FOUNDATION UNIVERSITY ISLAMABAD

SMART PHONE GAMING USING KINECT IN VIRTUAL REALITY

(SPGKVR)

Submitted by

F161BSCS046 – WAJID MUJIB

F161BSCS005 – FAHMEEN MAZHAR

F161BSCS039 – HARIS SHAHID

F161BSCS019 – HAROON MAZHAR

BACHELORS OF SCIENCE

IN

COMPUTER SCIENCE

YEAR

2016-2020

Project Advisor

Dr. TEHMINA KARAMAT

Project Coordinator

Dr. SHARIQ HUSSAIN

Submitted to the Department of Software Engineering of Foundation University Islamabad, in partial fulfilment for the requirements of a Bachelors of Computer Degree in Software Engineering

DEDICATION

To Almighty Allah, for whose greatness we do not have enough words. To our parents and friends, without whose unflinching support and unstinting cooperation, a work of magnitude would not have been possible.

ACKNOWLEDGMENTS

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report and guided us during the whole semester. A special gratitude we give to our Final Year Project supervisor, **Dr Tehmina Karamat**, whose contribution to stimulating suggestions and encouragement helped us to coordinate our project especially in writing this report.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of Foundation University, who permitted us to use all the required equipment and the necessary materials to complete the project. Last but not least, many thanks to the coordinator, **Dr. Shariq Hussain** who has invested full effort in guiding the team in achieving the goal. We have to appreciate the guidance given by other supervisors as well as panel especially in our project presentation that has improved our presentation skills thanks to their command and advice.

STATEMENT OF ORIGINALITY

We namely **Wajid Mujib, Fahmeen Mazhar, Haris Shahid, and Haroon Mazhar,** **s**ubmitting a thesis titled "**Smart Phone Gaming using Kinect in Virtual Reality (SPGKVR)**". State that we clearly understand what plagiarism is and we have also read about it using online sources. As such, we claim that this entire work (code, reports, etc.) is our effort. We have not used/copy or pasted or paraphrased even a single line of code or sentence from any other source without giving a proper reference to it. Based on our confidence in our originality, we allow the University to run anti-plagiarism software on the same.

We understand that if now, or in the future, ever it is found out that we have not been honest and have plagiarized, the university will take action and will result in disciplinary action, such as the cancellation of our degree and we can be liable for any subsequent punishments as deemed appropriate by the University as well as the Higher Education Commission Pakistan.

____

Wajid Mujib

____

Fahmeen Mazhar

____

Haris Shahid

____

Haroon Mazhar

COPYRIGHT STATEMENT

Copyright in the text of this report rests with the student authors **Wajid Mujib**, **Fahmeen Mazhar**, **Haris Shahid**, and **Haroon Mazhar**. Copies (by any process) either in full or of extracts may be made only in accordance with instructions given by the author. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the author.

The ownership of any intellectual property rights which may be described in this project is vested in FUI, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the FUI, which will prescribe the terms and conditions of any such agreement.

CERTIFICATE OF APPROVAL

It is certified that Project titled **"Smart Phone Gaming using Kinect in Virtual Reality",** presented in **15th July 2020**, has been duly approved by the evaluation committee.

Project Advisor: ____________________

Dr. Tehmina Karamat

Project Coordinator: ____________________

Dr. Shariq Hussain

Manager Graduate Off: ____________________

Dr. Arif Jamal

ABSTRACT

This product has been built based on researches conducted in different research papers and Journals we have merged and optimized these researches to build a more refined and fun product. Our product titled, as "**Smart Phone Gaming using Kinect in Virtual Reality**" is a low-cost Smart Phone Virtual Reality Game which uses Microsoft Kinect sensor for detecting user's gestures and Body motion data to control the in-game characters in an approach which is consisting of natural user interface and this provides an interactive way to provide more immersion. We believe that the new opportunities for revolutionizing Gaming experiences using smartphones will be even greater due to research now being conducted in this field. In the future, the benefits will be that the virtual experience will come with more immersion and will eventually take over the traditional gaming or interaction systems. All these designs will feature a radically different interface.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS AND DEFINITIONS

Term	**Definition**
HMD	**HMD (Head-Mounted Display)** is a display device, worn on the head or as part of a helmet that has a small display optic in front of one or each eye. An HMD has many uses including gaming, aviation, engineering, and medicine. Head-mounted Displays are the primary components of virtual reality headsets.
Kinect	***Kinect*** is a line of motion sensing input devices produced by Microsoft. Initially, the ***Kinect*** was developed as a gaming accessory for Xbox 360 and Xbox One.
NUI	**Natural User Interface**, or NUI, or natural interface is a user interface that is effectively invisible and remains invisible as the **user** continuously learns increasingly complex interactions.
VR	**Virtual Reality** is a simulated experience that can be similar to or completely different from the real world. Applications of virtual reality can include entertainment and educational purposes. Other, distinct types of VR style technology include augmented reality and mixed reality.
Unity 3D	**Unity** 3D is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as a Mac OS X-exclusive game engine. As of 2018, the engine had been extended to support more than 25 platforms.
User	A user is a person who utilizes a computer or network service. Users of computer systems and software products generally lack the technical expertise required to fully understand how they work.
Blender	**Blender** is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, and computer games.
SPGKVR	**Smart Phone Gaming with Kinect in Virtual Reality**
SDK	**Software Development Kit**
UML	**Unified Model Language**
HCI	**Human-Computer Interaction**

C h a p t e r 1

INTRODUCTION

1.1 Introduction to Report

The purpose of this report is to clarify the reason behind the construction of this product and also state all related work done in the field concerning the existing problems that lead to this idea and in accordance specify the purpose of the SPGKVR and address its components and features.

1.2 Existing System

Existing virtual reality application that offers some relevant features include different product like VR-chat, Box-VR, Oculus, and Dream-VR some of them are standalone products which offer a virtual reality experience on their own All of them have their quirks but most of them are generally expensive and also lack immersion to a complete level as they expect the user to control or play with controllers most of them don’t offer a tracking feature at all.

1.3 Literature Review

Through the latest generation of commercial systems based on **Head-Mounted Displays (HMDs)**, such as HTC Vive, Oculus Rift, and Samsung Gear, **VR** has left the research labs and is slowly entering the homes of consumers. Most **VR** systems do not allow the perception of the user’s body inside a 3D scene. As such, the immersion level is drastically lowered. Some devices allow the Users to use their hands to interact with the scene through the device’s controllers. A **NUI** is a special type of interface developed to be used without traditional instruments such as mice, keyboards, and physical controllers. It is called natural as it is based on the movements of the user’s body. A **Kinect** skeleton is made up of 20 body joints (head, hands, feet, hip center, etc.). Among the Kinect skeleton data (20 joints coordinates in 3D space), the KOS game uses those corresponding to the player's right hand to manipulate the gun aiming. So, to build an **NUI** based **VR** application, there is a need for a skeletal system to recognize these joint coordinates or gestures and a simplex transmission server to collect the data and transmit it whereas the smartphone application should have a system to recognize these points.

1.4 Problem Definition

Most **VR** systems do not allow the perception of the user’s body inside a 3D scene. As such, the immersion level is drastically lowered. Some devices allow the user to use their hands to interact with the scene through the device’s controllers. Although this type of solution improves significantly the user’s immersion level inside the 3D Scene, it is still far from reality just think about the controllers floating in mid-air instead of being in the user’s hands. Most standalone **Virtual Reality Head-Mounted Displays** that offer tracking functionalities are very costly.

1.5 User Needs

Table 1.1: User Tasks

User Tasks	Description
UT#01	The user will first set up the device and install the necessary software on their personal computer and check whether the device is on.
UT#02	The user will check through the **SDK** whether the device is tracking or not.

UT#03	The user will add their current imp-address so the device can connect to the application through the local area network.
UT#04	The user will open and test the application
UT#05	The user will select a game they want to play.
UT#06	The user will calibrate their movement in front of the device to play the game.
UT#07	The user will select the exit button to exit the game.

1.6 Organization of Report

Chapter 1: The first chapter is the introductory chapter it contains a general introduction, an existing system, a problem definition of systems and contains the literature review.

Chapter 2: The second chapter gives an introduction about the proposed system and gives an overview and some background of the project and in detail explains the problem description and the objectives of the project and explains the scope and the features the system will perform. This chapter contains all the diagrams related to this project.

Chapter 3: The third chapter is the requirement specification of the SRS *which* explain*s* in detail the functional and non-functional requirements the system will perform and is also written primarily for the developers describes the performance, safety, and security and software quality attributes.

Chapter 4: The fourth chapter is about design in which every detail is defined. This chapter is all about designing the system internally and externally both.

Chapter 5: The fifth chapter is about the testing phase. Testing is done on every module and unit. Test cases are shown and the test report is produced about the overall system.

Chapter 6: The sixth chapter is about the conclusion in which product conclusion is described and future work. This gives the overall conclusion about the product.

Chapter 2

INTRODUCTION TO PROPOSED SYSTEM

2.1 Introduction

Our proposed system will decrease some of the problems in the **Virtual Reality** game. The user can easily play games at low-cost and experience **Virtual Reality**.

