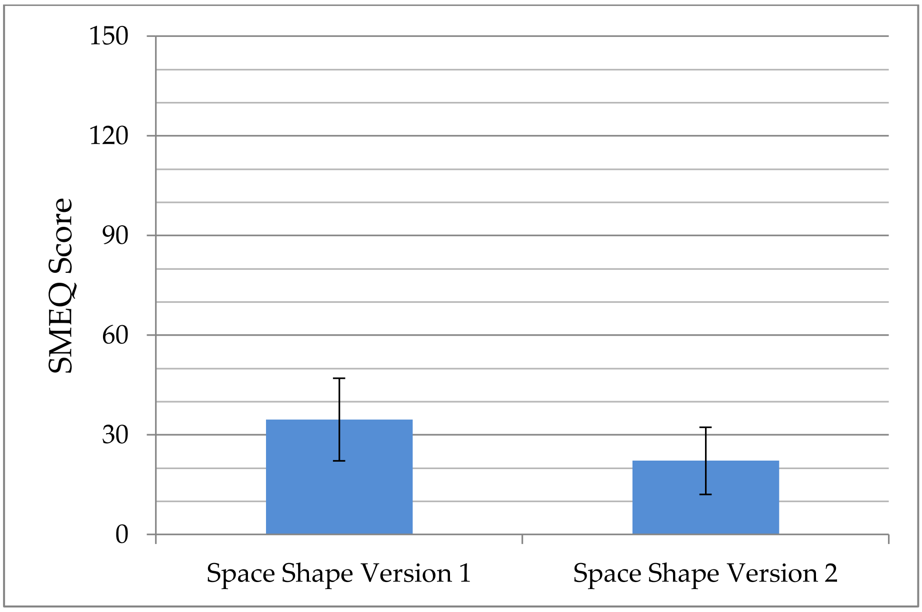


Figure 4.13: SMEQ Score Histograms for Version 1 and Version 2 of “Space Shape”

#### 4.4.1.3. User Preference

Each participant is requested to state his/her preference between interaction versions on a set of five two way scales ranging from much prefer version 1 (score 1) to much prefer version 2 (score 5). Preference score distribution is presented in Figure 4.14. User preference between version 1 and version 2 has a mean of 4.13 and standard deviation of 1.088 indicating that users have a strong preference for interaction version 2 over interaction version 1.

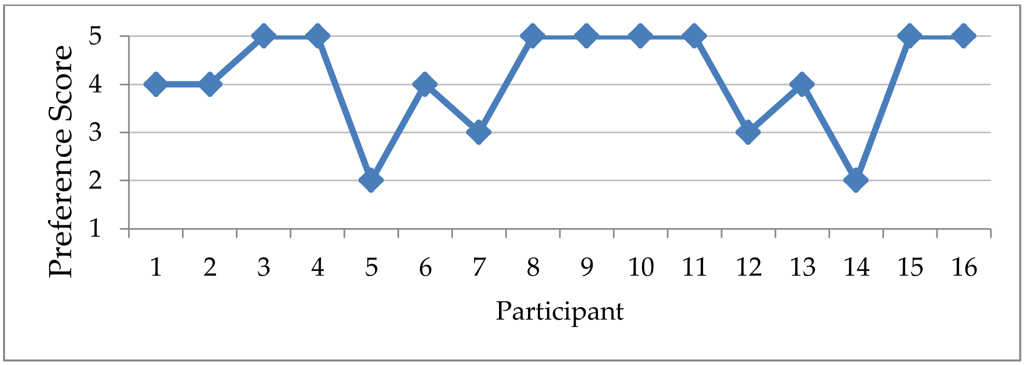


Figure 4.14: Preference Score Distribution of Participants for “Space Shape”

#### 4.4.1.4. Quantitative Results

In addition to the user responded questionnaires, there are quantitative measures that can be interpreted to find out the effect of interaction on user experience. In the following subsections, these quantitative measures are presented with their statistical analysis.

**Finishing Time:** Time it took the participants to finish the game by achieving 10 successes is measured to examine effect of interaction on gameplay. Finishing time distribution of participants is shown in Figure 4.15 and average finishing time histograms for both interaction versions are shown in Figure 4.16.

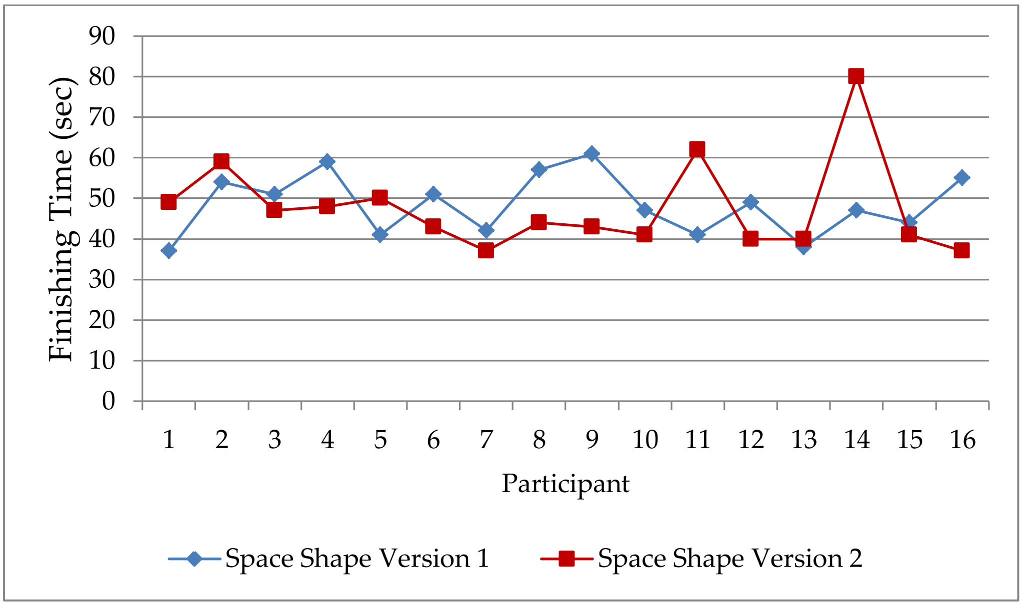


Figure 4.15: Finishing Time Distribution of Participants for "Space Shape"

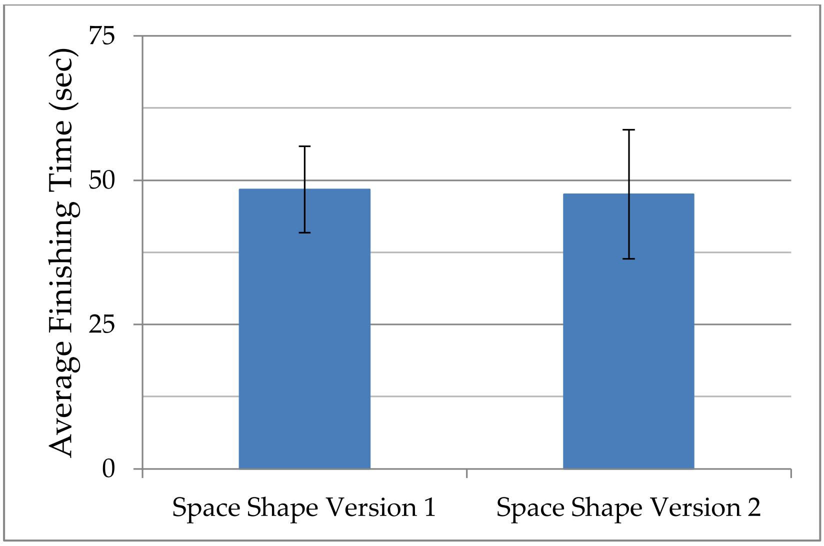


Figure 4.16: Average Finishing Time Histograms for Version 1 and Version 2 of "Space Shape"

Finishing time of the game that is played with interaction version 1 has a mean of 48.38 seconds with a standard deviation of 7.508 seconds. Similarly, finishing time of the game that is played with interaction version 2 has a mean of 47.56 seconds with a standard deviation of 11.189 seconds. The average finishing times on both versions are very close as can be observed from the histogram chart, meaning that interaction type did not have an impact on the game's finishing time for "Space Shape". To investigate significance of these results, a t test is performed. As the mean of the finishing times is compared for version 1 (M = 48.375, SD = 56.383) and version 2 (M = 47.563, SD = 125.196) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 0.232 and p value of 0.8196 meaning that no difference is detected in the players' finishing time of the game due to the interaction styles.

**Number of Fails:** Number of fails that participants made until finishing the game by achieving 10 successes is measured to examine effect of interaction on gameplay. Average number of fails histograms for both interaction versions are shown in Figure 4.17. Number of fails made with interaction version 1 has a mean of 3.13 with a standard deviation of 2.187. Similarly, number of fails that are made with interaction version 2 has a mean of 3.19 with a standard deviation of 2.926. Number of fails that are made on both versions are close as can be observed from the histogram chart, indicating that interaction type did not have an impact on the number of fails the players made.

