Minor Project-II Final Report on

# Real Time 3D Model Animation using Pose Estimation

Submitted in Partial Fulfillment of the Requirements for the Degree of

**Bachelor In Software Engineering** under Pokhara University


Submitted by:

**Binayak Dotel, 171613**
**Siddhanta Poudel, 171732**
**Nishant Joshi, 171622**
**Krishna Kumar Shrestha, 171617**

**Under the supervision of**
Assoc. Prof. Saroj Shakya


Date:
**02-10- 2021**



**Department of Software Engineering**
**NEPAL COLLEGE OF**
**INFORMATION TECHNOLOGY**
Balkumari, Lalitpur, Nepal

# Acknowledgement

This project has been in this working state due to combined efforts of many individuals who have contributed for the project development. It has been a pleasure for us to acknowledge the support and contributions that were very crucial and helpful throughout the project. We would like to extend our sincere thanks to all of them who were directly and indirectly involved with this project.

We owe special thanks to a number of people who has been devoted much of their time and expertise without which it would haven't been easy for us to complete our project entitled "**Real Time 3D Model Animation using Pose Estimation**".

We would like to take the opportunity to express our humble gratitude to our supervisor Mr. Saroj Shakya ,Assoc. Professor, NCIT, under whom we did our project. His guidance and readiness to share his knowledge helped us to enhance our knowledge and made easier for us to complete the various tasks with much perfection. Without his effort and support this project may not have been fruitful.

We would like to provide our special thanks to Dr. Roshan Chitrakar, for providing technical support with enormous suggestions, which helped us to coordinate our project. Furthermore, we would like to express our gratitude to our teachers, classmates, and families who have always been there as our strong source of inspiration and lent support and views during the entire development time of the project.

Binayak Dotel
Siddhanta Poudel
Krishna Kumar Shrestha
Nishant Joshi

# Abstract

In this project, A motion capture system where the computer should be able to estimate the human pose using a simple camera has been developed. The key points of the human postures are estimated and fed to the rigged human model which generates animation.
Pose Estimation is the computer vision technique that detects the human posture in images or videos, so that the body parts of a human are determined, for example, where the elbow, knee or shoulder is.

3D animation of humans in action is quite challenging as it involves using a huge setup with several motion trackers all over the person's body to track the movements of every limb. This is time-consuming and may cause the person discomfort in wearing high end bodysuits with motion sensors. In this project, A trivial yet effective solution to generate 3D animation from a camera is developed.

We will be using Unity for generating animation and rendering human models, Pose Estimation technology for determining human posture and client server system for real time data sharing from Python to Unity.
The key points for animating the rigged 3D model will be transmitted to unity JSON format.
We developed this project to help people animate their own character on their own without using high end technology like Motion Capture suits, etc.

*Keywords*: Unity, Computer Vision (OpenCV) , Pose Estimation, Motion Capture, JSON and Python.

# Contents

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

## 1.1 Problem Statement

Animating a person in 3D virtual character, like: Human, Animals etc, requires a huge set up with motion trackers to track the person's movements and also takes time to animate each human key joint manually. This discourages the beginners who have just started 3D modelling or Game Development, since such high-end motion capture devices are not accessible by independent developers.

Also, Facebook, Apple and Google are continuously developing devices capable of performing real time virtual meetings using AR/VR technology where the virtual character (or Clone) of the person can move as per the real human movement. For this, motion tracking plays an important role. Though our project does provide the complete technology for virtual meetings, it can contribute a small amount of technological solutions.

This is time-consuming and expensive since it requires high end body suits with motion sensors. In this project, We are aiming to provide a time-saving method for real-time pose estimation using a normal camera. We present a trivial yet effective solution to generate 3D animation of a person through a simple camera.

## 1.2 Project Objectives

Our team being concerned about the problems are setting up some goals to try and contribute few solutions:

1. Help beginner 3D modelers and game developers to create their own animation for their indie games using motion capture technology.
2. Make learning techniques more efficient, interactive and interesting for students.
3. Providing users with a motion capture technology that helps them animate a rigged model as per the real time movement.

## 1.3 Significance of Study

The results of the study will be of great benefit to the following:

1) **Students**: Students learning game development and 3D modelling can get benefits from the system. They can develop their own 3D animation using pose estimation for the indie projects.

2) **Storytelling**: The best part of this study is that it can be helpful to change the traditional aspect of seeing things by introducing a virtual platform with some stories or news. The user gets keen to become a part of events and promotions of the virtual products.

3) **Content creation:** Cartoon and VFX creators can get facilitated for creating animated characters for movies. They can get rid of the high-end devices and sensors to create animations for their characters.

4) **Activity Recognition:** This technique can also be used in fitness apps where the activity of the human is determined and also provides the posture correcting measures.

5) **Motion Capture and Augmented Reality:** This technology can also be beneficial for creating group conference systems where the users can collaborate and communicate with one another by simulating their virtual avatar in the real-world environment of another user.