The applications are organized in a better way according to the user.

2.2 Project Background or Overview

The expectations of fully immersive **VR** cannot be met by such devices, and a natural user interface using gestures and spoken commands is required to enable users to control and interact with objects in the virtual world. This App will allow anyone to play games at a low-cost. The new Kinect for Windows 2 sensor will include the basic abilities of the new Kinect. A Kinect sensor is an object that represents a Kinect sensor. We can enable color/depth/infrared/skeletal tracking for a sensor etc. Although there have been many improvements in terms of tracking and graphics rendering, the physical representation of the user in VR environments has been a poorly considered problem. As such, the immersion level is drastically lowered.

2.3 Problem Description

Although there have been many improvements in terms of tracking and graphics rendering, the physical representation of the user in VR environments has been a poorly considered problem. Most VR systems do not allow the perception of the user’s body inside a 3D scene. As such, the immersion level is drastically lowered. Although this type of solution improves significantly the user‟s immersion level inside the 3D scene, it is still far from reality.

2.4 Project Objectives

We present an approach based on a natural user interface and virtual reality that allows the user's body to be visualized and tracked in-side a virtual environment. We aim to improve the sensation of virtual reality immersion through low-cost technology such as DESTEK V4 and Microsoft Kinect 2. The system will be developed using the Unity 3D game engine and the C# language. Our approach will be validated through the implementation of an application where the user will be able to interact through hand gestures to play a game.

2.5 Project Scope

Although other types of solution improve significantly the user's immersion level inside the 3D scene, they are still far from reality So our product will enable the user to build a low-cost full body tracking system to play games with body movement in real-time.

2.6 Product or Project Features

Kinect for V2 brings you the latest in human computing technologies, enabling the development of applications that allow people to interact with the system naturally by gesture reorganization and motion detection.

Following are the product or project features enlisted:

Table 2.1: Project Features

Product Functionality #	Product Functionality	Description
PF-01	Gesture/Motion Control	It should be able to provide Immersion to the user by providing control through gestures which are the necessary implication of the natural user interface approach as the user needs to feel fully immersed in the virtual scene which cannot be attained if the user is controlling through controllers.

PF-02	Natural User Interface	It should provide NUI (Natural User Interface) for different components of the system such as providing the user with options to select between different games.
PF-03	Feedback/Response	It should provide the user with Sensory Feedback so the user knows that he/she is interacting with the system.
PF-04	Role-play	It should provide the user with a role so the user feels comfortable in the system by interaction.
PF-05	Real-time Capture	It should capture user gesture and motion data in real-time.
PF-06	Instantaneous Communication	It should instantly transfer user data in real-time to application.

2.7 Block Diagram

The following is a block diagram illustrating the working of the SPGKVR system. It shows different modules of the system and how they are interlinked to each other. When the user initializes the Kinect devices and links it to the computer and the system will automatically track and transmit his data to a server point and the user needs to calibrate his movement following the Kinect device.

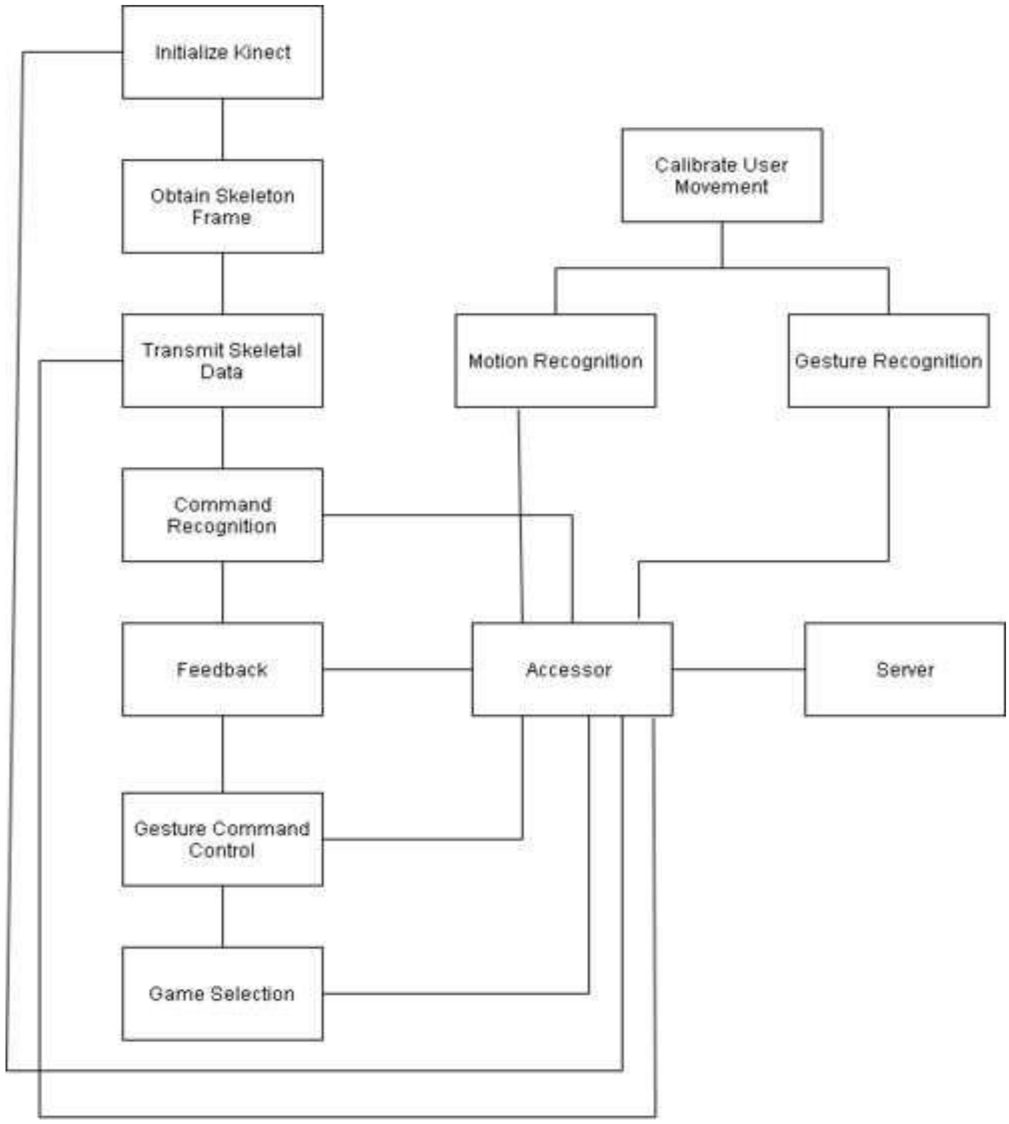

Figure 2.1: Block Diagram

2.8 Uses-Case Diagram

A use case diagram is the simplest way to represent a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

Figure 2.2 is the use case diagram:

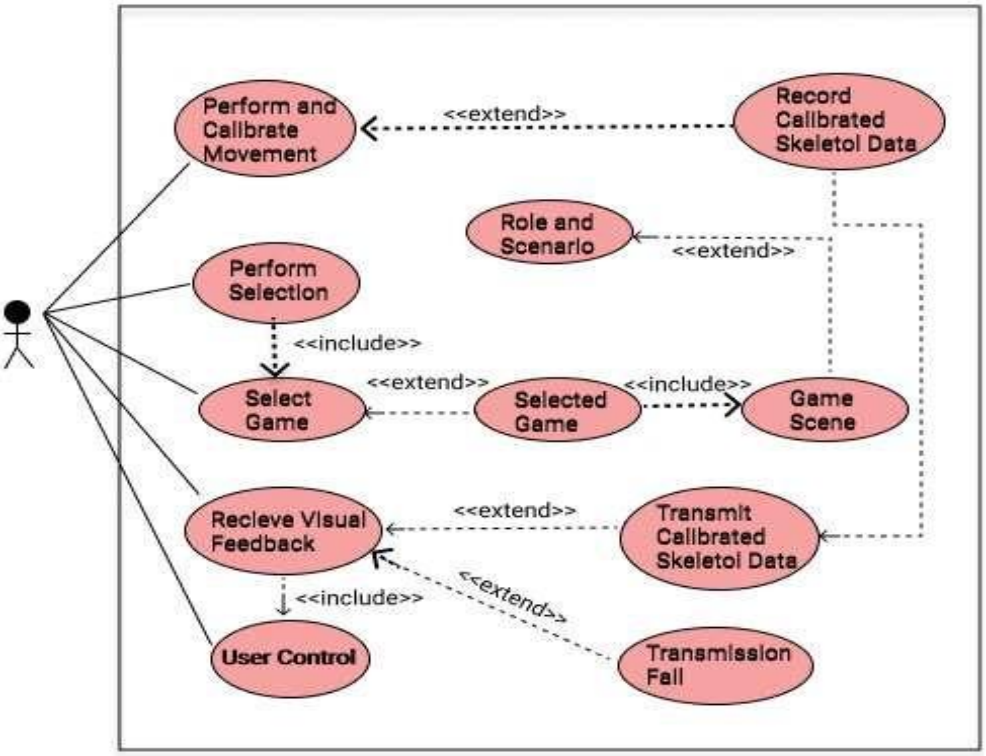

Figure 2.2: Use Case Diagram

The use case table for figure 2.2 and their description and traceability are stated below:

Table 2.2: Use Case Table

UC#	Use Case Name	Description	Traceability
UC1	Calibrate Movement	The user must calibrate their movement according to the Kinect sight to be detected.	FR-1
UC2	Perform selection	The user will be able to perform selection based on his gesture	FR-3
UC3	Select game	The user will select one of the games in the selection menu	FR-2
UC4	User control	The user will be able to control the system	FR-2
UC5	Game Scene	The user will be taken into the gaming scene	FR-4
UC6	Selected game	The game will be selected	FR-2
UC7	Shoot	The user will shoot	FR-1
UC8	Movement	The user will be able to move	FR-1

UC9	Record Calibrated data	The user's movement and gesture data will be recorded in real-time	FR-5
UC10	Transmit skeletal data	The user's data will be sent to the application in real-time	FR-6
UC11	Role and scenario	The user will play within different scene and scenarios and also have a role following the scenario	FR-4
UC12	Visual Feedback	The user will receive proper visual feedback so he knows he’s interacting with the system	FR-3
UC13	Sign-out	The user will be able to sign out at any time from the game scene	FR-2
UC14	Damage	The user will be able to see the damage score	FR-6
UC15	Score	The user will be able to see the score	FR-6

Chapter 3

REQUIREMENT SPECIFICATIONS

3.1 Introduction

It is expected that VR content combined with an HMD like the Oculus Rift will be used in a wide variety of industrial sectors, such as video gaming, film and media, education, sightseeing, the healthcare, and medical field, sports, and advertising. However, the majority of VR games that have been in development over the past few years rely on traditional controllers like a keyboard, a mouse, and a gamepad. The expectations of fully immersive VR cannot be met by such devices, and a natural user interface using gestures and spoken commands is required to enable users to control and interact with objects in the virtual world.

3.2 Functional Requirements

The functional requirements for the product are:

Table 3.1: Functional Requirements

Functional Requirement #	Description	Traceability with Features
FR-01	The character animation in the game must move with user movement and gestures and must trigger proper respective movement inside the game.	PF-1
FR-02	If the player quits the game at any level It must take them to the title screen.	PF-1

FR-03	The start menu screen must load every time the game is launched.	PF-2
FR-04	If the player quits the game at any level it must exit the application.	PF-2
FR-05	The system performs output action against user input action.	PF-3
FR-06	The system must coordinate with user input.	PF-3
FR-07	The system provides the user with a certain role so the user can visualize his task/mission.	PF-4
FR-08	The system generates must show different scenes in all games and every game must have its specified role for the system.	PF-4
FR-09	The hardware system must correctly track and record user data in real-time.	PF-5
FR-10	The data from the tracking hardware must be transmitted to the application.	PF-6

3.3 System Sequence Diagram

A system sequence diagram (SSD) is a sequence diagram that shows a particular scenario of a use case, the events that external actors generate their order, and possible inter-system events.

Figure 3.1 is the system sequence diagram of the project:

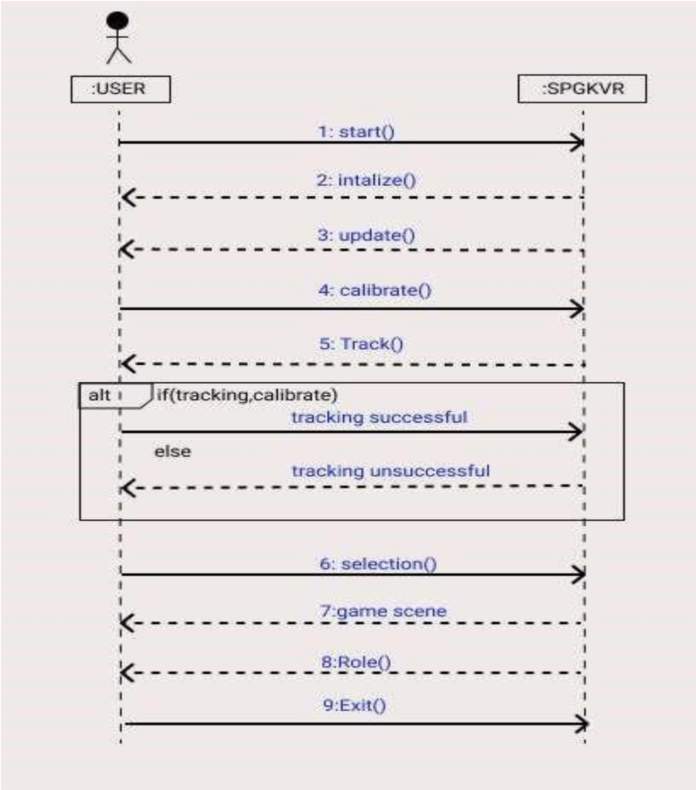

Figure 3.1: System-Sequence Diagram

3.4 Domain Model

A domain model is a conceptual model of the domain that incorporates both behavior and data. In ontology engineering, a domain model is a form of representation of knowledge domain with concepts, roles, data type, and rules and typically grounded in description logic**.**

3.5 User Interface or Graphical User Interface

Since there is only one type of user so the application will only have a simple start menu at the beginning of the application and then after user selection, it will move onto the game display. When the user starts the application, this act will call the function Start (), which will initialize the Scripts and take the user to the selection menu. In the Selection menu the user will be able to choose between a list of different games and after selection function Update (), which will implement game behavior and load a scene related to the selected game and then the user moves to the Game scene until he quits the application by the mean of control button “EXIT” at the bottom of the screen.

- Front-end software: Unity 3D
- Back-end software: Unity 3D C# with visual studio

Following are the details and snapshots of the process:

3.5.1 Guidance Menu

The guidance menu contains the instructions to connect to phone VR. This interface tells the user to connect to their phone by downloading the API and provides a QR image.

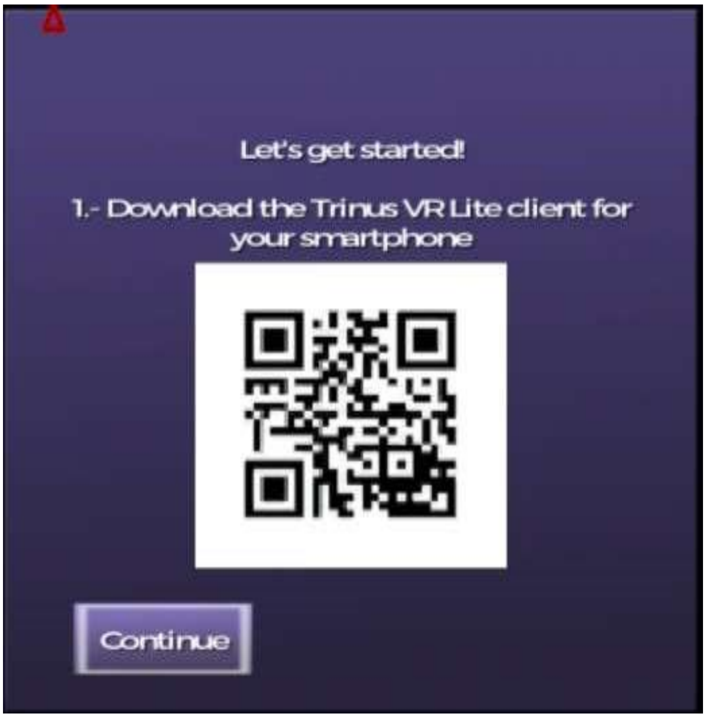

Figure 3.2: Guidance Menu

3.5.2 Connection Briefing Menu

The connection briefing menu gives a set of visual representation of steps that are required to connect by a wireless connection or by a physical connection.

Figure 3.3: Connection Briefing Menu

3.5.3 Instruction Menu

The instruction menu gives a set of important instruction for the main scenes and how they are managed for an optimized experience.

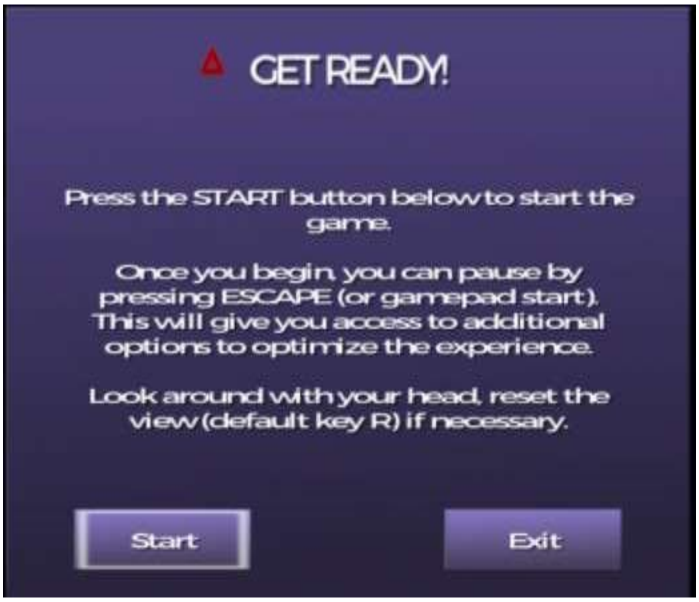

Figure 3.4: Instruction Menu

3.5.4 Connection verification menu

The connection menu gives a set of options so if a connection isn't established It takes the user back to the first menu.