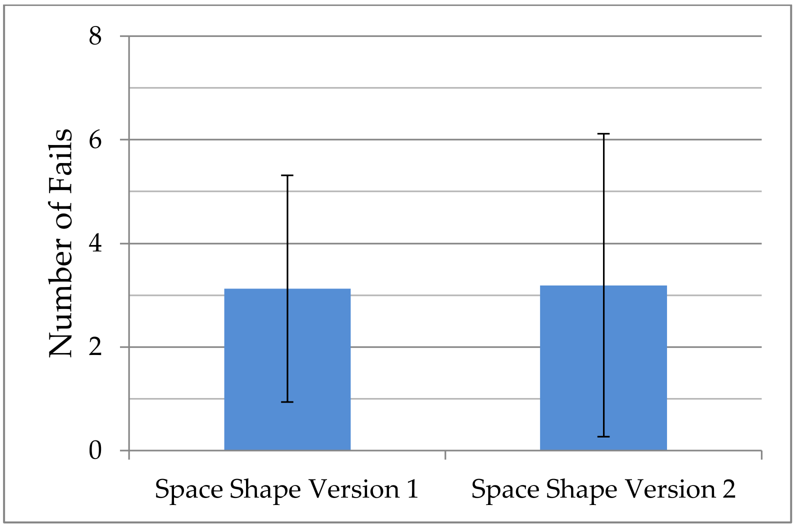


Figure 4.17: Average Number of Fails Histograms for Version 1 and Version 2 of "Space Shape"

To investigate significance of these results, a t test is performed. As the mean of the number of fails is compared for version 1 (M = 3.125, SD = 4.783) and version 2 (M = 3.188, SD = 8.563) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of -0.061 and p value of 0.9519 meaning that no difference is detected in the number of fails made due to the interaction styles.

**Average Success Time:** Average success times of the participants are measured to examine effect of interaction on gameplay. Average success time distribution of participants is shown in Figure 4.18. It can be observed from the figure that in version 1, it took more time for the players to perform successes.

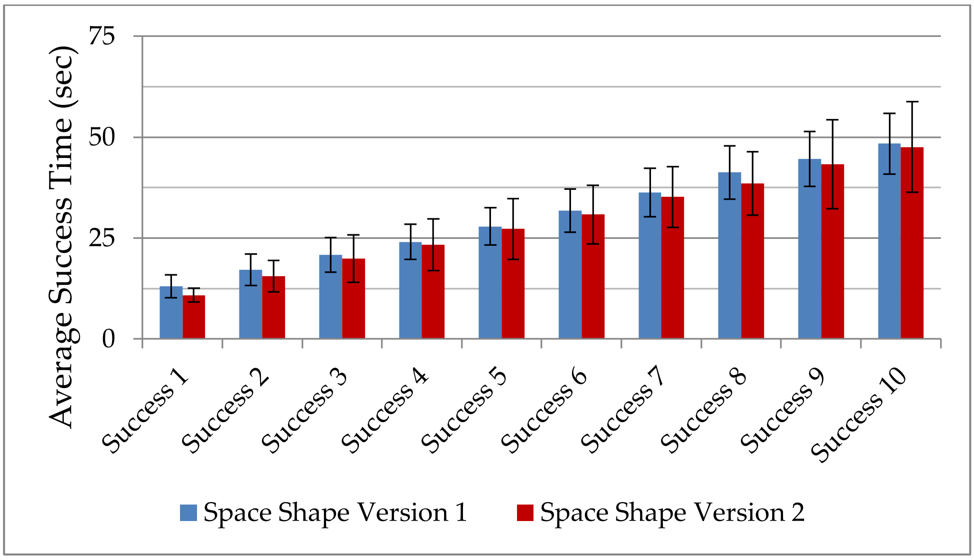


Figure 4.18: Average Success Time Histograms for Version 1 and Version 2 of "Space Shape"

Since the game has a continuous gameplay, a delay in the first success may affect the whole distribution. In order to overcome this bias, time between consecutive successes are calculated and to investigate whether interaction type has an effect on time between consecutive successes, a t test is performed. As the mean of the finishing times is compared for version 1 (M = 4.138, SD = 0.564) and version 2 (M = 4.056, SD = 1.252) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 0.232 and p value of 0.8196 meaning that no difference is detected in the time between consecutive successes due to the interaction styles.

**Directions Screen Passing Time:** Time it took the participants to pass the instructions screen is measured to examine effect of interaction on learning requirement. Average directions screen passing time histograms are shown below in Figure 4.19. Average directions screen passing time for interaction version 1 has a mean of 19.88 seconds with a standard deviation of 11.587 seconds. Similarly, average directions screen passing time of the game that is played with interaction version 2 has a mean of 15.06 seconds with a standard deviation of 6.875 seconds. It can be observed from the histogram chart that directions screen passing times are close to each other in both versions, meaning that interaction type did not have an impact on the players' directions screen passing time.

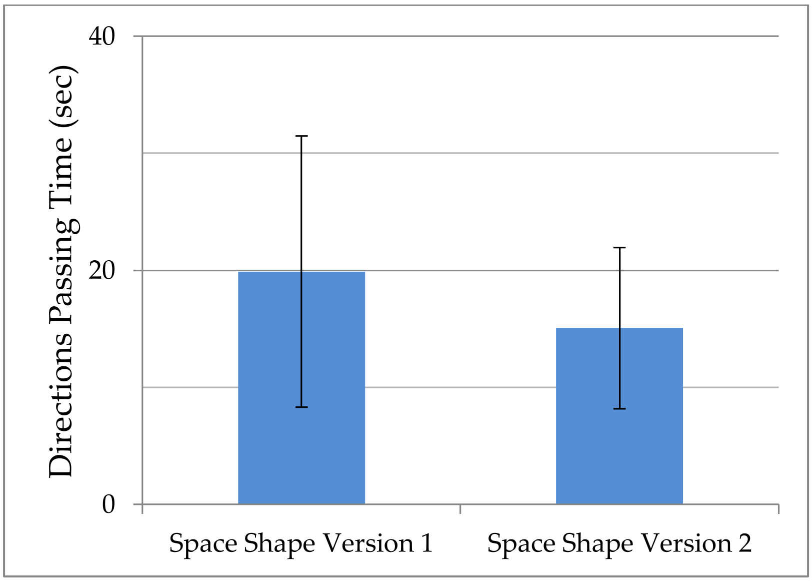


Figure 4.19: Average Directions Screen Passing Time Histograms for Version 1 and Version 2 of "Space Shape"

To investigate significance of these results, a t test is performed. As the mean of the directions screen passing time is compared for version 1 (M = 19.875, SD

= 11.587) and version 2 (M = 15.063, SD = 6.875) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 1.645 and p value of 0.0604 meaning that no difference is detected in the players' directions screen passing time due to the interaction styles.

#### 4.4.1.5. User Reviews

In addition to the questionnaires and quantitative measurements, users are requested to state the most positive and most negative experience he/she had with the experimented interaction version on the third form they filled out. Some of these user comments are stated below.

A participant mentions his most positive experience for version 2 as "I really enjoyed controlling the spaceship with the rotation of my hand, it was so familiar, like the plane imitations we perform in our childhoods I think." pointing out the familiarity of the interaction version 2 provides. Another participant states her most positive experience for version 2 as "I felt like I was in all the control of the spaceship. It was easier to control." emphasizing the feeling of being in control she experienced while controlling the spaceship with one hand in the experiment version. A participant mentions his most positive experience with the second experiment version as "It was enjoyable to feel a precise control using my hand. It was 'really' like I drove that spaceship with my hand, I liked to see the spaceship transport according to the commands I gave by just turning my hand.” pointing out the accurate control version 2 provides and expressing his enjoyment of seeing the identical motion projection of his hand into gameplay. A similar review is made by another participant who states his most positive experience with controlling version 2 as “It gives pleasure seeing your actions exactly on the screen. It was more realistic to control the plane with one hand directly making rotations, as compared to two hands control.” mentioning the enjoyment of seeing his actions projected accurately into gameplay and pointing out that he found the interaction of version 2 more realistic.

A participant stated her most positive experience with version 2 as “It was easy to control and realistic.” and stated her most negative experience for both versions as “The game was too easy, without challenges.”. A similar review from another participant is as “A different and creative game playing style, I liked it. It was very easy to play the game though.” appreciating the gameplay but criticizing the game to be too easy again. Another participant stated her most positive experience for both versions as “The idea of the game is fun and unique.”.

### 4.4.2. Results of the “Rotate Hoopla” Experiment

Results of the experiment involving two different controlling styles of the game “Rotate Hoopla” are discussed in the following subsections with their statistical analyses under relevant focus groups.

#### 4.4.2.1. Questionnaire Results

Results of the experiment involving two different controlling styles of the game “Rotate Hoopla” are discussed in the following subsections under relevant focus groups with their statistical analyses.