6) **Training Robot:** Instead of manually programming robots to follow trajectories, robots can be made to follow the trajectories of a human pose skeleton that is performing an action.

   A human instructor can effectively teach the robot certain actions by just demonstrating the same. The robot can then calculate how to move its articulators to perform the same action.

## 1.4 Scope and Limitation

**Scope**:

1) This technology can enable people to create animations automatically without use of complex animation software and expensive devices with high end sensors.
2) This technology can help indie game developers to animate their characters without explicitly animating the character.
3) This technology can create an interactive environment for virtual characters for AR/VR technology.
4) Reduces the use of complex devices and sensors that enables anyone to get facilitated by the service.

**Limitation**:

1) Since this project relies on computer vision and machine learning, its accuracy and reliability depends upon the model trained and datasets used.
2) Though it can animate the 3D character, still the user needs to have their 3D model designed and rigged such that it is ready to adapt with our system.
3) The user must have knowledge about 3D model rigging and designing.

# 2. LITERATURE REVIEW/STUDY

Nowadays, many fields either movies or games, are very much dependent on computer graphics and creating virtual characters. Either it is a large-scale open world game Like: GTA V, Far Cry and Watchdog etc., or small-scale mobile games Like: Subway surfer, Temple Run, etc. have a lot of 2D or 3D animations. This animation development is done by using expensive devices with high end sensors or using 3D animation developing software where manual key points movement is done for moving any character.

Animating requires a huge set up with motion trackers to track the real object movements and also takes time to animate each human key joint manually. This discourages the beginners who have just started 3D modelling or Game Development, since such high-end motion capture devices are not accessible by independent developers. There are few software and hardware devices which are developed for estimating pose and performing interaction:

1) **Microsoft Kinect:** Kinect is Microsoft's motion sensor device that provides a natural user interface (NUI) and is able to estimate human posture inputs that allows users to interact with the system without any intermediary device, such as a controller. The Kinect system identifies individual players through face recognition and voice recognition. A depth camera, which "sees" in 3-D, creates a skeleton image of a player and a motion sensor detects their movements. This device is inbuilt with high end sensors which makes it expensive and unaffordable. [7]

2) **RADiCAL:** RADiCAL is a web-based software developed with a computer vision technology focused on detecting and reconstructing 3D human motion from 2D content. It means that here the software tracks motion from the 2D content (i.e., Video) and generates a .fbx file along with captured animation.
   But this technology is not capable of tracking and creating animation in real time. Here, first a captured video is uploaded to the system, then it performs the processing on the cloud server and generates an fbx file which can be further downloaded and used in our games, AR/VR, etc. [8]

3) **DeepMotion:** DeepMotion is also a web-based software developed with a computer vision technology focused on detecting and reconstructing 3D human motion from 2D

content.

Here, the system is capable of real time motion tracking and generating 3D animation to a rigged model. But, in terms of pricing it is more expensive for any beginner to work with for their indie projects. [9]

These are a few sources we reviewed and studied before starting our project. We aim to create a software system that fulfills the objective of the user as well as is better than these already existing software.

# 3. METHODOLOGY

We used an Iterative and Incremental model for developing this project. Iterative and incremental software development is a method of software development that is modeled around a gradual increase in feature additions and a cyclical release and upgrade pattern. The outcome of the subsequent iteration is an enhanced working increment of the product. This is repeated until the product accomplishes the required functionalities. [10]