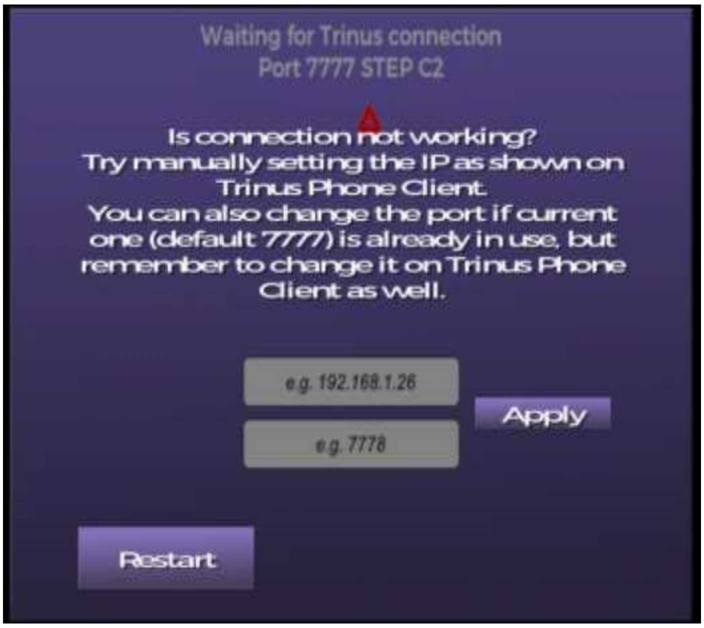

Figure 3.5: Connection verification menu

3.5.7 Main Menu

The selection menu contains the option to select a demo. This interface tells the user to select one demo out of four demos.

Figure 3.6: Main Menu

3.5.8 Game Scene

This is the main scene of the first demo which is a replica of classical game known as beat saber.

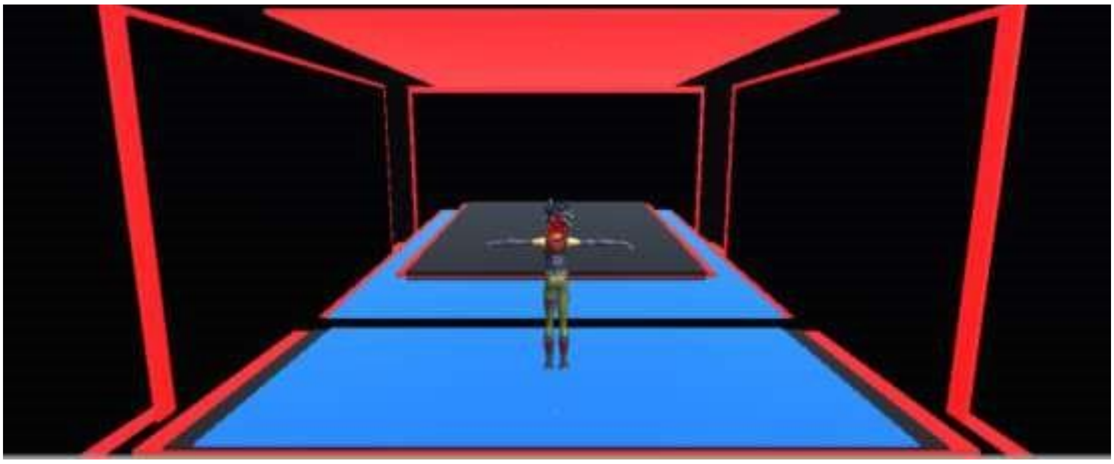

Figure 3.7: Game Scene

3.5.9 Quit Game Menu

This menu contains an option to quit the game. It will open a dialog box for the user whether he wants to quit the game or not.

Figure 3.8: Quit Game Menu

3.6 Data Model (ERD)

An Entity-Relationship Diagram describes interrelated things of interest in a specific domain of knowledge and shows a relationship between entities.

Figure 3.2 is the ER Diagram:

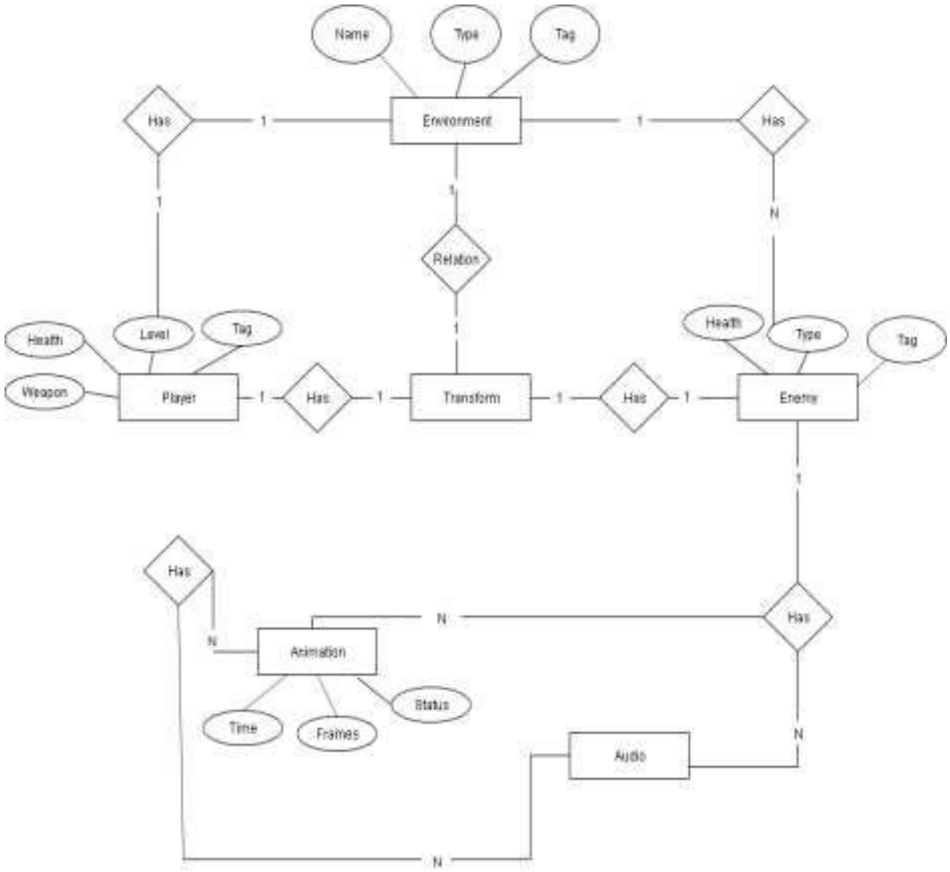

Figure 3.9: ER Diagram

3.7 Activity Diagram

An Activity Diagram is a diagram that depicts the behavior of a system. An Activity Diagram portrays the control flow from a starting point to a finishing point showing the various decision paths that exist while the activity is being executed.

Figure 3.3 is the Activity Diagram:

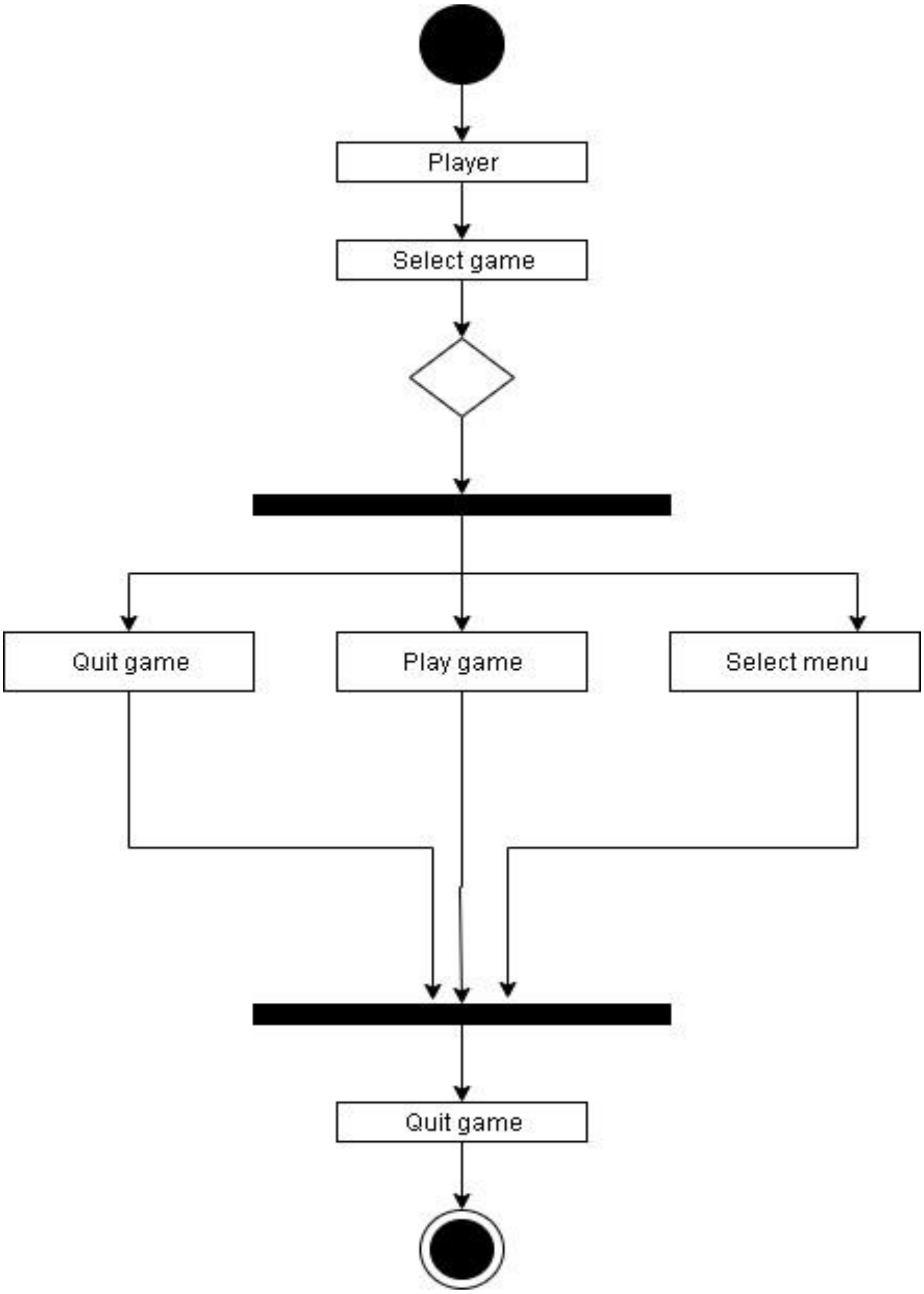

Figure 3.10: Activity Diagram

3.8 State Transition Diagram

It is a Black-Box Testing technique in which the user analyses the behavior of an application under test for different input conditions in a sequence.

Figure 3.4 is the state transition Diagram:

Figure 3.11: State Transition Diagram

3.9 Non-functional Requirements

Following are the systems non-functional requirements:

Table 3.2: Non-functional Requirements

SR. NO	Non-functional Requirements	Description
NFR001	Performance	The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are immediate delays.
NFR002	Portability	The game must be able to run at 30 fps on a smartphone application.
NFR003	Latency	When connecting to the server the delay is based editing on the distance of the system and the configuration between them so there is a high probability that there will be or not a successful connection in less than 20 sec for sake of good communication
NFR004	Precision	The tracking capability of the system should be very precise and accurate to near distances.
NFR005	Safety	Reduce scene lights and rapid screen movements down to a minimum.
NFR006	Correctness	The system needs to able to track motion and gestures correctly. Since the game will need to work based on real-time tracking event and timestamp them with extremely high accuracy, it is important that the timing is accurate and consistent.
NFR007	Usability	If the users were to be confused or annoyed by the controls, then the game would be less enjoyable. There are also special considerations for autistic children to include when developing NUI.

3.10 Benchmarking

Table 3.3 is an overview of our system with some existing system:

Table 3.3: Comparison of different products

Modeling Technique	**VR-Chat**	**Box-VR**	**SPGKVR**
Motion-Tracking	×	×	✓
Gesture-Tracking	×	×	✓
Gesture-Control	×	×	✓
Smartphone-based	×	✓	✓
The first person based	✓	✓	✓
Head-Mounted Display	✓	✓	✓

C h a p t e r 4

DESIGN SPECIFICATIONS

4.1 Introduction

The design needs specific diagrams and system to develop software which follows several rules and laws to work in a public. A Virtual Reality Game will provide a whole emersion experience and allow any player to play a game. The main goal of our system is to provide a low-cost game in Virtual Reality using Kinect V2.

4.1.1 Purpose

Smartphone gaming in Virtual Reality uses the latest Kinect sensor and SDK. You will get more precision, responsiveness, and intuitive capabilities. You need to develop interactive applications that respond to movements, gestures, and hand gestures. This document covers all the features the system intends to provide.

4.1.2 Proposed Solutions Overview

We present an approach based on a Natural User Interface (NUI) and Virtual Reality (VR) that allows the user's body to be visualized and tracked in-side a virtual environment. We aim to improve the sensation of Virtual Reality (VR) immersion through low-cost technology such as DESTEK V4 and Microsoft Kinect V2. The system will be developed using the Unity 3D game engine and the C# language. Our approach will be validated through the implementation of an application where the user will be able to interact through hand gestures to play a game.

4.2 System Architect

The applications of human body movement recognition, detection, and monitoring technology have expanded from the traditional visual gaming realm to non-traditional consumers and enterprises. The research work of vision-based body motion tracking and analysis has focused on tracking and labeling each part of the body that performs some kind of actions [10].

The system architects are a professional figure in information and communications technology. Systems architects define the architecture of a computerized system to fulfill certain requirements.

The human body movement recognition process is implemented on a server component that is connected with the Kinect sensor. The Kinect server maintains a communication process with Kinect sensors through (1) initiating a device instance for each connected Kinect sensor; (2) receiving skeleton frames from the sensor; (3) recording user‟s biometrics; and (4) tracking active user's status [10].

4.2.1 Main Features of Proposed System

Following are the main features of the proposed system:

4.2.1.1 Human Gesture and Body Motion Recognition

The Kinect technology has provided great opportunities for accurately, robustly, and interactively tracking the human body without adhesive markers. In this section, we focus on reviewing recent research in hand gestures, human pose tracking, and body motion recognition and construction in real-time using Kinect [10].

4.2.1.2 Natural User Interface (NUI)

The Natural User Interface (NUI) provides the system to be more interactive and makes the novice user to an expert level.

4.2.1.3 Feedback and Response

While playing the game the system can give feedback and response. Providing interactive feedbacks to users is an important part of many Kinect applications. The recognition process should provide feedback to patients to inform them about any incorrect movement

4.2.1.4 Virtual Reality Gaming and Real-time environment

The recent innovative human body motion tracking and recognition technology allow users to interact with augmented objects freely in real-time for computer-based virtual reality applications and augmented reality games using Kinect [10].

4.2.1.4 Natural Interaction and Control Application

Microsoft Kinect sensor provides a great opportunity to allow the computer to interact with a human by using body gestures, body movement, and verbal commands. There are many natural interaction and control applications emerged recently using the Kinect sensor [10].

4.2.1.5 Hand Gesture Recognition

The research on hand gesture recognition based on Kinect has drawn great attention. We present an integrated system using both head and hands tracking algorithm that aims to achieve robustness and real-time efficiency in the presence of occlusion, disturbance, and the apparent variation in Kinect images [10].

4.2.2 Introduction to Tools and Technologies

Tools and Technologies that we used:

- Kinect for Windows V2
- Unity 3D
- Visual Studio
- Android Studio

4.2.3 Programming Interface for Kinect Server

The programming interface for the Kinect server is a program module that implements communication between the Kinect sensor and server. It receives image data from the Kinect sensor and sends data messages to a device for invoking services or querying status. It provides the interface suitable to the Kinect device so that its image frames can be processed in the server, as well as the interface to a wearable device that can receive or send message descriptions through a smartphone application. The communication calls are based on network data messages description. The primary programming interface for the Kinect server consists of two components: a) communication between the Kinect sensor and server; b) HTTP interface between device and server through a mobile device, which is a smartphone in our system.

4.2.4 General Constraints

Following are the constraints:

- The main design constraint is that the game requires the Kinect device for tracking which happens to be at a low cost but is discontinued from Microsoft and is not manufactured anymore so the user needs to have a Kinect v2 device.
- The application also requires an adapter to connect the Kinect v2.0 device to the windows laptop or PC and a software development kit to enable the device to perform skeletal tracking.
- The application will be written in the C# platform and uses a unity gaming engine. Therefore, it will comply with the programming practices of C#.

4.2.5 Hardware Constraints

Following are the hardware constraints:

- The Kinect V2 sensor requires the following:
- 64-bit (x64) processor
- Physical dual-core 3.1 GHz (2 logical cores per physical system) or faster processor
- USB 3.0 controller dedicated to the Kinect for Windows v2 sensor or the Kinect Adapter for Windows for use with the Kinect for Xbox One sensor
- 4 GB of RAM
- Graphics card that supports DirectX 11
- Windows 8 or 8.1, Windows Embedded 8, or Windows 10
- **Android** device running **Android** 4.4 'KitKat' (API level 19) or higher
- 18 Megabytes of free RAM

- 1 Gigabytes of free storage
- More than or equal to 512 Megabytes RAM

4.2.6 Project Objectives

Main objectives of the proposed system are:

- Create a smartphone-based App that allows the user to play the game.
- To provide an easy interface and low-cost Virtual Reality Application

4.3 Design Methodology

Design methodology refers to the development, implementation, procedures, and tools for designing a unique system or product. It refers to the various kinds of activities that a designer might use within an overall design process.