**Enjoyment:** To assess the effect of interaction on player enjoyment, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the enjoyment score for version 1 (M = 2.854, SD = 0.850) with mean of the enjoyment score for version 2 (M = 3.646, SD = 0.887) with an alpha level 0.05 and the test was found statistically significant with a t value of -5.322 and p value of 0.0000 meaning that version 2 provided the players more enjoyment.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on enjoyment closely. As the mean of the question “I enjoyed playing the game.” is compared for version 1 (M = 2.875, SD = 0.806) and version 2 (M = 3.875, SD = 0.885) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.303 and p value of 0.0024 meaning that players enjoyed playing the game while using the interaction of version 2 more.

As the mean of the question “I liked the game.” is compared for version 1 (M = 3.063, SD = 0.772) and version 2 (M = 3.750, SD = 0.856) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.416 and p value of 0.0145 meaning that players liked the game more as it is controlled with the interaction of version 2.

As the mean of the question “I would play this game again.” is compared for version 1 (M = 2.625, SD = 0.957) and version 2 (M = 3.313, SD = 0.873) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.905 and p value of 0.0007 meaning that players were more likely to play the game again with interaction version 2.

**Frustration:** To assess the effect of interaction on player frustration, one question is asked to the players. As the mean of the question “I was frustrated while playing the game.” is compared for version 1 (M = 2.563, SD = 1.094) and version 2 (M = 1.938, SD = 0.929) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 2.825 and p value of 0.0064, meaning that interaction style version 1 made the players frustrated more.

**Gameplay:** To assess the effect of interaction on gameplay, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the gameplay score for version 1 (M = 2.896, SD = 0.951) with mean of the gameplay score for version 2 (M = 3.583, SD = 1.028) with an alpha level 0.05 and the test was found statistically significant with a t value of -4.905 and p value of 0.0000 meaning that version 2 provided the players a better gameplay.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on gameplay closely. As the mean of the question "It was easy to score." is compared for version 1 (M = 2.188, SD = 0.834) and version 2 (M = 2.875, SD = 0.806) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.416 and p value of 0.0145 meaning that players scored easier while using the interaction of version 2.

As the mean of the question "I understood how to play the game easily." is compared for version 1 (M = 3.438, SD = 0.727) and version 2 (M = 4.188, SD = 0.655) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.503 and p value of 0.0016 meaning that players understood how to play the game easier while using the interaction of version 2.

As the mean of the question "The game was easy." is compared for version 1 (M = 3.063, SD = 0.854) and version 2 (M = 3.688, SD = 1.138) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.611 and p value of 0.0098 meaning that players perceived the game to be easier with the interaction of version 2.

**Control:** To assess the effect of interaction on control, four questions are asked to the players. As these four questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 3.219, SD = 0.934) with mean of the gameplay score for version 2 (M = 3.906, SD = 0.886) with an alpha level 0.05 and the test was found statistically significant with a t value of -4.660 and p value of 0.0000 meaning that version 2 provided the players a better control.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on gameplay closely. As the mean of the question "I interacted with the game easily." is compared for version 1 (M = 3.438, SD = 0.964) and version 2 (M = 4.063, SD = 0.998) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -1.987 and p value of 0.0928 meaning that players interacted with the game easier while using the interaction of version 2.

As the mean of the question “The game was easy to control.” is compared for version 1 (M = 2.563, SD = 1.031) and version 2 (M = 3.750, SD = 0.931) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.048 and p value of 0.0041 meaning that players found the game easier to control with the interaction of version 2.

As the mean of the question “I felt that the control was on me during the game.” is compared for version 1 (M = 3.063, SD = 0.772) and version 2 (M = 3.688, SD = 0.947) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.440 and p value of 0.0138 meaning that players felt the control on them more with the interaction of version 2.

As the mean of the question “I remember the gestures that are used for controlling the game.” is compared for version 1 (M = 3.813, SD = 0.403) and version 2 (M = 4.125, SD = 0.619) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -2.076 and p value of 0.0277 meaning that players remembered the gestures used in the game more easily afterwards with the interaction version 2.

**Learning Requirement:** To assess the effect of interaction on learning requirement, two questions are asked to the players. As these two questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 3.313, SD = 0.931) with mean of the gameplay score for version 2 (M = 2.000, SD = 0.880) with an alpha level 0.05 and the test was found statistically significant with a t value of 6.038 and p value of 0.0000 meaning that version 2 imposed the players less learning requirement.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on learning requirement closely. As the mean of the question “It was difficult to learn the gestures that are required to control the game.” is compared for version 1 (M = 3.063, SD = 0.998) and version 2 (M = 1.750, SD = 0.856) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 3.882 and p value of 0.0007 meaning that players had more difficulty in learning the gestures that were required to control the game for version 1.

As the mean of the question “It was difficult to remember which movement to perform to do the required actions during the game.” is compared for version 1 (M = 3.563, SD = 0.814) and version 2 (M = 2.250, SD = 0.856) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 4.612 and p value of 0.0002 meaning that players had more difficulty in remembering the required actions during gameplay for version 1.

**Naturalness:** To assess the effect of interaction on naturalness, four questions are asked to the players. As these four questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 3.047, SD = 1.147) with mean of the gameplay score for version 2 (M = 4.125, SD = 0.882) with an alpha level 0.05 and the test was found statistically significant with a t value of -7.038 and p value of 0.0000 meaning that version 2 provided the players a more natural way of interaction.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on naturalness closely. As the mean of the question "I think the interaction was natural." is compared for version 1 (M = 3.688, SD = 0.947) and version 2 (M = 4.375, SD = 0.957) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.467 and p value of 0.0017 meaning that players found interaction of version 2 more natural.

As the mean of the question "Controls were intuitive." is compared for version 1 (M = 3.063, SD = 0.998) and version 2 (M = 4.125, SD = 0.719) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -4.000 and p value of 0.0006 meaning that players found controlling version 2 more intuitive.

As the mean of the question "It was familiar for me to control the game." is compared for version 1 (M = 3.438, SD = 1.153) and version 2 (M = 4.188, SD = 0.750) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.000 and p value of 0.0045 meaning that players found controlling version 2 more familiar.

As the mean of the question "I felt that I already knew how to play the game." is compared for version 1 (M = 2.000, SD = 0.730) and version 2 (M = 3.813, SD = 1.047) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -4.416 and p value of 0.0003 meaning that players felt more like they knew priorly how to play the game with version 2.

**Immersion:** To assess the effect of interaction on immersion, one question is asked to the players. As the mean of the question "I forgot everything around me during gameplay." is compared for version 1 (M = 2.438, SD = 0.727) and version 2 (M = 2.813, SD = 0.655) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -1.861 and p value of 0.0413, meaning that players got more immersed into the game with version 2.

**Performance:** To assess effect of interaction on performance, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 2.771, SD = 1.036) with mean of the gameplay score for version 2 (M = 3.833, SD = 1.098) with an alpha level 0.05 and the test was found statistically significant with a t value of -6.368 and p value of 0.0000 meaning that version 2 provided the players better performance.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on performance closely. As the mean of the question "I felt what was happening in the game was my own doing." is compared for version 1 (M = 3.250, SD = 0.856) and version 2 (M = 4.063, SD = 0.772) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -4.961 and p value of 0.0001 meaning that players felt more like they effected the outcomes of the game with their actions in version 2.

As the mean of the question "I think I performed well on the game." is compared for version 1 (M = 2.563, SD = 1.153) and version 2 (M = 3.375, SD = 1.310) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.896 and p value of 0.0007 meaning that players felt more like they performed well on the game in version 2.

As the mean of the question "I felt like I really performed the actions in the game." is compared for version 1 (M = 2.500, SD = 0.966) and version 2 (M = 4.063, SD = 1.063) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of -3.830 and p value of 0.0008 meaning that players felt more like they really performed the controlling actions in version 2.

**Effort:** To assess the effect of interaction on effort, three questions are asked to the players. As these three questions are analyzed statistically using a related samples t test to compare mean of the control score for version 1 (M = 2.729, SD = 0.818) with mean of the gameplay score for version 2 (M = 2.313, SD = 1.056) with an alpha level 0.05 and the test was found statistically significant with a t value of 2.702 and p value of 0.0048 meaning that version 1 required the players to spend more effort.

Independent T-tests for the questions constituting the group are also performed to examine different effects of the interaction on effort closely. As the mean of the question "My right hand got tired during gameplay." is compared for version 1 (M = 2.563, SD = 0.814) and version 2 (M = 2.625, SD = 0.806) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of -0.269 and p value of 0.3957 meaning that different interaction versions did not have an effect on the tiredness of the right hand.

As the mean of the question “My left hand got tired during gameplay.” is compared for version 1 (M = 2.438, SD = 0.727) and version 2 (M = 1.250, SD = 0.447) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 5.216 and p value of 0.0001 meaning that left hand of the players got more tired in version 1.