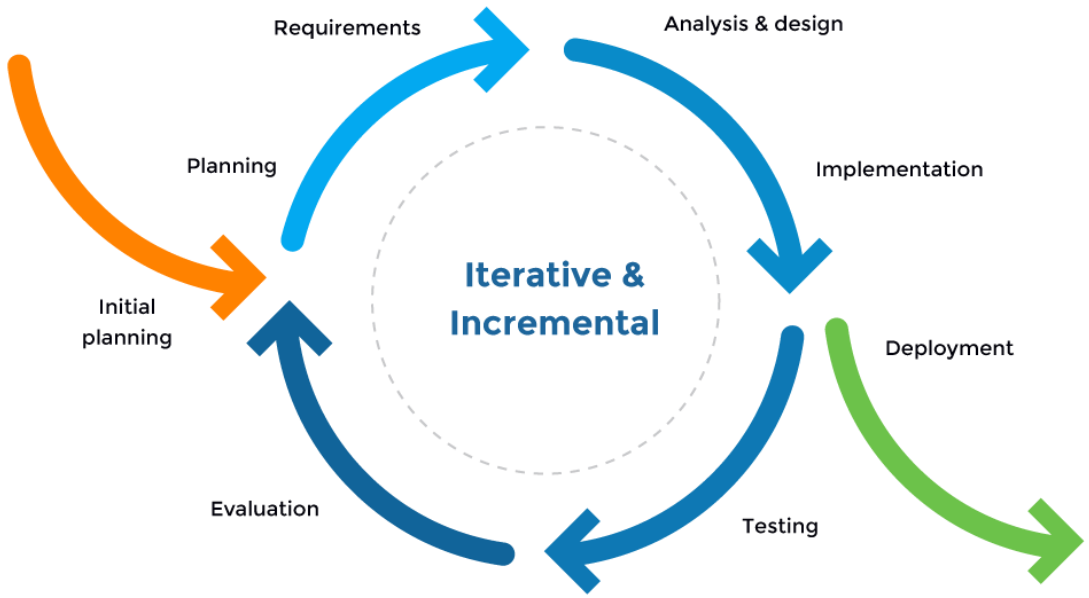


**Figure 1: Iterative and Incremental Model** [10]

Reasons for choosing Iterative and Incremental model:

A. It allows us to develop prioritized requirements first.
B. Initial products with basic working functionalities can be delivered faster.
C. Each release is a product increment, so that we get a working product at hand all the time.
D. Each time we can test and gather feedback from each product increment, such that we can improve the product and also add new features, fix errors and risks.
E. Thus, Requirements can be modified and changes can be easily accommodated.

## 3.1 Time Schedule Estimation

**Planning** :
First of all we figured out the methodology that we were going to follow in order to achieve our goal. Here, we planned everything that would be usable while developing our project. We discussed which language and frameworks that we were going to be using in our project. For the planning purpose it took us around a week to make a proper layout of planning. We also figured out whether we can accomplish the project or not.

**Analysis and Design**:
Here, we analyse and gather the requirements that were needed for the development of the project. We started the design with a basic understanding and grew the architecture throughout the process with each increment.

**Implementation:**
After the planning, analysing and designing ,the next job to be done was the implementation of the system as per the initial requirements and designs. Here,we implemented everything that we planned and designed to fulfill our project requirements and needs. Initially, we implemented the functional requirements and later implemented non-functional requirements with each increment.

**Testing:**
We tested every deliverable increment of the project to figure out whether it was up to the mark or not. By testing we also got knowledge about what to be added and what to be removed.

**Documentation:**
This is the most important task that has to be done for the proper planning of any project. Here, we documented everything we planned, analysed, designed, implemented and tested from day 1 of the project. We documented everything step by step as we completed the different phases of the project.

| Task | Start Date | End Date | Duration |
|---|---|---|---|
| Planning | 03/20/2021 | 03/25/2021 | 7 |
| Analysis & Design | 03/22/2021 | 04/02/2021 | 10 |
| Implementation | 04/03/2021 | 04/232021 | 20 |
| Testing | 04/06/2021 | 04/21/2021 | 4 |
| Documentation | 03/24/2021 | 04/20/2021 | 25 |