4.3.1 Software Requirement Specification (SRS)

The software requirement specification of this system is in the documentation part in which we analyze what is this system for, what its purpose, what are the features, how can we develop this system, what are the components of the proposed solution of the system.

4.4 High-Level Design

High-level Design explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

- **Selection Menu**

  The user will select the list of options available on the interface. The user can select whether he wants to start the game or quit the environment.

 Game Scene

The game scene will show the user the gaming environment.

 Quit Game Menu

The user will select the option of „Quit Game‟ after losing or winning the game. This will take the user to the main interface/screen that is „Selection Menu‟.

4.4.1 Functional Requirements

A description of the facility or features required. Functional requirements deal with what the system should do or provide for users. They include descriptions of the required functions.

This system has the following functional requirements:

- The character animation in the game must move with user movement and gestures and must trigger proper respective movement inside the game.
- If the player quits the game at any level it must take them to the title screen.
- The start menu screen must load every time the game is launched.
- If the player quits the game at any level it must exit the application.
- The system performs output action against user input action.
- The system must coordinate with user input.
- The system provides the user with a certain role so the user can visualize his task/mission.
- The system generates must show different scenes in all games and every game must have its specified role for the system.
- The hardware system must correctly track and record user data in real-time.
- The data from the tracking hardware must be transmitted to the application.

4.5 Data design

Data design in Software Engineering designs the first design activity, which results in fewer complexes, modular, and efficient program structure. The information domain model developed during the analysis phase is transformed into data structures needed for implementing the software.

4.5.1 User Interface Design

Following are the design of the User Interface that will be followed throughout the project:

- The User Interface (UI) of our system will be simple and clear.
- Easy to use and read.
- Easy to play the game and use the gaming interface.

4.6 Detail Design or Interaction Diagram

Detailed design is given as a use case.

Use Case Diagram is a kind of behavioral diagram defined by UML. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals represented as a Use Case.

Table 4.1: Use Cases

UC#	Use Case Name	Description	Traceability
UC1	Calibrate Movement	The user must calibrate their movement according to the Kinect sight to be detected.	FR-1
UC2	Perform selection	The user will be able to perform selection based on his gesture	FR-3
UC3	Select game	The user will select one of the games in the selection menu	FR-2
UC4	User control	The user will be able to control the system	FR-2

UC5	Game Scene	The user will be taken into the gaming scene	FR-4
UC6	Selected game	The game will be selected	FR-2
UC7	Shoot	The user will shoot	FR-1
UC8	Movement	The user will be able to move	FR-1
UC9	Record Calibrated data	The user's movement and gesture data will be recorded in real-time	FR-5
UC10	Transmit skeletal data	The user's data will be sent to the application in real-time	FR-6
UC11	Role and scenario	The user will play within different scene and scenarios and also have a role following the scenario	FR-4
UC12	Visual Feedback	The user will receive proper visual feedback so he knows he's interacting with the system	FR-3
UC13	Sign-out	The user will be able to sign out at any time from the game scene	FR-2
UC14	Damage	The user will be able to see the damage score	FR-6
UC15	Score	The user will be able to see the score	FR-6

A Use Case Diagram is the simplest way to represent a user‟s interaction with the system that shows the relationship between user and different use cases.

Figure 4.1 shows the Use-Case Diagram:

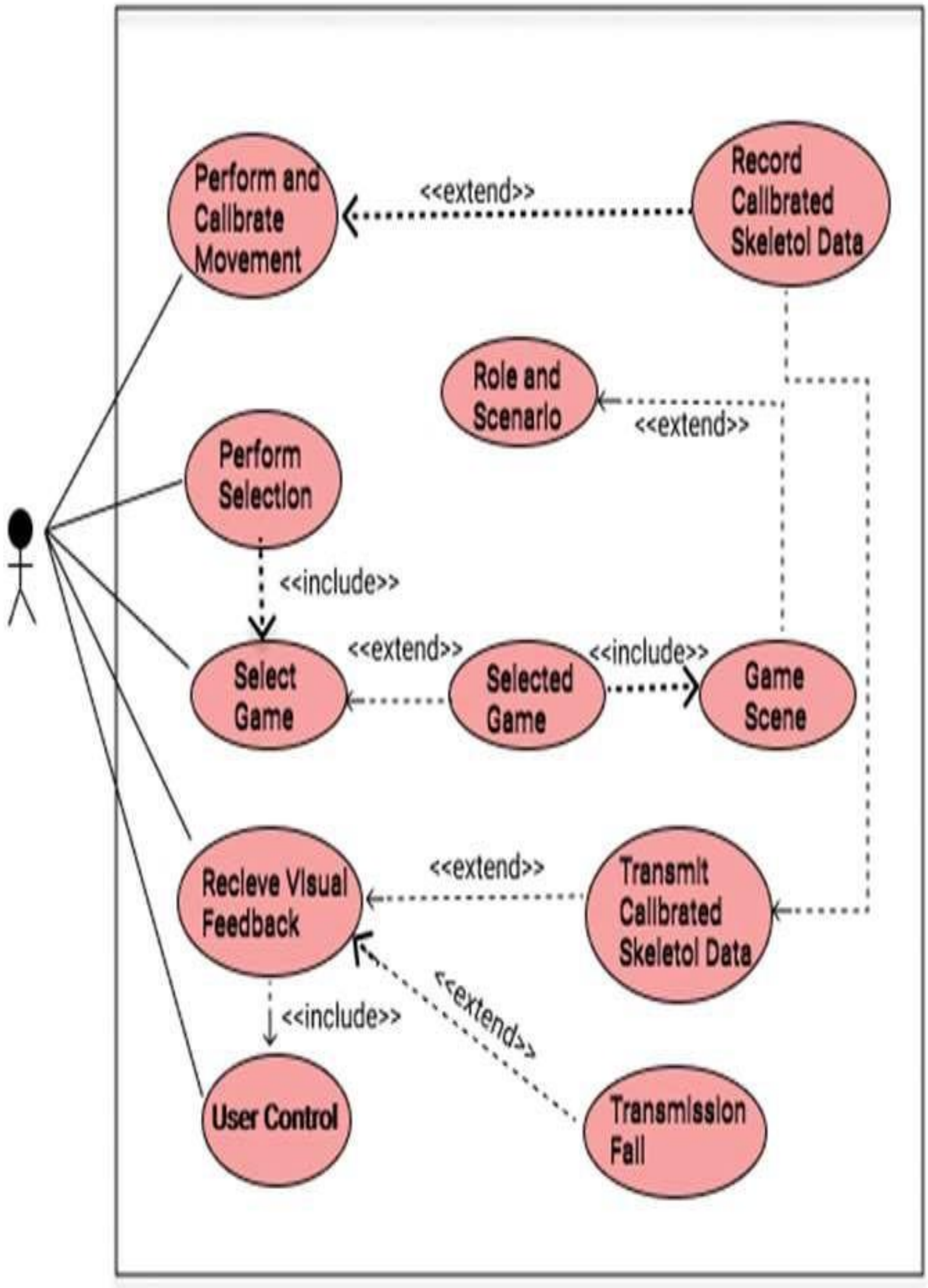

Figure 4.1: Use Case Diagram

4.7 Class Diagram

A class diagram in Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

Figure 4.2 is the Class Diagram:

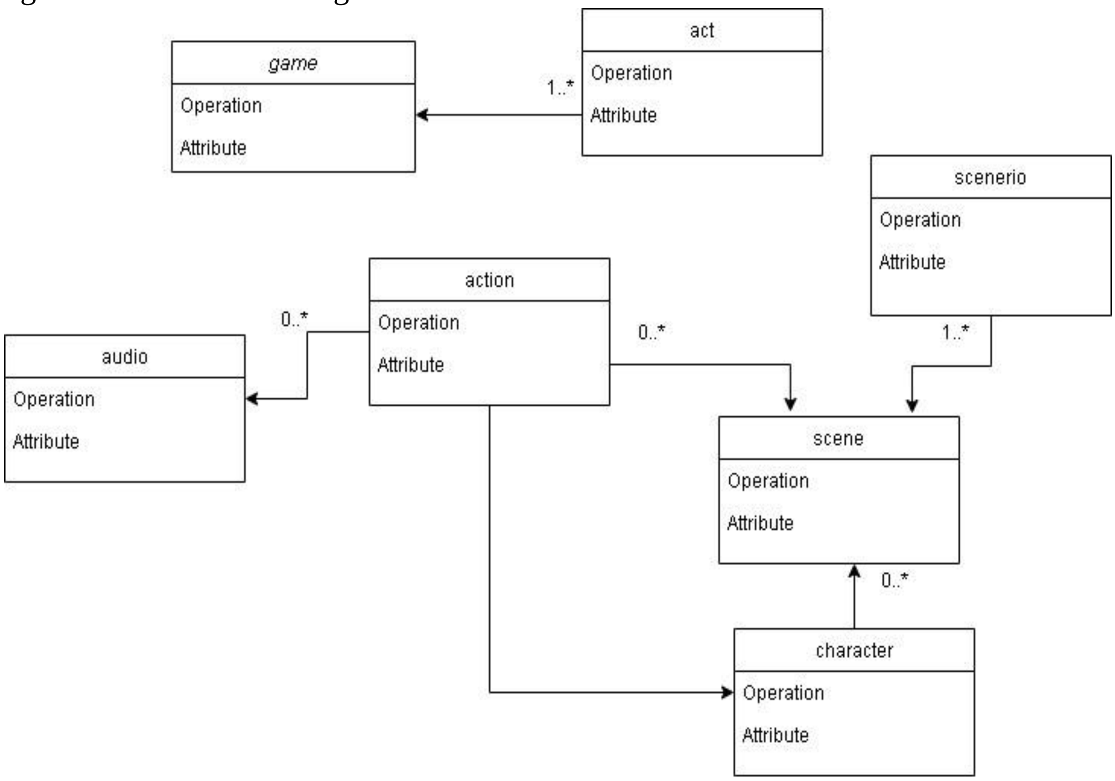

Figure 4.2: Class Diagram

4.8 Deployment Diagram

Deployment diagrams are used for describing how software is deployed into the hardware system. It visualizes how software interacts with the hardware to execute the complete functionality. It is used to describe software to hardware interaction and vice versa.

Figure 4.3 shows the Deployment Diagram:

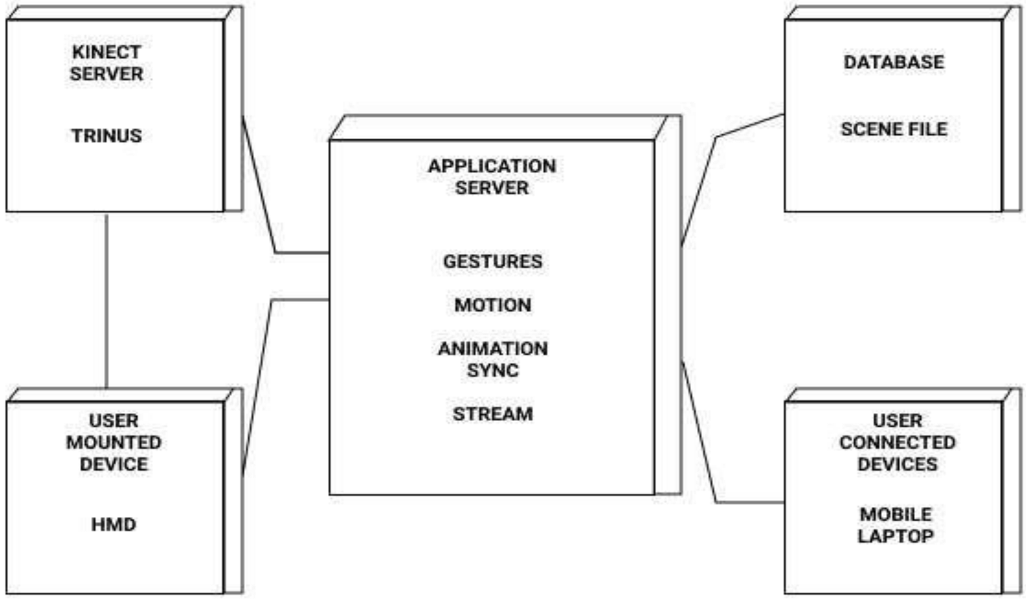

Figure 4.3: Deployment Diagram

C h a p t e r 5

TEST SPECIFICATIONS

5.1 Introduction

This section describes the detailed summary of what scenarios will be tested, how they will be tested, and how often they will be tested. It specifies the specific test, identifies the required inputs and expected results, provides step-by-step procedures for executing the test and outlines the pass/fail criteria for determining acceptance.

5.2 Test Design

Before any project is ready to be delivered to the customer it must be thoroughly tested. Each project requires a test plan document to be designed. Effectively the test plan document is linked to the requirement specification document. *Test Case Specification* has to be done separately for each unit based on the approach specified in the test plan, the feature to be tested for each unit must be determined. The overall approach of testing is to refine into specific test techniques that should be followed and into the criteria to be used for evaluation. Test case specification is performed for system testing by keeping in mind several issues that are discussed in the following subtopics.

5.2.1 White-Box Test Cases

White-Box Testing is a testing of the product's internal structure, design, and coding. Inputs and outputs are tested directly at the code level and the results are compared against specifications.

The White-Box Testing criteria apply for purposes of focusing on internal program structure and discover all its internal program errors. This form oftesting ignores the function of the program under test and will focus only on its code and structure of that code. The code is also visible to the tester.

5.2.1.1 Testing for Connection

Table 5.1: White-box test code 01

Test Case	Steps	D1	D2	Outputs
TC-01	Connect phone / allow access	T	T	Connection Established
TC-02	Connect phone / allow access	F	F	Connection Failed

5.2.1.2 Testing for Stream Sync

Table 5.2: White-box test code 02

Test Case	Steps	D1	Outputs
TC-01	Phone Screen shows VR view	T	Connection Established
TC-02	Phone Screen shows VR view	F	Connection Failed

5.2.1.3 New Scene Event

Table 5.3: White-box test code 03

Test Case	Steps	D1	Outputs
TC-01	Button Event works	T	Scene Event transfers to next level
TC-02	Button Event works	F	Scene event Not working

5.2.1.4 Kinect UI Cursor

Table 5.4: White-box test code 04

Test Case	Steps	D1	Outputs
TC-01	UI cursor Viewing / selection enabled	T	Kinect Manager synchronized
TC-02	UI cursor Viewing / selection enabled	F	Kinect Manager Not synchronized

5.2.1.5 Kinect Avatar sync

Table 5.5: White-box test code 05

Test Case	Steps	D1	Outputs
TC-01	Is the avatar synced with gesture and motion	T	Avatar movement
TC-02	Is the avatar synced with gesture and motion	F	Avatar not moving

5.2.2 Black-Box Test Cases

Black-Box Testing is a testing of product functionality without knowing its internal structure and working. The main purpose of black-box testing is to validate that the application works as the user will be operating it and the environment of their systems. Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.

5.2.2.1 Test Case 01: Trinus Connection Sync

Table 5.6: Black-box test code 01

Test Scenario No	01			
Test Case ID	TCUC01-01			
Purpose	Trinus connection sync			
Cross reference	Scene must stream to phone			
Scenario	**User Action**		**System Response**	
Expected output	1. User will establish Trinus connection by wireless connection or by wire		2. A scene will be displayed on the phone screen	
Actual output	User will be able to interact with menu			
Post-Condition	User will view menu			
Author	User can establish a connection whenever needed			
Execution	WAJID MUJIB			
History	**Date**	**Result**	**Version**	**Run By**
	April 15, 2020	Pass	1.0	WAJID MUJIB

5.2.2.2 Test Case 02: View Scene

Table 5.7: Black-box test code 02

Test Scenario No	02			
Test Case ID	TCUC02-02			
Purpose	View Scene			
Cross reference	Connection must be established			
Scenario	**User Action**		**System Response**	
Expected output	1. User will click on demo 1 button to load scene		2. A demo scene will be displayed on the phone screen	
Actual output	User will be able to interact with demo			
Post-Condition	User will be able to interact whenever connected			
Author	User can establish a connection whenever needed			
Execution	WAJID MUJIB			
History	**Date**	**Result**	**Version**	**Run By**
	April 15, 2020	Pass	2.0	WAJID MUJIB

5.2.2.3 Test Case 03: Avatar Syncing

Table 5.8: Black-box test code 03

Test Scenario No	03			
Test Case ID	TCUC03-03			
Purpose	Establish movement by sensor			
Cross reference	Connection must be established			
Scenario	**User Action**		**System Response**	
Expected output	1. User will try to move and provide motion data		2. A demo scene will be displayed on the phone screen	
Actual output	User will be able to interact with Avatar			
Post-Condition	User will be able to interact whenever connected			
Author	User can establish a connection whenever needed			
Execution	WAJID MUJIB			
History	**Date**	**Result**	**Version**	**Run By**
	April 30, 2020	Pass	3.0	WAJID MUJIB

5.2.2.4 Test Case 04: Natural User Interface / Virtual Reality View

Table 5.9: Black-box test code 04

Test Scenario No	04			
Test Case ID	TCUC04-04			
Purpose	To Give a Virtual reality split screen View to the user			
Cross reference	Connection must be established			
Scenario	**User Action**		**System Response**	
Expected output	1. User will try to calibrate and set his phone in manual Vr glasses		2. A demo scene with virtual view will be displayed on the phone screen	
Actual output	User will be able to interact with scene from virtual glasses			
Post-Condition	User will be able to interact whenever connected			
Author	User can establish a connection whenever needed			
Execution	WAJID MUJIB			
History	**Date**	**Result**	**Version**	**Run By**
	June 20, 2020	Pass	4.0	WAJID MUJIB

5.2.2.5 Test Case 05: Game scenes with sync loading

Table 5.10: Black-box test code 05

Test Scenario No	05			
Test Case ID	TCUC05-05			
Purpose	The game scene appropriate turns load without delay and properly			
Cross reference	Connection must be established			
Scenario	**User Action**		**System Response**	
Expected output	1. User will try to select scene		2. A demo scene will be perfectly called on button event	
Actual output	User will be able to interact with different demos			
Post-Condition	User will be able to interact whenever connected			
Author	User can establish a connection whenever needed			
Execution	WAJID MUJIB			
History	**Date**	**Result**	**Version**	**Run By**
	June 28, 2020	Pass	5.0	WAJID MUJIB

5.2.3 GUI Test Cases

In software engineering, GUI testing, it is the process of testing a product's graphical user interface (GUI) to ensure that it meets its required specifications and designs. It is the testing of products User Interface (UI), its size, and different sections of the screen and whether the font is readable or not.