As the mean of the question “I got tired during playing the game.” is compared for version 1 (M = 3.188, SD = 0.750) and version 2 (M = 3.063, SD = 0.854) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of 0.522 and p value of 0.3046 meaning that different interaction versions did not have an effect on the overall tiredness of the players.

To make sure that participants’ answers have not affected by a possible bias of different versions motion recognition accuracy, a question regarding this issue is asked as “Motion recognition was sensitive.” As the mean of the answers is compared for version 1 (M = 3.563, SD = 0.727) and version 2 (M = 3.750, SD = 0.577) using a related samples t test with an alpha level 0.05, the test was found statistically insignificant with a t value of -0.824 and p value of 0.2115 meaning that no difference is detected by the players in the accuracy of motion recognition between the two versions.

### 4.4.2.2. Mental Effort Results

SMEQ score distribution of participants is presented in Figure 4.20 for both version 1 and version 2. Mean SMEQ scores with standard deviations are presented in Figure 4.21 as histograms for both versions of the game. Players stated the mental effort they spent for version 1 with a score having mean 93.06 and standard deviation 23.399 whereas they stated the mental effort they spent for version 2 with a score having mean 78.44 and standard deviation 13.401. The scores indicate that players spent more mental effort for version 1, which is expected since the gesture required for rotation is not as natural as mimicking the onscreen motions of the game object as in version 2, incurring more mental load to the player.

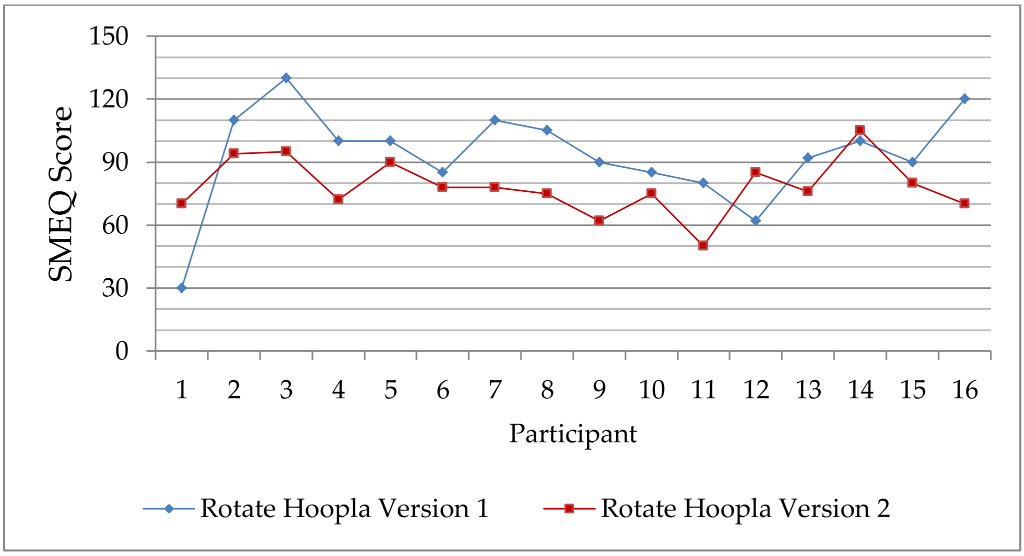


Figure 4.20: SMEQ Score Distributions for Version 1 and Version 2 of "Rotate Hoopla"

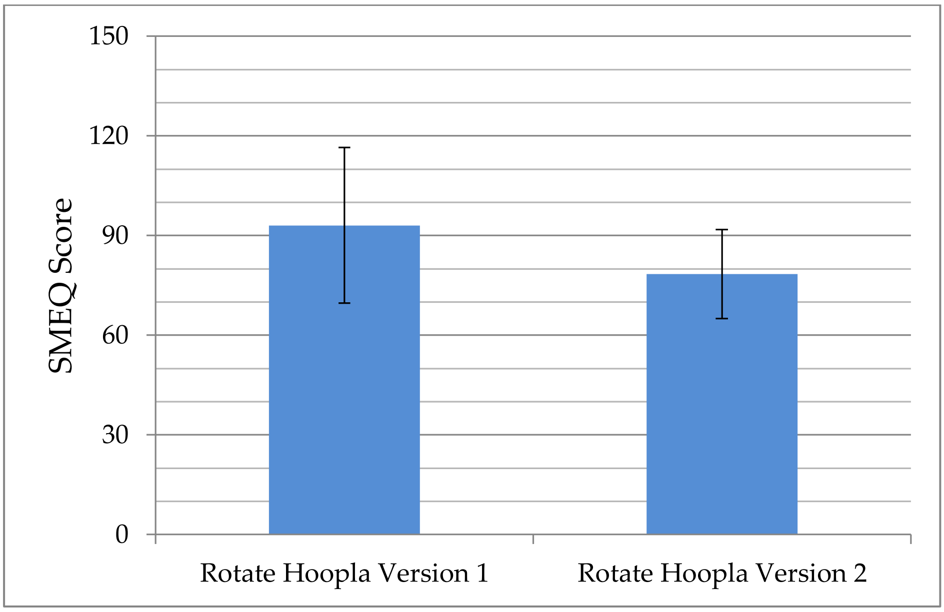


Figure 4.21: SMEQ Score Histograms for Version 1 and Version 2 of "Rotate Hoopla"

#### 4.4.2.3. User Preference

Each participant is requested to state their preference between two interaction versions on a set of five two way scales ranging from much prefer version 1 (score 1) to much prefer version 2 (score 5). Below, preference score distributions are presented in Figure 4.22. User preference between version 1 and version 2 has a mean of 3.94 and standard deviation of 1.237 indicating that users have a preference for interaction version 2 over interaction version 1.

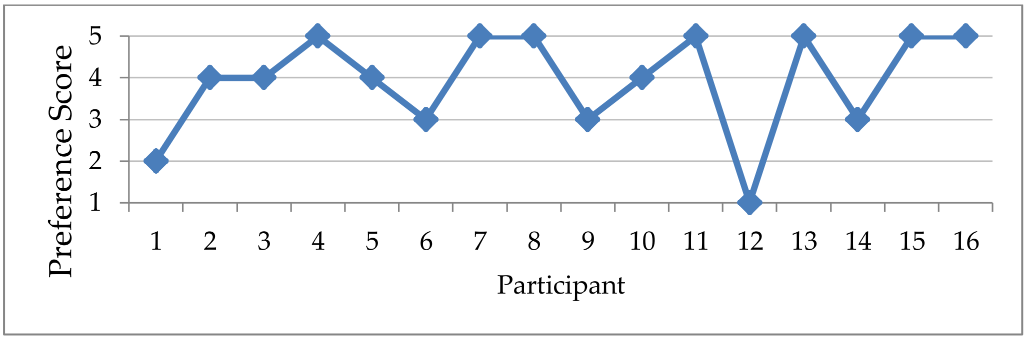


Figure 4.22: Preference Score Distribution of Participants among Version 1 and Version 2 of “Rotate Hoopla”

#### 4.4.2.4. Quantitative Results

In addition to the user responded questionnaires, there are quantitative measures that can be interpreted to find out the effect of interaction on user experience. In the following subsections, these quantitative results are presented with their statistical analysis.

**Finishing Time:** Time it took the participants to finish the game by achieving 5 successes is measured to examine effect of interaction on gameplay. Finishing time distribution of participants is shown in Figure 4.23 and average finishing time histograms for both interaction versions are shown in Figure 4.24. Finishing time of the game that is played with interaction version 1 has a mean of 199.25 seconds with a standard deviation of 75.839 seconds. Finishing time of the game that is played with interaction version 2 has a mean of 144.81 seconds with a standard deviation of 48.862 seconds, similarly. The average finishing times on both versions are considerably different as can be observed from the histogram chart, meaning that interaction version 2 provided the players to finish the game quicker.