**Table 1: Time allocation for the project**

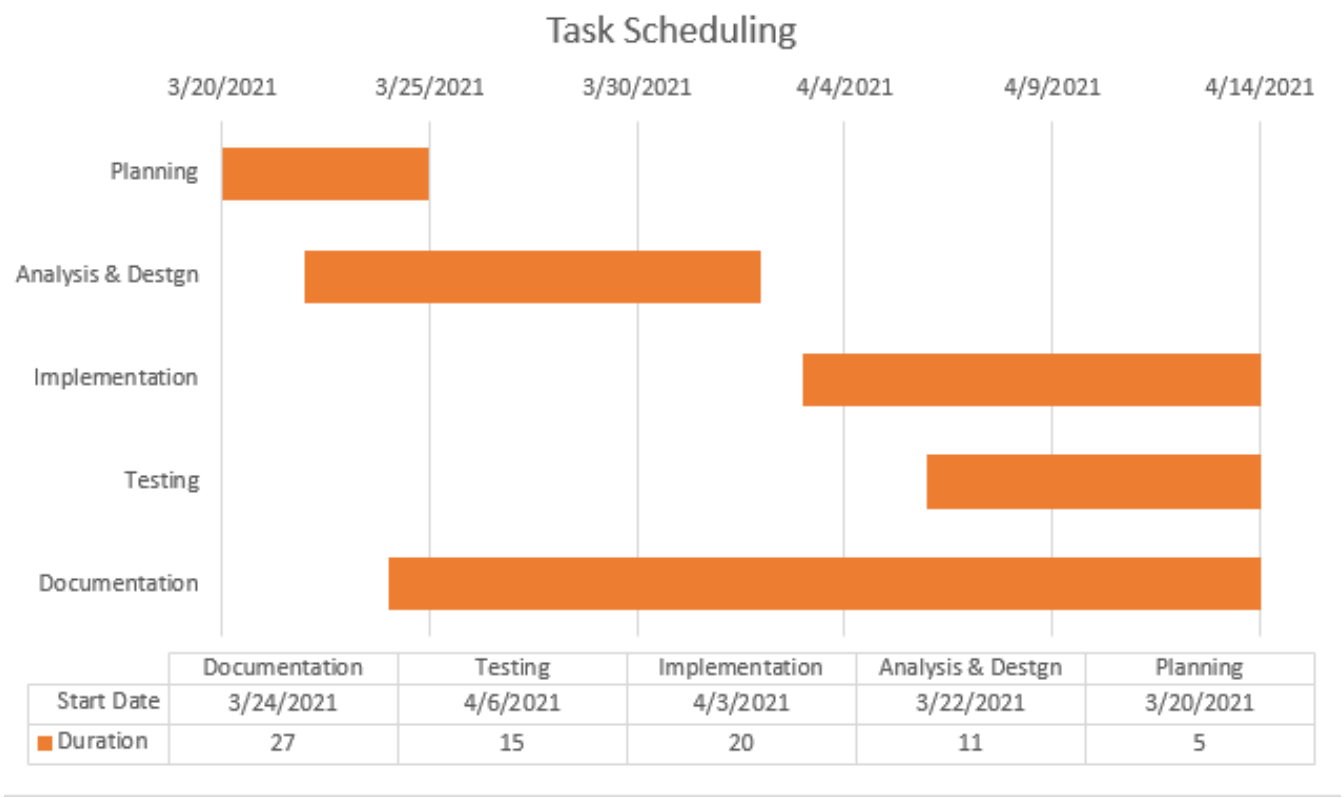


**Figure 2: Time Estimation for the project**

## 3.2 Tools and Technologies used in the development of Project

| Tools Use |
| --- |
| 1. **Unity** -For rendering of 3D models and animating. |
| 2. **OpenCV & Mediapipe** -For estimating human body postures.. |
| 3. **Python & C#** -For programming and scripting functionality. |
| 4. **Socket Programming** -For data sharing in real time |

**Table 2:Tools and technologies used**

Since, in this project we are using computer vision for pose estimation, a rendering engine that is capable of rendering the 3D model which is to be animated. We require various tools and technologies:

1) **Unity:** Unity is a graphics rendering engine that is capable of creating 2D/3D Game, Architectures, Films, Animations, etc. In this project, we used unity to render the 3D model that is to be rendered and animated. [1]

2) **OpenCV:** OpenCV is a computer vision library that helps a computer to gain high-level understanding from digital images or videos. In this project, computer vision is used to get information from the video stream with the help of *mediapipe,* which is a machine learning solution that performs pose estimation. [3]

**3) Python & C#**: Python is used on the *computer vision* side, where we perform pose estimation. C# is used as a scripting language in *unity* to perform animations.

4) **Socket Programming**: For sharing data between pose estimating program and unity script, we tried multiple solutions.
FireBase real time database was our first attempt, but due to internet latency and decrease in frame rate, we have to drop this idea and search for a new one.
Finally, we decided to use socket programming that lets data sharing without use of the internet locally in the same machine.

## 3.2 Team Work and Task Scheduling

| Team Member | Task Done |
|---|---|
| 1. Binayak Dotel | ➔ Unity Scripting, Animation creation, 3D model movement and Task allocation. |
| 2. Siddhanta Poudel | ➔ Socket programming and JSON data handling. |
| 3. Nishant Joshi | ➔ Pose Estimation and keypoint tracking. |
| 4. Krishna Kumar Shrestha | ➔ Documentation, Testing and performance analysis. |

**Table 3:Team Work and Task Scheduling**

# 4. SYSTEM DESIGN & UML DIAGRAM

## 4.1 Use Case Diagram

Use -case diagram is the dynamic diagram which shows the dynamic behavior, there should be some internal or external for making the interaction. It consists of actors, use cases and their relationships. Use case diagrams are modelled to present the outside view. It also shows the interaction among the requirements are actors. In use case diagrams we use different notations like include, extend, generalization where include is used in required condition and extends is used in optional condition and generalization is used in the condition of sub type.(UML - Use Case Diagrams)