Table 5.11: GUI Test Cases

Test Case ID	Test Scenario	Expected Result	Pass/Fail	Retest Date	Comments
TCUC01-01	Trinus Connection Sync	Connection Established	Pass	-	-

TCUC02-02	View Scene	Scene can be viewed	Pass	-	-
TCUC03-03	Avatar Syncing	Avatar moving with respect to motion	Pass	-	-
TCUC04-04	Natural User Interface / Virtual Reality View	Scene streaming in Vr View	Pass	-	-
TCUC05-05	Game scenes with sync loading	Loading flawlessly	Pass	-	-

5.2.4 Other NFR Test Cases

This testing is a specification of the other inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

5.2.5 Usability Testing

The following outlines the type of usability testing that will be done for integration and system testing.

5.2.5.1 Integration Testing

Some modules will need to be integrated.

5.2.5.2 System Testing

The goal of system testing is to detect faults that can only be exposed by testing the entire integrated system or some major part of it. System testing is mainly concerned with areas such as performance, validation, security, stress, and configuration. System testing focuses on function, performance, reliability, and how the particular installation is done.

5.2.5.3 Unit Testing

Unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logical errors, or test particular functions or code modules. The unit test cases are designed to test the validity of program correctness.

5.2.6 Software Performance Testing

In software testing, the performance testing is in general, a testing practice performed to determine how the system performs in terms of responsiveness and stability under particular criteria.

5.2.6.1 Performance Testing

The test will be conducted to evaluate the fulfillment of the system with specified performance requirements. It will be done using the black-box testing method and this will be performed by:

- Loading the game
- Detecting the gestures
- Skeleton detection and hand-gestures checking
- Controlling the object

5.2.7 Compatibility Testing

Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments, and mobile devices.

5.2.8 Load Testing

Our system is evaluated under certain conditions in all previous techniques, and no conditions where the system can fail are tested.

5.2.9 Security Testing

The basic purpose of electronic testing is to condemn illegal and unfair means practiced in the test. The security testing is done to checkout authentication and authorized features of the system. Several security checks have been included in the system.

5.2.10 Installation Testing

Most software systems have installation procedures that are needed before they can be used for their main purpose. Testing these procedures to achieve an installed software system that may be used is known as installation testing. These procedures may involve full or partial upgrades and install/uninstall processes.

5.2.11 Acceptance Test Cases

Software acceptability is there a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

5.3 Defect or Bug Sheet

The Defect Sheet is a document that identifies and describes a defect detected by a tester. The purpose of a defect report is to state the problem as clearly as possible so that developers can replicate the defect easily and fix it.

Table 5.12: Defect and Bug Sheet

Syncing and Delay	Delay due to less memory and excessive use of it
Summary	There is a big delay whenever a system has less memory and It results in quite a delay other probable reason is use of ports to sync and connect the sensor
Platform	Windows

5.4 Test Report

It is a document that records data obtained from an evaluation experiment in an organized manner, describes the environmental or operating conditions, and shows the comparison of test results with test objectives.

Table 5.13: Test Report

Executed	**Passed**	**20**
	Failed	**2**
	(Total) Tests Executed (Passed + Failed)	**22**
	Pending	**0**
	In Progress	**0**
Block		**0**
(Sub-Total) Tests Planned		22
(Pending + In Progress + Blocked + Executed) Deferred		**0**

Deferred	**0**

Chapter 6

CONCLUSION

6.1 Introduction

HMD technology has attracted significant interest from the gaming industry. One of the most widely known HMDs is the Oculus Rift, a relatively cheap and easy-to-use VR HMD developed by Oculus VR. It is expected that VR content combined with an HMD like the Oculus Rift will be used in a wide variety of industrial sectors, such as video gaming, film and media, education, sightseeing, the healthcare, and medical field, sports, and advertising. However, the majority of VR games that have been in development over the past few years rely on traditional controllers like a keyboard, a mouse, and a gamepad. The expectations of fully immersive VR cannot be met by such devices, and a natural user interface using gestures and spoken commands is required to enable users to control and interact with objects in the virtual world. This App will allow anyone to play games at a low-cost. The new Kinect for Windows 2 sensor will include the basic abilities of the new Kinect. A Kinect sensor is an object that represents a Kinect sensor. We can enable color/depth/infrared/skeletal tracking for a sensor etc. More and more technologies have been developed in the field of VR. Although there have been many improvements in terms of tracking and graphics rendering, the physical representation of the user in VR environments has been a poorly considered problem. Most VR systems do not allow the perception of the user's body inside a 3D scene. As such, the immersion level is drastically lowered

6.2 Conclusion

We have presented a system for simulating a user‟s body in VR using interactions based on an NUI [1]. The system was developed through the Unity 3D game engine

[1]. The described system has proven to be effective, allowing interactions to a good degree of precision with the virtual environment without the need for additional controllers or sensors [1]. Although this type of solution improves significantly the user's immersion level inside the 3D scene, it is still far from reality so our product will enable the user to build a low-cost full-body tracking system to play games with body movement in real-time. The perception of one’s body is valid, and the movements are faithful, leading to a high sensation of immersion within the virtual environment. However, although the Kinect is a low-cost tracking system, there are hardware limits; for example, the user can't turn around an object, which would result in interruption of the tracking by the device. Furthermore, Kinect hand-tracking ability is very low, in effect limiting the possibility of using single fingers to make choices. Kinect does not allow the user’s position to be tracked in 3D space, which would certainly have simplified its ability to integrate with an HMD. To achieve this, more invasive and expensive tracking systems would be needed [1].

6.3 Future Work

Future work will include the use of multiple Kinect devices (made possible by the client-server architecture of our software), which will be positioned on several sides of the room-scale, to allow the user to perform body rotation of 360 degrees inside the virtual environment instead of always staying frontally to the Kinect [1]. In future work, we can transfer this system into other contexts to evaluate the benefits of tracking a user's body in virtual environments. For instance, we want to test this approach with a 3D information visualization tool used for the comprehension of the object-oriented software system. Finally, we could also widen the set of 3D controllers, to derive insights into which type of user interface could be most beneficial for body tracking inside a virtual environment [1].

BIBLOGRAPHY

[1] *N. Capece, U. Erra, and G. Romaniello, "A Low-Cost Full Body Tracking System in Virtual Reality Based on Microsoft Kinect," in International Conference on Augmented Reality, Virtual Reality and Computer Graphics, 2018: Springer, pp. 623-635.*

[2] *N. Villaroman, D. Rowe, and B. Swan, "Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor," in Proceedings of the 2011 conference on Information technology education, 2011: ACM, pp. 227-232.*

[3] *D. Lee, K. Baek, J. Lee, and H. Lim, "A development of virtual reality game utilizing kinect, oculus rift and smartphone," International Journal of Applied Engineering Research, vol. 11, no. 2, pp. 829-833, 2016.*

[4] *Available:"http://kinectvr.com/ (accessed Friday, 29 November 2019).*

[5] *R. F. Berriel, T. M. Paixao, I. Nicchio, and T. Oliveira-Santos, "Controlling First-Person Character Movement: A Low-Cost Camera-based Tracking Alternative for Virtual Reality," in 2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 2018: IEEE, pp. 67677.*

[6] *B. Kwon, J. Kim, K. Lee, Y. K. Lee, S. Park, and S. Lee, "Implementation of a virtual training simulator based on 360° multi-view human action recognition," IEEE Access, vol. 5, pp. 12496-12511, 2017.*

[7] *O. Patsadu, C. Nukoolkit, and B. Watanapa, "Human gesture recognition using Kinect camera," in 2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE), 2012: IEEE, pp. 28-32.*

[8] *Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," IEEE transactions on multimedia, vol. 15, no. 5, pp. 1110-1120, 2013.*

[9] *M. Tang, "Recognizing hand gestures with microsoft's kinect," Palo Alto: Department of Electrical Engineering of Stanford University:[sn], 2011. [10] https://engagedscholarship.csuohio.edu/etdarchive/1039/*