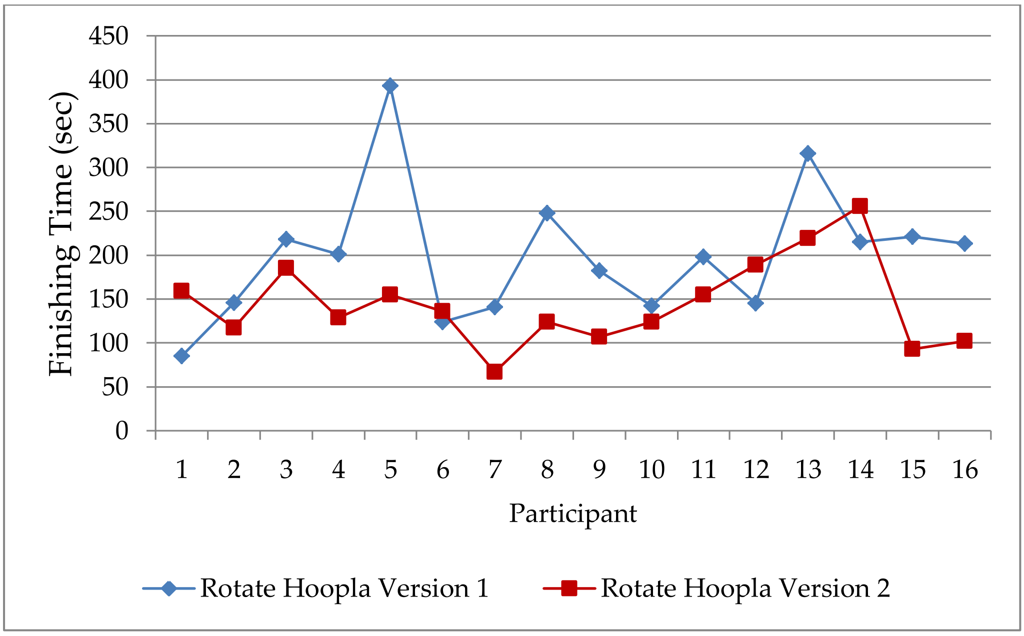


Figure 4.23: Finishing Time Distribution of Participants for Version 1 and Version 2 of "Rotate Hoopla"

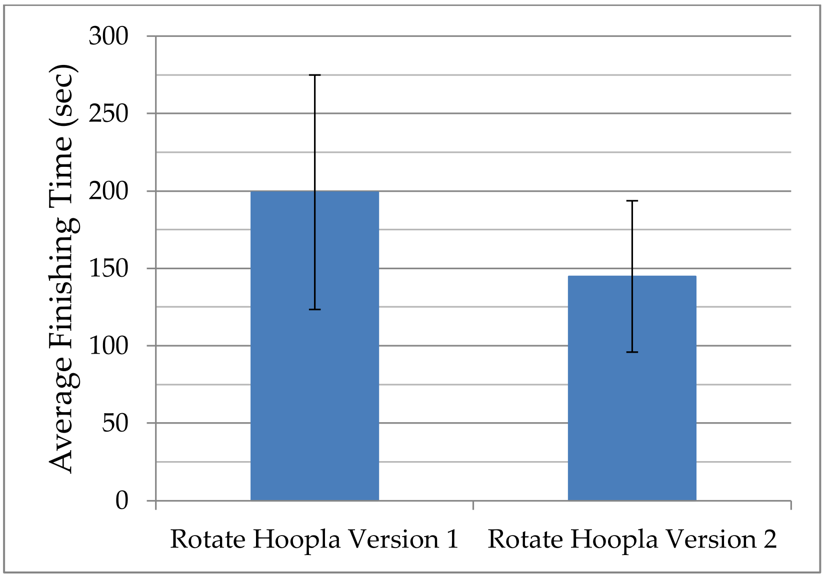


Figure 4.24: Average Finishing Time Histograms for Version 1 and Version 2 of "Rotate Hoopla"

To investigate significance of these results, a t test is performed. As mean of the finishing times is compared for version 1 (M = 199.25, SD = 75.839) and version 2 (M = 144.81, SD = 48.862) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 2.775 and p value of 0.0141 meaning that players finished the game more quickly while playing with interaction version 2.

**Number of Fails:** Number of fails that participants made until finishing the game by achieving 5 successes is measured to examine effect of interaction on gameplay. Average number of fails histograms for both interaction versions are shown in Figure 4.25. Number of fails made with interaction version 1 has a mean of 59.63 with a standard deviation of 25.3479 whereas number of fails made with interaction version 2 has a mean of 41.50 with a standard deviation of 16.305. Number of fails that are made on two versions are considerably different as can be observed from the histogram chart, indicating that version 1 caused the players to make more number of fails.

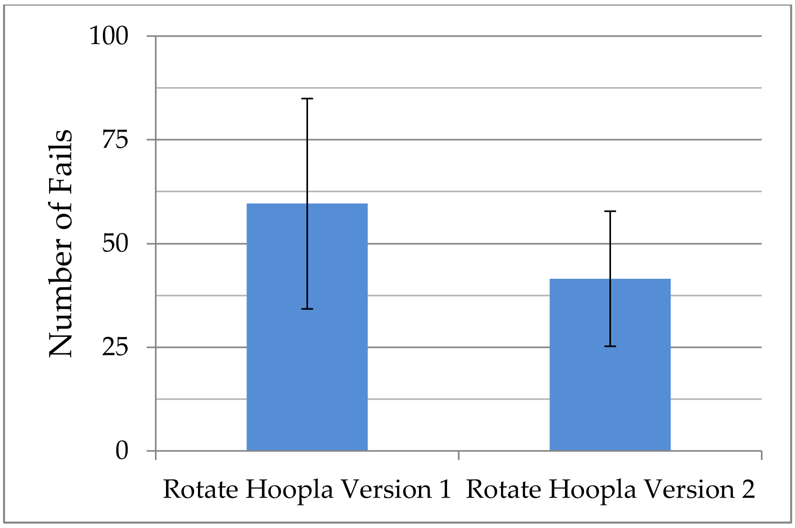


Figure 4.25: Average Number of Fails Histograms for Version 1 and Version 2 of “Rotate Hoopla”

To investigate the significance of these results, a t test is performed. As means of the finishing times are compared for version 1 (M = 59.63, SD = 25.3479) and version 2 (M = 41.50, SD = 16.305) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 2.768 and p value of 0.0144 meaning that players made more number of fails to finish the game while playing with interaction version 1.

**Average Success Time:** Average success times of the participants are measured to examine effect of interaction on gameplay. Average success time distribution of participants is shown below in Figure 4.26. It can be observed from the figure that in version 2, it took considerably less time for the players to perform successes. But since the game has a continuous gameplay, a delay in the first success may affect the whole distribution. In order to overcome this bias, time between consecutive successes are calculated and to investigate whether interaction type has an effect on time between consecutive successes, a t test is performed. As means of the average success times are compared for version 1 (M = 37.850, SD = 15.168) and version 2 (M = 26.963, SD = 9.772) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 2.775 and p value of 0.0141 meaning that time between consecutive successes were less in interaction version 2.

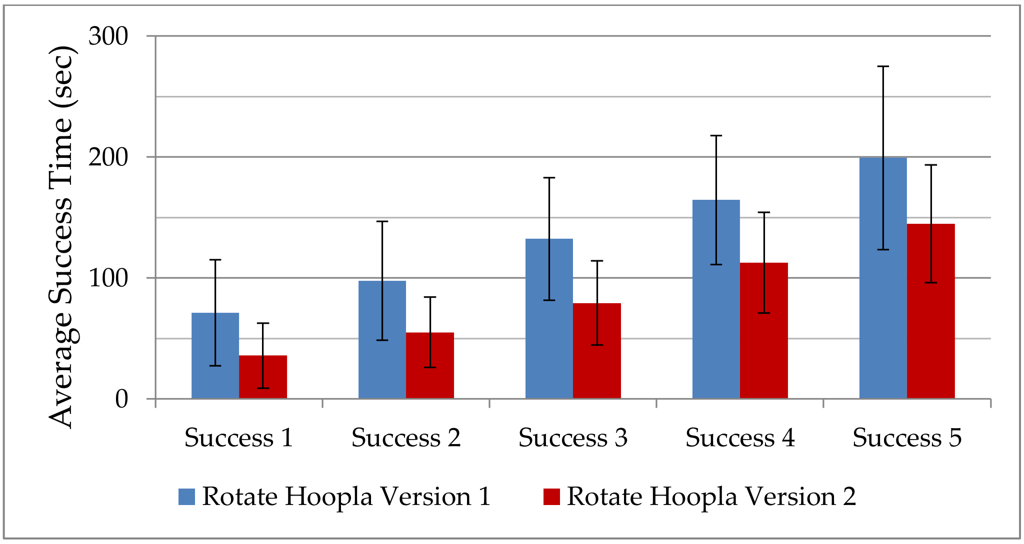


Figure 4.26: Average Success Time Histograms for Version 1 and Version of "Rotate Hoopla"

**Directions Screen Passing Time:** The time it took the participants to pass the instructions screen is measured to examine effect of interaction on learning. Average directions screen passing time histograms are shown below in Figure 4.27. Average directions screen passing time for interaction version 1 has a mean of 36.19 seconds with a standard deviation of 11.473 seconds. Average directions screen passing time for interaction version 2 has a mean of 29.88 seconds with a standard deviation of 11.063 seconds. It can be observed from the histogram chart that there is a difference between directions screen passing times of version 1 and version 2, meaning that users spent more time learning instructions of version 1.