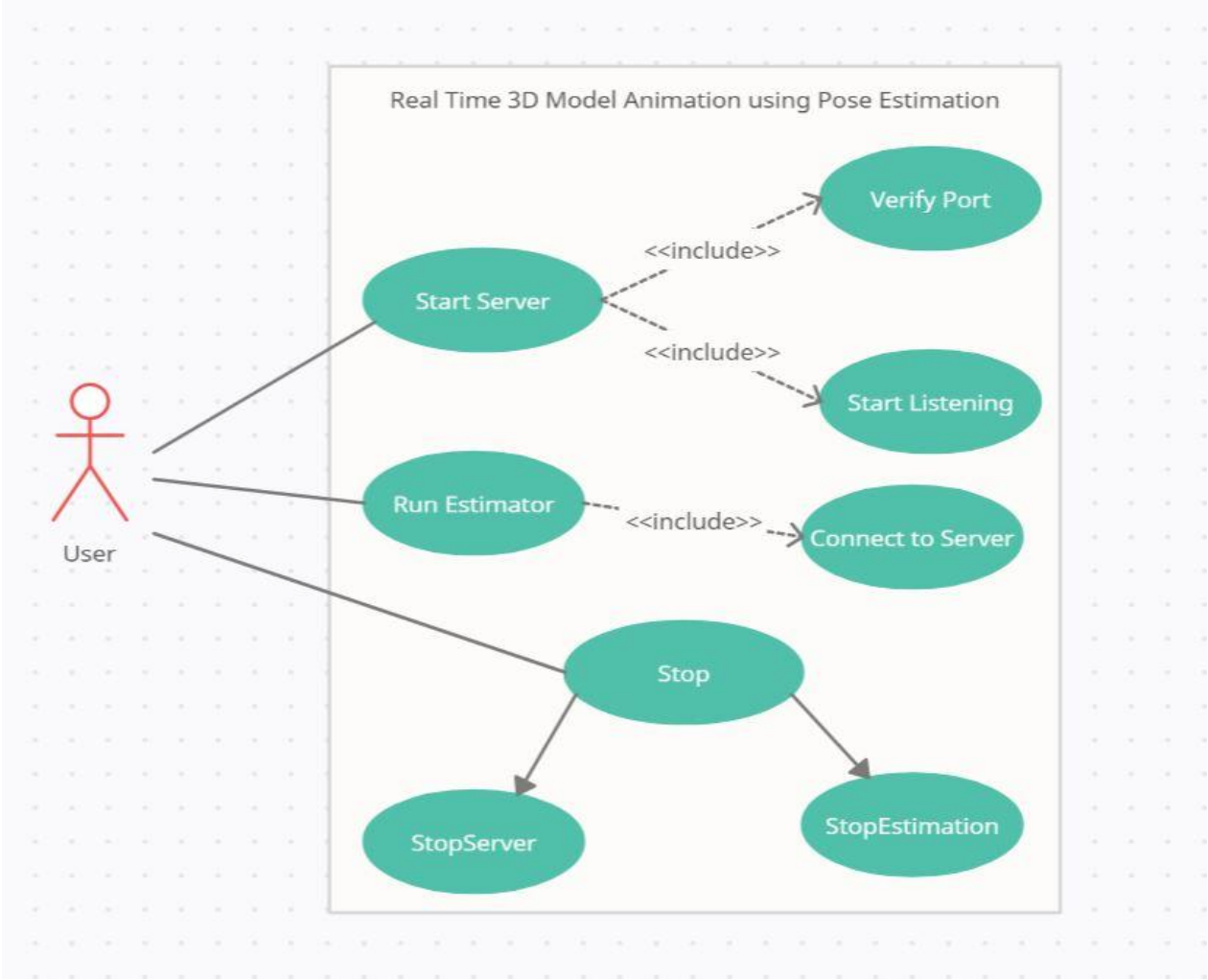


**Figure 3:Use case diagram**

## 4.2 Class Diagram

Class diagram is the static diagram which represents the static view of an application. This diagram is used for visualizing and describing the aspects of a system. Class diagram is also used for constructing the code of the system. In class diagram there is attribute and method in a class. Class diagram is also known as a structural diagram. It analysis and design the static view of the system. Class diagram use different notation like aggregation implies a relationship where the child can exist independently of the parent, composition implies a relationship where the child cannot exist independent of the parent, generalization is used for combining similar classes of objects into a single, more general class.(UML - Class Diagram)

