

# Project Documentation

---

**Project Name:-** FIFA Controller using Kinect

## **Team Members:-**

Manish Sharma,  
Anirudh Kumar,  
Preetish Jaiswal,  
Pratik Likhar

## **Mentors:-**

Vatsal Sharan,  
Shubham Gupta

## **Aim**

Our aim for the project was to create a controller for playing FIFA with our body actions using Microsoft Kinect.

## Motivation

Controlling computers by body movements, speech commands are the next step in human-computer interaction. Microsoft Kinect developed by Microsoft is the first commercially available depth camera with a lot many features for easily developing the human-computer interaction interface. Making a game controller out of it was the most fun thing to do.

## Microsoft Kinect



A view of how a KINECT looks like.

Microsoft Kinect is a motion sensing device my Microsoft for XBOX 360 and windows PC. It allows interacting with the XBOX or PC using Natural through a [natural user interface](#) using gestures and [spoken commands](#).

Kinect builds on [software technology](#) developed internally by [Rare](#), a subsidiary of [Microsoft Game Studios](#) owned by Microsoft, and on [range camera](#) technology by [Israeli](#) developer [PrimeSense](#), which developed a system that can interpret specific gestures, making completely hands-free control of electronic devices possible by using an [infrared](#) projector and camera and a special [microchip](#) to track the movement of objects and individuals in three dimension. This [3D scanner](#) system called *Light Coding* employs a variant of image-based 3D reconstruction.

In terms for development Kinect gives RGB + Depth data for each pixel ,along with this it also gives the location 20 points of the human skeleton for upto four players at a time .

## Software used:

- Visual Studio 2010
- Microsoft Kinect SDK(Software Development Kit) v1.5
- FIFA 11

Microsoft released Kinect [software development kit](#) for Windows 7 on June 16, 2011.<sup>[22][23][24]</sup> This SDK allows developers to write Kinecting apps in [C++/CLI](#), [C#](#), or [Visual Basic .NET](#).

" We decided to use to C# for programming. "

## Hardware

1. A standalone Kinect Sensor.
2. A Windows PC

## Our methodology

Our project involved mainly two steps: -

1. Taking input using the Kinect and then determining the gesture executed or command has to be given.
2. Based on the gesture decided above we have to send the appropriate KeyDown command to the Game.

We did the sending keys part of the project in first. Now the game Fifa uses DirectX which is for generating better 3d graphics and acceleration. It also uses [DirectInput](#) which takes the input from directly the keyboard buffer, thus bypassing the OS. C# has a function called `SendInput()` for adding keys to the keyboard buffer.

We used a wrapper of the SendInput() from the site codeplex.  
Download it from the given link.(Download the src version)  
<http://inputsimulator.codeplex.com/>

### **To use the above:**

1. Create a WPF application.
2. Add a the namespace "using windowInput"
3. For the keys you want to use in your program substitute those keys DirectXScanCodes in the input simulator file by opening it in the C#.And compile it .
4. Add a reference to the .dll file of the above download using solution explorer.
5. We used two function given SimulateKeyDown() and SimulateKeyUp() with some delay between the two using System.threading .thread.sleep(t) , where t is in 'ms'.

For deciding which command to be executed we divided the code into two parts which were:-

1. Deciding the movement
2. Deciding the kick type

For movement we used the tilt of the upper torso of the body to determine the direction of motion of the player. We made a vector from the head to the hip and then we calculated the angle of the vector in the ground plane from the line formed by joining the player and the Kinect. Based on which range the angle belonged to we decided the direction of player motion.

We added another feature that was if the user wants the player to sprint in the game he can do that by tilting more.

## **Implementing Kicks**

After looking out various gesture detection algorithms on net, we found Hidden Markov Models (HMM) as the widely used one in Kinect Apps. But due to much complications with HMM, we finally adopted a self-designed crude algorithm for implementing kicks through gesture forms.

Under the aforementioned algorithm, we create a virtual area around the player's legs. As the Kinect app is executed, a single player is selected (if tracked in frame). Elsewise, multiple skeletons if encountered are discarded. The player is then prompted to advance his/her leg in forward and right directions to calibrate the parameters for the virtual area. Next, the same is done for head movements. Once initialised, the game is launched.

During the gameplay, firstly the skeleton is checked for any kick action. For this, distance between the two legs of the player is determined relative to the virtual area. If outside the area, the gesture form is then scrutinized for the kick type i.e. short pass, lob pass, shoot, through ball. Now we have the following definitions for the above types :

- Short Pass has both legs nearly grounded
- Shoot is generally improvised by lifting the right leg backwards to some height and then kicking the ball
- Lob Pass by lifting the right leg in the appropriate direction save backwards.
- Through Ball is triggered by an uplifted and directed movement of the hand.

Having determined the gesture form for the kick type, we go ahead and simulate the appropriate key for that action in game.

If the gesture form does not meet the similarities with any of the above types, we go on to check for the movement controls.

## Future Scope

Our current algorithm are very basic and self-developed and thus have scope of lot of improvement. In terms of feature we were could have implemented the following features :

1. A Header gesture
2. Some gesture implementation for special kicks in the FIFA game.
3. Sliding tackle

## Useful Links

- **Our code :-**  
<https://docs.google.com/document/d/1Belk5UfLt5Tuinu-apUOvszCXXLvFfl-s1ixkrK-kJY/edit>
- **Kinect for Windows Homepage :-**  
<http://www.microsoft.com/en-us/kinectforwindows/develop/>
- **Coding explanation MSDN :-**  
<http://msdn.microsoft.com/en-us/library/hh855347.aspx>
- **Our Project Wiki :-**  
[http://students.iitk.ac.in/projects/eclub\\_fifacontroller](http://students.iitk.ac.in/projects/eclub_fifacontroller)



## **Word of Thanks**

Our team would specially like to thank our team mentors Vatsal Sharan and Shubham Gupta, Club Coordinators Anurag Dwivedi, RP Suman and Nikhil Gupta for the dedicated help and guidance without which our project would not have been what it is today.