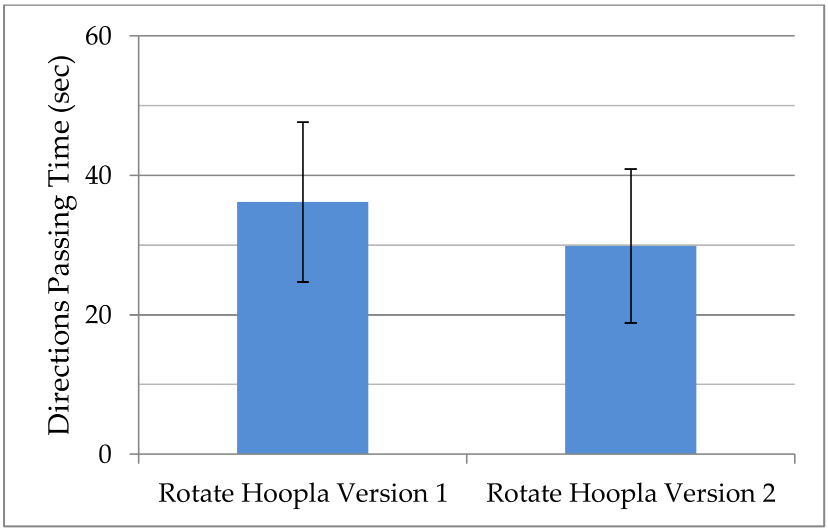


Figure 4.27: Average Directions Screen Passing Time Histograms for Version 1 and Version 2 of "Rotate Hoopla"

To investigate significance of these results, a t test is performed. As the mean of the directions screen passing time is compared for version 1 (M = 36.19, SD

= 11.473) and version 2 (M = 29.88, SD = 11.063) using a related samples t test with an alpha level 0.05, the test was found statistically significant with a t value of 2.316 and p value of 0.0176 meaning that the players spent more time before passing directions screen in interaction version 1.

#### 4.4.2.5. User Reviews

In addition to the questionnaires and quantitative measurements, users are requested to state the most positive and most negative experience he/she had with the experimented interaction version. Some of these user comments are stated below.

A participant stated his most positive experience on version 2 as "Being able to turn the sword was a more realistic way of playing, I really much enjoyed having so much control in my hand." pointing out the realism and control power the interaction provided. Another participant stated his most positive experience with version 2 as "It was more like I was playing it, I was scoring." again mentioning the realism of the interaction version 2 provided. A participant expressed her opinions on two versions as "Imaginary sword game was too difficult and it felt bad to be unsuccessful. Real sword was better, more realistic.".

From another point of view, a participant stated her experience on two versions as “Real Sword version is more realistic and there is more variety in the playing, I certainly prefer that one. In Imaginary Sword version, it was hard for me to relate the left hand with the sword in my right hand, I didn’t like it” mentioning the difficulty in mental interpretation of the actions performed in version 1. Another participant stated his most negative experience with version 2 as “In the arm rotated version, it was difficult to condition my brain not to turn the sword in my right hand that I was seeing on screen but rather to use my other arm for rotation. I automatically turned the sword though it had not recognized.” pointing out the difficulty he had experienced with mental projection of the actions in version 1. The same participant stated his most positive experience with version 2 as “In the real sword version, the game was like more than half easier to me, since that bothering additional left hand motion was eliminated and only the one I performed automatically in the other version remained. It was much better, much realistic.” expressing the naturalness and realism the interaction provides.

Related with the mental load of version 1, a participant stated that “In imaginary sword game, I forgot which side to direct my arm to rotate the sword in the direction I wanted. It was very difficult.” and he stated his most positive experience with version 2 as “It was easier to control the rotation with the real sword.”. Another participant stated his opinions on two versions as “I had coordination difficulty in Imaginary Sword Version.” and “In Real Sword Version movements were more realistic and comfortable.” mentioning the mental movement projection problem that he encountered with version 1. A participant reported a similar review stating his most negative experience for version 1 as “I spent too much mental effort such that I could not enjoy the game while thinking about the correct rotation of my left arm.” and expressing his most positive experience with version 2 as “Using right hand for both catching the disks and rotating the sword is much easy and realistic.”.

## 4.5. Discussion

Results of the user study indicate that developed algorithms provide the players a more natural way of interaction, requiring less mental effort and learning requirement, in which they feel more control over gameplay. User study results are discussed in the following subsections for the two games “Space Shape” and “Rotate Hoopla”. The section ends with presentation of the study limitations.

### 4.5.1. “Space Shape” Results Discussion

Results of the user study indicate that version 2 of “Space Shape” provided the players more enjoyment which is expected since it provides a more natural and familiar experience which the players may enjoy more by seeing the real time projections of their movements into gameplay. It is found out by the study that players found the interaction version 2 more intuitive, familiar and natural which is expected since human actions are projected identically into gameplay in version 2, providing a more natural and innate interaction. The interaction effect on player frustration is found to be insignificant which is not expected since version 2 requires less mental load and does not involve any learned gestures. The insignificancy may arise from the easiness of the game which the users also mentioned in their reviews. Since there are not many variables or complex interaction mechanics involved in the experimental game to let the user focus on interaction, players may not have been frustrated in any version, accomplishing the win condition easily. Although the game is designed to be easy, effects of interaction difference is reflected on ease of interaction and scoring. It is found in the study that players scored more easily with the interaction version 2 which is expected since additional learned gesture which is present in version 1 is eliminated in version 2 which provides a more natural way of interaction and easier projection of desired player actions into gameplay. It is also found that players interacted with the game easily with version 2, finding the interaction easier which is expected due to the same reasons of easier scoring. It is found out by the user study that players perceived the game to be easier with the interaction version 2 which may not be desirable for a game since it may bore the players and lose them. So, it should be kept in mind during motion controlled game design that realistic interaction with a simple gameplay may be perceived too easy by the players and should better be overcome with additional challenges designed. It is also found out by the study that version 2 provided the players better performance which is expected since it provides easier mental motion projection using which the players may be able to perform better.

Results of the user study also revealed that players understood how to play the game easily with version 2 which is expected since there are fewer variables and more natural motion controlling in version 2. Controlling the rotation with the additional hand may confuse the players more as compared to an identical projection of actions version 2 provides. Similarly, it is found out in the study that version 2 provided the players less learning requirement and mental load to remember the gestures in gameplay which is expected since there are no additional learned gestures in version 2. On the other hand, results of the study revealed that different interaction style in two versions did not have an effect on the players' remembering the gestures that are used in the game afterwards which is not expected since there are more gestures in version 1. It may be due to the low number of additional gesture assigned in version 1. Since there are mot many gestures assigned to the player in both versions, interaction version may not have been affected the users remembering used gestures afterwards. As the number of gestures assigned to the player increase, this result may change.

It is found out by the results of the study that the players felt more control on them while using interaction of version 2 which is expected since the players see identical motion projection which creates the feeling of control and being in charge to realize that their own actions affect the gameplay completely. Results of another question revealed that players also felt that they really performed the actions more with version 2 which is expected again due to the identical motion projection. It is also found out that users felt more freedom with version 2 which may be caused from not to be bounded by additional gestures as in version 1 and identical motion projection. In the game, a spaceship flies which the user controls with hand rotation, which may create a feeling of freedom on the user too.

It is found out by the study that players got more immersed into game with interaction version 2 but the test resulted in a big alpha value of 0.0481. Identical motion projection is expected to create more immersion on the player but the high value of alpha can be explained with the game's being so easy to immerse the player. As more complex interactions are designed and tested, immersion effect may be measured differently.

It is found out by the study that right hand tiredness is not changed with two versions which is expected since right hand is extended forward in both versions and rotational motion of an extended hand does not incur considerable amount of additional physical load. Controversially, left hand tiredness did change with the two versions, version 1 causing left hand of the players more tiredness which is expected since left hand is employed in version 1 whereas is not employed in version 2. Different interaction versions did not have an effect on the overall tiredness of the player which is not expected since version 1 requires more actions to be performed but can be explained with the game's having an easy interaction mechanics, not causing the player tiredness in both versions. If the learned gesture used in version 1 to rotate the spaceship were designed to be different, more tiring such as moving the hand up and down, tiredness result may be affected from this change.

More mental effort results in SMEQ for version 1 is higher than version 2 which is expected since there are more variables and gestures in version 1 whereas there is less to think in version 2 since the spaceship mimics player's movements. Users stated a strong preference for interaction version 2 over version 1 meaning that they are more pleased with the developed motion controlling which provides a more natural and accurate way of interaction.

As the quantitative measures are examined, no difference is detected in the players' finishing time of the game due to the interaction styles which can be explained due to game's being easy and learned gesture's being designed reasonable so that interaction versions did not make the players struggle with the game. No difference is detected in number of fails made due to the interaction styles which can be explained again by control versions having been designed not to challenge the player. No difference is detected in the time between consecutive successes between two interaction styles which can be explained by the same reasons. Results revealed that no difference is detected in the players' directions screen passing time due to the interaction styles which is not expected since version 1 presents a learned gesture but can be explained by the few number of gestures present in version 1 and their having been assigned as natural and meaningful as possible.