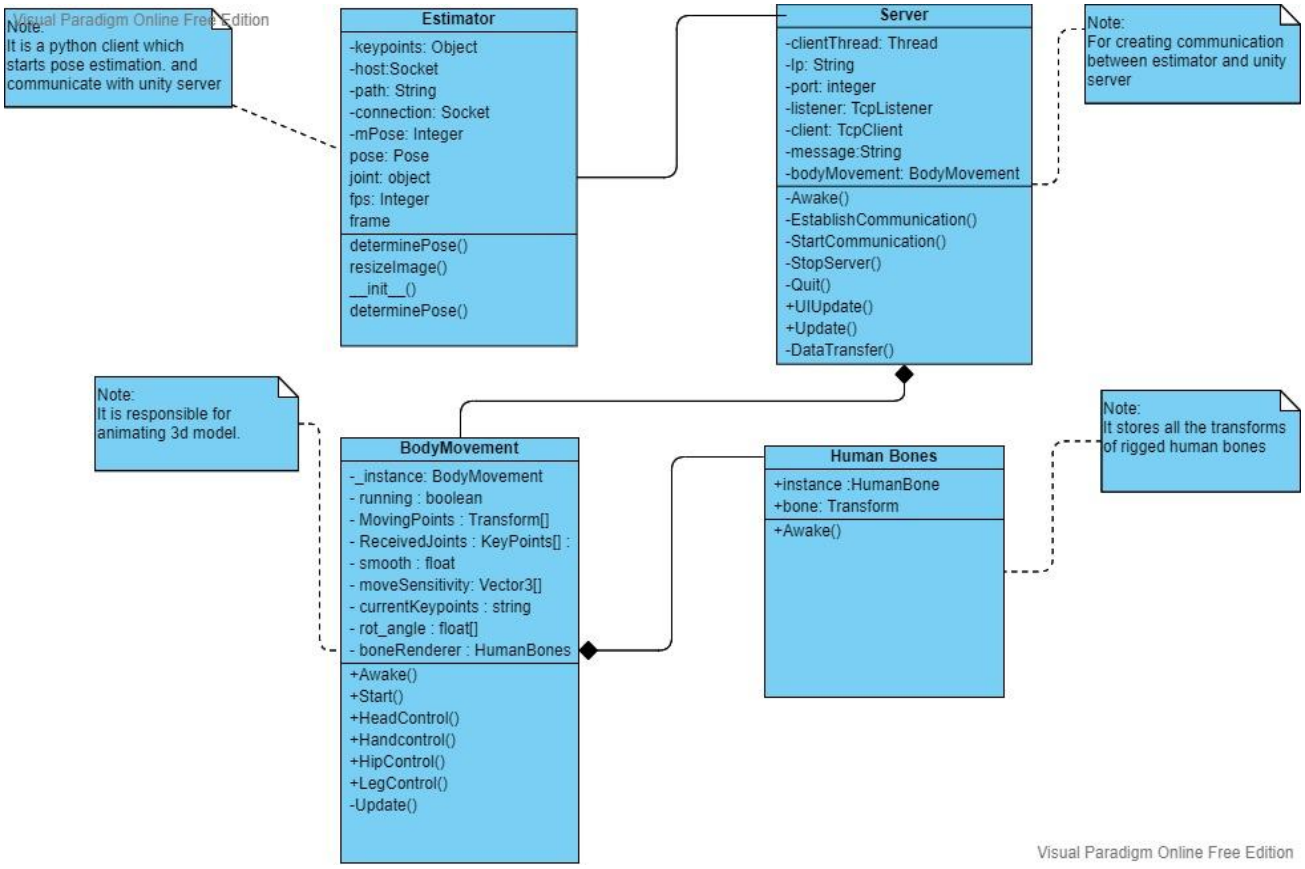


**Figure 4: Class diagram**

# 5. RESULTS

## Results

After completing our system development, we now have a working architecture which showcases the real time tracking of human body keypoints using computer vision. Firstly, we identify the coordinates of each joint via computer vision. The data obtained is then sent to unity via sockets and the 3D model subsequently reacts as to how we move our body parts in real life and moves accordingly.

Here, first we run the application in unity and start the server. Once the server is started unity waits for the client to connect to the server. And, when the client connects to the server it executes pose estimation and continuously shares the data to the unity server. Now, the unity server obtains the data and moves the 3D model joints as per the data received in the form of JSON