Positive user reviews on the developed interaction version 2 were on familiarity, feeling of being in control, accuracy, ease of control, realism and identical motion projection. Some of the users also stated that they liked the game and it had an enjoyable style. Negative user reviews were on the game's being too easy and difficulty of motion projection with version 1.

### 4.5.2. "Rotate Hoopla" Results Discussion

Results of the user study indicate that version 2 of "Rotate Hoopla" provided the players more enjoyment which is expected since it provides a more natural and familiar experience which the players may enjoy more by seeing the real time projections of their movements into gameplay. It is found out by the study that players found the interaction version 2 more intuitive, natural and familiar which is expected since human actions are projected identically into gameplay in version 2, providing a more natural and intuitive interaction. Effect of interaction on player frustration is found to be significant with version 1 getting the players more frustrated during gameplay which is expected since version 2 requires less mental load and does not involve any learned gestures whereas players need to learn the rotation gesture in version 1 which does not have identical motion projection. Player frustration may most probably have arisen from motion projection difficulty encountered in version 1. It is found out in the study that players scored more easily with the interaction version 2 which is expected since additional learned gesture which is present in version 1 is eliminated in version 2 which provides a more natural way of interaction and easier projection of desired player actions into gameplay. It is also found out that players interacted with the game easily using version 2, which is expected since a more natural interaction is provided in this version. It is found out by the user study that players again perceived the game to be easier with the interaction version 2 although the game involves more spatial variables than “Space Shape”. It is also found out by the study that version 2 provided the players better performance which is expected since it provides easier mental motion projection enabling the players perform better.

Results of the user study also revealed that players understood how to play the game easily with version 2 which is expected since there are fewer variables and more natural motion controlling in version 2. Controlling the rotation with the additional hand may confuse the players more as compared to an identical projection of actions version 2 provides. Similarly, it is found out in the study that version 2 provided the players less learning requirement and mental load to remember the gestures in gameplay which is expected since there are no additional learned gestures in version 2, the player interacts with the game by using only the handheld sword. Results of the study revealed that players remembered the gestures used in the game more easily afterwards with the interaction version 2 which is expected due to the greater number of gestures present in 1.

It is found out by the results of the study that the players felt more control on them while using interaction of version 2 which is expected since the players see identical motion projection which creates the feeling of control and being in charge to realize that their own actions affect the gameplay completely. Results of another question revealed that players also felt like that they really performed the actions more with version 2 which is expected again due to the identical motion projection.

It is found out by the study that players got more immersed into game with interaction version 2 which is expected since version 2 provides identical motion projection which engages the players more.

It is found out by the study that right hand tiredness is not changed with two versions which is expected since right hand is used to grab the sword in both versions and rolling motion of the sword does not incur considerable amount of additional physical load. Controversially, left hand tiredness did change with the two versions, version 1 causing left hand of the players more tiredness which is expected since left hand is employed in version 1 whereas is not employed in version 2. Different interaction versions did not have an effect on the overall tiredness of the player which is not expected since version 1 requires more actions to be performed but can be explained with the game's having an easy interaction mechanics, not causing the player tiredness in both versions.

More mental effort results in SMEQ for version 1 is higher than version 2 which is expected since there are more variables and gestures in version 1 whereas there is less to think in version 2 since the virtual sword mimics player's movements with the handheld sword. Users stated a preference for the interaction of version 2 over interaction of version 1 meaning that they are more pleased with the developed motion controlling which provides a more natural and accurate way of interaction.

As the quantitative measures are examined it is observed that players finished the game more quickly while playing with interaction version 2 which may be due to the easy motion projection the interaction provided which also help spatial coordination of the user. Players made more number of fails to finish the game while playing with interaction version 1 which may be caused by the difficult mental motion projection. Time between consecutive successes was less in interaction version 2 which may be explained by the easier interaction's providing the player better performance. It is found out by the results that the players spent more time before passing directions screen in interaction version 1 which is expected since it involves more learned gestures than version 2 which take time for the player to memorize. These measures are different from results of "Space Shape" indicating that different game designs affect the measures. "Rotate Hoopla" has more spatial variables in it, resulting in a more difficult game design as compared to "Space Shape" which reflects to the user study results.

Positive user reviews on the developed interaction version 2 were on realism, naturalness and feeling of being in control of the sword while negative user reviews were on the difficulty of motion projection encountered with version 1, coordination problems and mental effort spent version 1 requires although learned gesture of version 1 is designed to be meaningful and natural as possible.

As user study results are examined for both games, it can be stated that a more natural motion controlled interaction is achieved with the addition of rolling motion recognition. Advantages of the developed interaction can be stated as follows:

- More natural gestures that the user already knows how to perform can be added into the scope of motion controlling with the addition of rolling movement. Turning their hands is a familiar action to players.
- More realistic interaction is provided to players with the increased degree of freedom providing a more natural, near to real-life interaction.
- Variety of gestural design that the developed interaction provides with the increased degree of freedom. Expressive and self-explanatory gestures may be added into motion controlled gameplay such as opening a door by turning a knob and rotating a sword during gameplay.
- Easier mental mapping of the performed actions is provided with the developed interaction since it projects rolling motion directly into gameplay without any transformations. Meaning that no additional gestures should be assigned for rotation which needed to be memorized by the players, incurring mental load. Number of learned gestures is decreased with the developed system.
- Developed motion controlling system provides more kinematic fidelity to players, which increase sense of presence and being in control as the players see that their actions directly affect gameplay and outcomes.

- The developed system provides easier adaptation of the players to motion controlling since it is a more natural and easier way of interaction which may decrease number of players lost during tutorials involving large number of learned gestures.
- More immersive and enjoyable gameplay since the players are provided with real time projections of their actions on the in game avatar, seeing the effects of their performed actions on the outcomes of gameplay, being more in control of what is happening in the game hence embodying the interaction and owning the game more with the feeling of being in charge.

Realism is a very important aspect of modern video games which is aimed by many of the released games to be more realistic. Naturalness and trying to incur the player as few learning requirements as possible are golden rules of motion controlled interaction design for video games established so far. Since proposed algorithms provide a more natural, accurate and instinctive way of interaction with increased degree of freedom, they may help game designers to employ more natural gestures and explore various possible gesture designs that are made possible by rolling axis of motion recognition.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

In this chapter, conclusions of the thesis study on providing an improved gameplay experience by introducing rolling axis of rotation into motion controlled gameplay and future work implications are presented in relevant subsections.

## 5.1. Conclusion

In this study, two algorithms are developed to detect rolling axis of rotation of a bare hand and a handheld object. Developed algorithms introduce an additional degree of freedom to the scope of motion recognition achieved with Kinect, hence providing more accuracy. Besides enabling additional variety of gestures and movements that can be incorporated into gameplay, rolling motion of bare hand can also be used in physiotherapy applications effectively. Two games are developed to assess the contribution of the developed algorithms on gameplay each having two versions one controlled with standard motion controlling provided by Kinect and one controlled with more accurate motion controlling achieved by developed algorithms with the addition of rotation recognition. A user study is performed to measure the effects of the different interaction styles on user experience. Finally, results of the user study are discussed with statistical analysis. User study results revealed that the developed algorithms provided a more natural and realistic motion controlled interaction yielding to easier mental mappings of the actions and more sense of control.

## 5.2. Future Work

Motion interaction is an increasingly shining area of game technologies. In the future, more advanced and accurate motion recognition that is as realistic as real-life interaction will most probably be an indispensable component of video game interaction but for now, motion controlling is its early stages and may benefit from every work scrutinizing the issue. As future improvements, hand rotation recognition may be expanded to understand rolling motion in all three axes not only global z; and rotational limitation of the object rotation can be removed which enable rotational recognition of even cylindrical objects.

# REFERENCES

[1] “Wii Official Site at Nintendo” Internet: http://www.nintendo.com/wii [Mar. 21, 2012].

[2] “PlayStation® Move Motion Controller - PlayStation® 3 Move Info, Games & Updates” Internet: http://us.playstation.com/ps3/playstation-move/ [Mar. 21, 2012].

[3] “Kinect - Xbox.com” Internet: http://www.xbox.com/en-US/kinect [Mar. 21, 2012].

[4] J. Giles. 2010. "Inside the race to hack Microsoft’s Kinect" NewScientist, Tech. http://www.newscientist.com/article/dn19762-inside-the-race-to-hack-the-kinect.html [December 4, 2010].

[5] “Guinness World Records - Home of the Longest, Shortest, Fastest, Tallest facts and feats” Internet: http://www.guinnessworldrecords.com/ [Nov. 13, 2011].

[6] J. Moen. 2005. "Dance-Based KinAesthetic Movement Interaction." School of Computer Science and Communication, KTH. http://movementinteraction.com.

[7] 2011 Microsoft Corporation. "Readme for Kinect for Windows SDK" Internet: http://www.microsoft.com/en-us/kinectforwindows/develop/readme.htm [June 5, 2012].

[8] J. LaViola and F. Keefe. 2011. "3D spatial interaction: applications for art, design, and science." In ACM SIGGRAPH 2011 Courses (SIGGRAPH '11). ACM, New York, NY, USA, , Article 1 , 75 pages. DOI=10.1145/2037636.2037637.

[9] "Human Interface Guidelines, Kinect for Windows v1.5.0" Internet: Official Microsoft Kinect for Windows Webpage. Learning resources and documentation. http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx [April 26, 2012].

[10] "Kinect Developer Downloads | Kinect for Windows" Internet: http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx [Mar. 21, 2012].

[11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. 2011. "Real-time human pose recognition in parts from single depth images." In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11). IEEE Computer Society, Washington, DC, USA, 1297-1304. DOI=10.1109/CVPR.2011.5995316 http://dx.doi.org/10.1109/CVPR.2011.5995316.

[12] M. Marriott. 2006 "At the Heart of the Wii Micron-Size Machines." The New York Times. http://www.nytimes.com/2006/12/21/technology/21howw.html?partner=permalink&exprod=permalink. [June 9, 2012].

[13] T. Shiratori and J. Hodgins. 2008. "Accelerometer-based user interfaces for the control of a physically simulated character." ACM Trans. Graph. 27, 5, Article 123 (December 2008), 9 pages. DOI=10.1145/1409060.1409076 http://doi.acm.org/10.1145/1409060.1409076.

[14] A. Shirai, E. Geslin, and S. Richir. 2007. "WiiMedia: motion analysis methods and applications using a consumer video game controller." In Proceedings of the 2007 ACM SIGGRAPH symposium on Video games (Sandbox '07). ACM, New York, NY, USA, 133-140. DOI=10.1145/1274940.1274966 http://doi.acm.org/10.1145/1274940.1274966.

[15] S. Mitra and T. Acharya. 2007. "Gesture Recognition: A Survey." Trans. Sys. Man Cyber Part C 37, 3 (May 2007), 311-324. DOI=10.1109/TSMCC.2007.893280 http://dx.doi.org/10.1109/TSMCC.2007.893280.

[16] S. Grandhi, G. Joue, and I. Mittelberg. 2011. "Understanding naturalness and intuitiveness in gesture production: insights for touchless gestural interfaces." InProceedings of the 2011 annual conference on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, 821-824. DOI=10.1145/1978942.1979061 http://doi.acm.org/10.1145/1978942.1979061.

[17] D. Saffer. 2008. "Designing Gestural Interfaces: Touchscreens and Interactive Devices." O'Reilly Media, Inc..

[18] R. Wang and J. Popović. 2009. "Real-time hand-tracking with a color glove." InACM SIGGRAPH 2009 papers (SIGGRAPH '09), Hugues Hoppe (Ed.). ACM, New York, NY, USA, , Article 63 , 8 pages. DOI=10.1145/1576246.1531369 http://doi.acm.org/10.1145/1576246.1531369.

[19] J. Wachs, M. Kölsch, H. Stern, and Y. Edan. 2011. "Vision-based hand-gesture applications." Commun. ACM 54, 2 (February 2011), 60-71. DOI=10.1145/1897816.1897838 http://doi.acm.org/10.1145/1897816.1897838.

[20] M. Roccetti, G. Marfia, and A. Semeraro. 2011. "A fast and robust gesture recognition system for exhibit gaming scenarios." In Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques (SIMUTools '11). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 343-350.

[21] J. Lee. 2010. "In search of a natural gesture." XRDS 16, 4 (June 2010), 9-12. DOI=10.1145/1764848.1764853 http://doi.acm.org/10.1145/1764848.1764853.

[22] B Simon. 2010. "Wii are Out of Control: Bodies, Game Screens and the Production of Gestural Excess". http://ssrn.com/abstract=1354043.

[23] B. Lok, S. Naik, M. Whitton, and F. Brooks, Jr.. 2003. "Effects of handling real objects and self-avatar fidelity on cognitive task performance and sense of presence in virtual environments." Presence: Teleoper. Virtual Environ. 12, 6 (December 2003), 615-628. DOI=10.1162/105474603322955914 http://dx.doi.org/10.1162/105474603322955914.

[24] G. Frasca. 2007. "Play the message: play, game and videogame rhetoric." Copenhagen, IT-University, Ph.D Thesis.

[25] B. Buxton. 2010. "Multi-touch systems that I have known and loved." http://www.billbuxton.com/multitouchOverview.html [March 25, 2012].

[26] J. Tang, J. Chan, and H. Leung. 2011. "Interactive dancing game with real-time recognition of continuous dance moves from 3D human motion capture." InProceedings of the 5th International Conference on Ubiquitous Information Management and Communication (ICUIMC '11). ACM, New York, NY, USA, , Article 50 , 9 pages. DOI=10.1145/1968613.1968674 http://doi.acm.org/10.1145/1968613.1968674.

[27] M. Roccetti, G. Marfia, and M. Zanichelli. 2010. "The art and craft of making the Tortellino: playing with a digital gesture recognizer for preparing pasta culinary recipes." Comput. Entertain. 8, 4, Article 28 (December 2010), 20 pages. DOI=10.1145/1921141.1921148 http://doi.acm.org/10.1145/1921141.1921148.

[28] P. Hamalainen, T. Ilmonen, J. Hoysniemi, M. Lindholm, and A. Nykanen. 2005. "Martial arts in artificial reality." InProceedings of the SIGCHI conference on Human factors in computing systems (CHI '05). ACM, New York, NY, USA, 781-790. DOI=10.1145/1054972.1055081 http://doi.acm.org/10.1145/1054972.1055081.

[29] “Harry Potter and the Deathly Hallows™ - Part 1” Internet: http://marketplace.xbox.com/en-US/Product/Harry-Potter-and-the-Deathly-Hallows-Part-1/66acd000-77fe-1000-9115-d802454108f9 [Jul. 29, 2012].

[30] “LucasArts.com | Kinect Star Wars” Internet: http://www.lucasarts.com/games/kinectstarwars/ [Jul. 29, 2012].

[31] “DECA SPORTS FREEDOM - Xbox.com” Internet: http://marketplace.xbox.com/en-US/Product/DECA-SPORTS-FREEDOM/66acd000-77fe-1000-9115-d802485507df [Jul. 29, 2012].

[32] “Kinect Sports - Xbox.com” Internet: http://marketplace.xbox.com/en-US/Product/Kinect-Sports/66acd000-77fe-1000-9115-d8024d5308c9 [Jul. 29, 2012].

[33] “Dance Central 3 - Xbox.com” Internet: http://marketplace.xbox.com/en-US/Product/Dance-Central-3/66acd000-77fe-1000-9115-d802373307d9 [Jul. 29, 2012].

[34] “WarioWare: Smooth Moves” Internet: http://www.nintendo.com/sites/software_warioware.jsp [Jul. 29, 2012].

[35] “Kororinpa” Internet: http://www.kororinpa.com/ [Jul. 29, 2012].

[36] G. Bozgeyikli. 2012. “Introducing Tangible Objects Into Motion Controlled Gameplay Using Microsoft Kinect.” M.Sc. thesis, Middle East Technical University, Ankara, Turkey.

[37] “Unity - Game Engine” Internet: http://unity3d.com/ [Jul. 29, 2012].

[38] “Unity3D” Internet: http://zigfu.com/en/zdk/unity3d/ [Jul. 29, 2012].

[39] M. Brown, A. Kehoe, J. Kirakowski and I. Pitt. 2010. “Beyond the Gamepad: HCI and Game Controller Design and Evaluation” Chapter 12 in Regina Bernhaupt (ed), "Evaluating User Experience in Games". Springer Verlag, London, UK, 197-219. DOI= 10.1007/978-1-84882-963-3_12 http://dx.doi.org/10.1007/978-1-84882-963-3_12.

[40] R. Bernhaupt, W. Ijsselsteijn, F. Mueller, M. Tscheligi and Dennis Wixon. 2008. “Evaluating user experiences in games.” In CHI '08 extended abstracts on Human factors in computing systems (CHI EA '08). ACM, New York, NY, USA, 3905-3908. DOI=10.1145/1358628.1358953 http://doi.acm.org/10.1145/1358628.1358953.

METU
LIBRARY

**TEZ FOTOKOPİ İZİN FORMU**

**ENSTİTÜ**

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☒

Deniz Bilimleri Enstitüsü ☐

**YAZARIN**

Soyadı : BOZGEYİKLİ
Adı : EVREN CAN
Bölümü : MODELLEME VE SİMÜLASYON/OYUN TEKNOLOJİLERİ

**TEZİN ADI** (İngilizce) : INTRODUCING ROLLING AXIS INTO MOTION CONTROLLED GAMEPLAY AS A NEW DEGREE OF FREEDOM USING MICROSOFT KINECT

**TEZİN TÜRÜ** : Yüksek Lisans ☒ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. ☒
2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐
3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası ............................ Tarih ............................