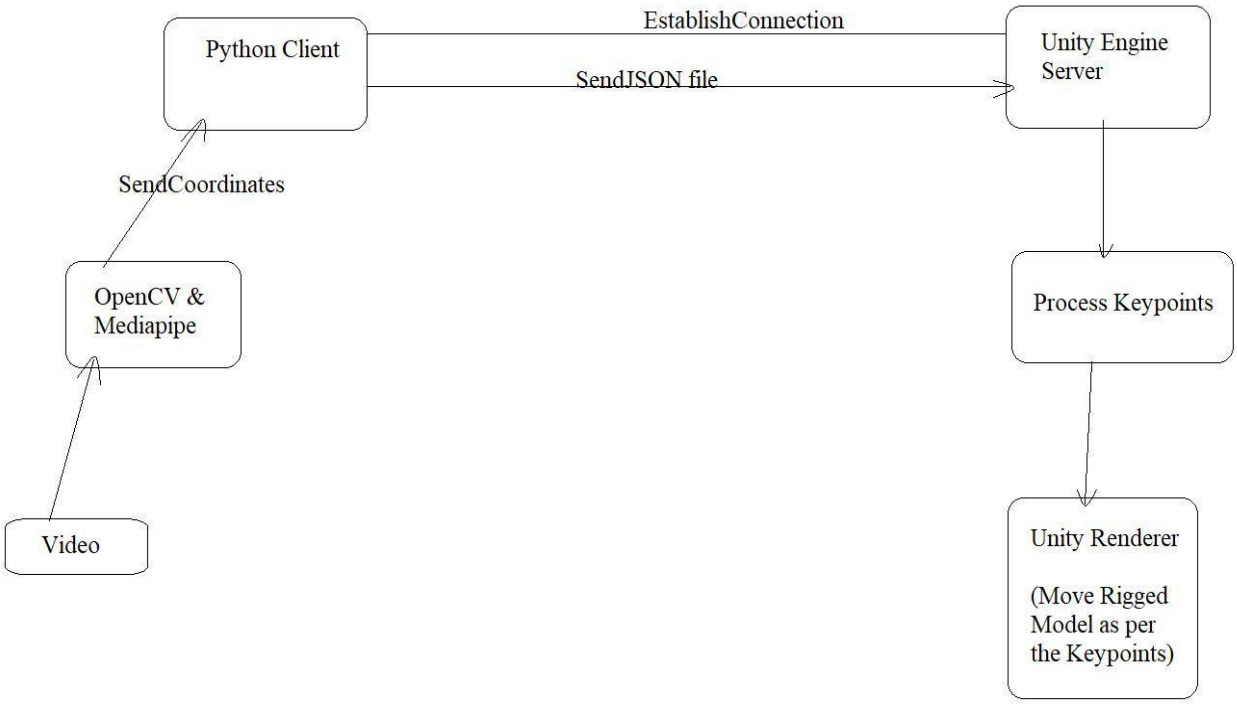


**Figure 5: Architecture of our System**

# 6. PERFORMANCE ANALYSIS & VALIDATION

The results observed was surprisingly on par with how we moved our legs in real time, the only setback being the Frames Per Second (FPS) drops (For an average system having lower specification), which occurs due to the communication delay of the sockets and the processing bottleneck which is created due to a delay in processing each individual image in a video which then maps the coordinates of each anchor like, right leg, left leg etc. thus creating slightly lower FPS in the actual output.

## Performance Analysis

Performance analysis helps us to determine the actual real performance by the means of data collection, rigorous testing and through means of data visualization.

## Data Collection

Firstly, we collect data such as coordinates and the actual video along with the data which is sent back by OpenCV.

## Rigorous Testing

Testing was done by moving our legs as fast as we can to create a worst-case scenario thus helping us to identify the worst performance.

## Data Visualization

We added an FPS counter at the top of the video which showcases how well the overall interaction of the different tools occurred.

## Results

Finally, the results observed were around 10-15fps.

## Validation

As for the validation of data we can compare the FPS mentioned in the visual counter with the actual observed FPS. It was observed that the FPS was close to what we can see on the counter.

For the validation of actual body anchors, we performed the same above method of observing the video and then comparing it with how the 3d models performed.

**Figure 6 :Human Pose Estimation**


**Figure 7: Still animating 3D model**

For analysing the performance of our product we used a profiler. The Profiler is a powerful profiling tool that is built into Unity. The Profiler helps us to diagnose different types of performance problems.

The graph shows how the performance is varying from frame to frame. At a specific frame (represented by white vertical bar), the CPU took 0.25ms to loop from one frame to the next. The graph is showing the performance is varying between less than 1ms, and more than 1ms. In the frames that are less than 1ms, the frame rate will be faster than 1000fps. In the occasional spike the frame rate will be closer to (probably) 500fps.Here, 0.1ms gives 10000 fps. If the spike did hit the 4ms line, then you'd get a fps of 250fps.

Here, At the specific frame our system is using 475.7MB of memory.

And, from above figure 6, we can see that our product is running on overall around 9-15fps for a simple device and upto 25fps for high specification devices.

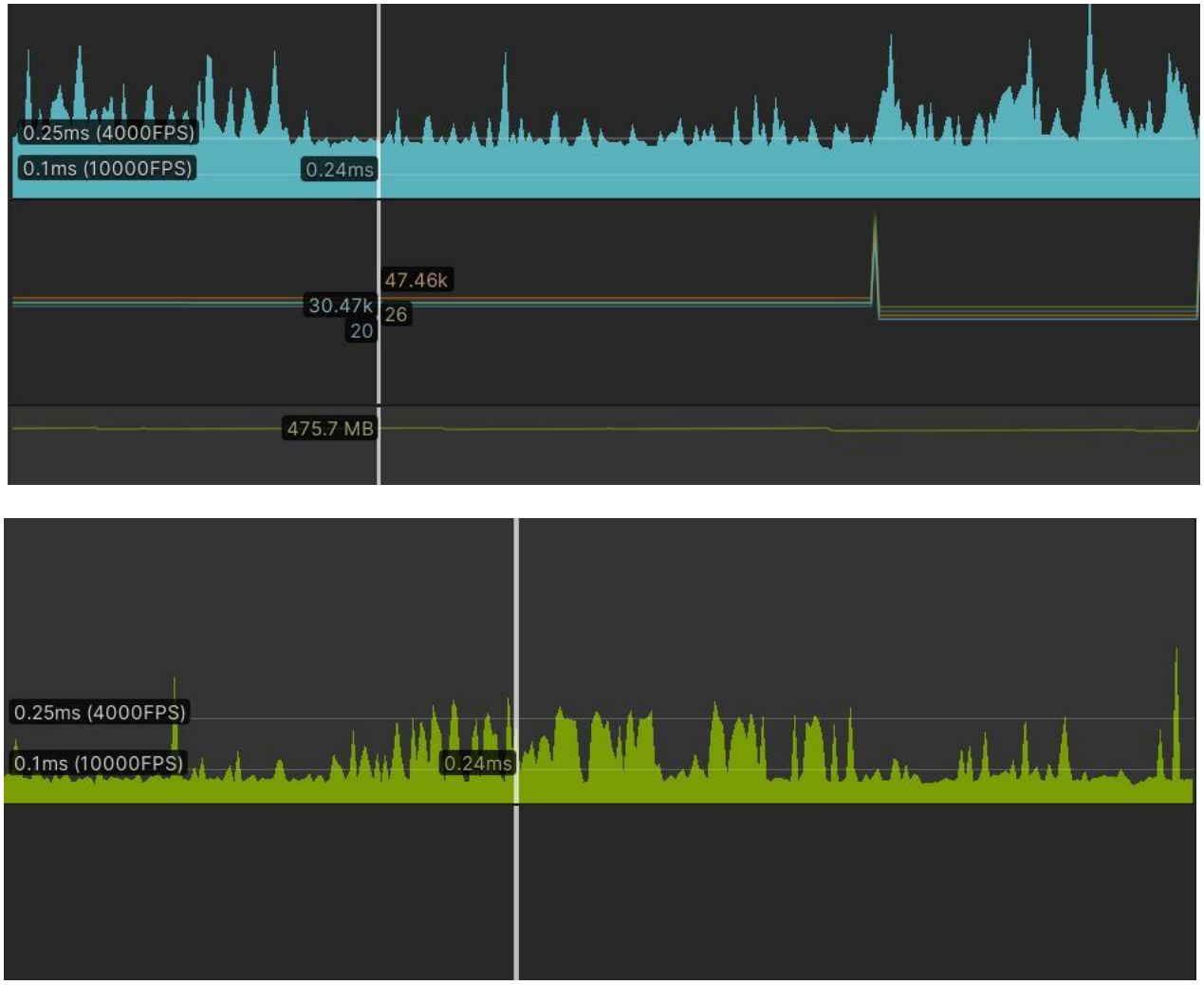


**Figure 8: Profiler performance analysis**

# 8. CONCLUSION

This project now is capable of estimating the human posture and determining the key points of the human body. Though we are not able to move each and every point of the 3D human body. We succeeded in moving parts like, legs, hands, head, spine and hips, etc. Our product is finally capable for:

- Interactive real time animation.
- Providing user service for live interaction with rigged 3D models.
- The camera continuously estimates the human body postures and determines the coordinates of the human key points and renders and moves the 3D model according to it.

# 9. FURTHER WORKS

Though we have successfully moved major parts of the 3D model as per the movement of the human body. But, we can still work further and move the rest of the parts like, hand and leg fingers, eyeballs, nose, etc. and make the animation more realistic and alive.

We can also record the movement of the 3D model and create an animation file (.anim) which can be imported and exported into multiple 3D models and can be used in creating games, movies, etc.

Since we are capable of moving virtual characters using real human posture, we can implement this technology for creating AR/VR conference technology where our virtual clones in a virtual space can be controlled.

# 10. REFERENCE


[1]Unity User Manual, https://docs.unity3d.com/Manual/index.html ,
https://learn.unity.com/tutorial/diagnosing-performance-problems#5c7f8528edbc2a002053b597

[2] Video to Animation Guide,
https://medium.com/analytics-vidhya/transforming-2d-video-to-3d-animation-dc48cf1227f2

[3] OpenCV Reference, https://learnopencv.com/getting-started-with-opencv,

[4] Pose Estimation Guide,
https://google.github.io/mediapipe/solutions/pose_classification.html,
https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b

[5] Animating and rigging guide,
https://docs.unity3d.com/Packages/com.unity.animation.rigging@0.2/manual/index.html

[6] Microsoft Kinect Guide,
https://searchhealthit.techtarget.com/definition/Kinect,
https://en.wikipedia.org/wiki/Kinect,
https://www.researchgate.net/publication/348384835_Evaluation_of_the_Azure_Kinect_and_Its_Comparison_to_Kinect_V1_and_Kinect_V2

[7] RADiCAL Guide,
https://getrad.co/explore/,
https://blogs.nvidia.com/blog/2018/05/08/radical-3d-motion-capture/

[8] DeepMotion,
https://www.deepmotion.com/3d-body-tracking,

[9] Iterative and Incremental Model reference,
https://www.digite.com/agile/iterative-and-incremental-development/
https://agility.im/frequent-agile-question/difference-incremental-iterative-development/