# 3D Modeling in Unity: Utilizing Algorithms and Digital Models to Reinvent Sports Analysis

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

**Brandon Nielsen**

Spring, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

# 3D Modeling in Unity: Utilizing Algorithms and Digital Models to Reinvent Sports Analysis

CS 4991 Capstone Report 2022

Brandon Nielsen
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
brn5jc@virginia.edu

**Abstract**

A local, sports-focused, motion capture technology company, decided to utilize their motion tracking suits to create visual models of their user's movements. The company found that giving their users raw data and graphs was not as effective in conveying the physical movements of an individual's body. By displaying a virtual model swinging a baseball bat, throwing a football, or rowing a boat, the user is quickly and easily able to grasp their movements and pinpoint areas of improvement. In order to create these moving, physical models, data for individual sessions of movement were collected and parsed in the Unity Development Platform because of its animation, rigging, and physics capabilities. After a few months of development, a model was generated that was not only capable of animating the movements of an athlete through a single session, but also able to overlap sessions together, track the movement of individual body parts, and move simultaneously with the athlete currently wearing the suit. This project was a success; however, moving forward, the company is looking to introduce personalized models for more precise movement metrics, as well as utilize Unity's physics engine to produce more features at clients' requests.

## 1. Introduction

Motion capture suits are very advanced and are able to track many different data values such as position, angles between body parts, velocity of individual limbs, and even reaction forces of objects interacting with those people; however, this influx of information is largely vague and incomprehensible for the average user. When athletes use these suits to track their movements, they do not want to sit around analyzing charts and devising formulas to calculate their movements; they understand that their last pitch or swing was slightly off and want to analyze what they did differently or incorrectly. This is the problem that a local company was facing with their clients. They could tell them that the angle of attack of their arm was fifteen degrees lower than needed, but that information was not always the best approach. The solution that this company came up with was to introduce a visual component to their motion analysis. When utilizing this component, an athlete could quickly and easily notice their movements were not in sync with their expectations, while adjusting to correct the mistake in future actions.

The overall goals of the company were two-fold: to create a model and to be able to use the data currently being collected

by their motion-capture suits to facilitate the model's movements; and to create and utilize this model to provide athletes the ability to analyze their movements in revolutionary new ways, improving their abilities in their sport.

## 2. Background

In order to start creating a model for an athlete's movements, it was first important to understand what data was already being collected by the motion capture suit, as well as the specific requirements needed in a development environment. The very first thing that needed to be considered was how the individual session data was being stored. A "session" is the term used to describe an instance in which the suit is turned on, records the athlete performing an action, and is then turned off when the action is completed. The suit records this data in JSON files, with a new data point being created every fraction of a second. This data was being stored in an AWS S3 bucket and could be accessed and extracted from there. Finally, the relevant data from this JSON file was determined to be the position data of each of the sensors. The session data from each of the JSON files was then determined to be necessary and useful for the movement of the model.

In addition to understanding how the suit stores data, the other background knowledge needed was the systems previously being used by the company. The company's app is being developed for IOS, which is using the Apple Xcode environment for Swift.

## 3. Related Words

The idea of sports-focused motion capture technology is not unique, nor is the utilization of a model to visualize the motion of an individual; however, there are very few competitors in the space and most information is kept private. One company that tracks a user's motion data is Fitbit, a company that produces wrist bracelets that capture an individual's walking data as well as heart rate and other metrics [1]. This project differs from a company like fitbit because it utilizes ten different motion capture sensors around the body. Fitbit also does not create models of people performing actions, which this project's company hoped to create.

The other large piece of related work is that of Unity game development. Unity is primarily a video game development platform and is largely utilized as such. Throughout this project, the Unity forums and tutorials were utilized to understand how modeling and rigging works in Unity [2,3]; however, these sources did not help provide information on how to accurately and precisely record movements because they focused on the animations of video game characters. Unlike video game animations, which have a model take certain actions to *look* like they are moving, this project looked to precisely track the *actual* movements of a human. These sources were helpful to understand the basics of Unity animations, but did not further this project's goal afterwards.

Overall, there were not many related works that directly followed the path of modeling an athlete's movements through use of motion capture technology with the use of the Unity development engine.

## 4. Project Design

After deciding to create a visual model imitating an athlete's movements, based on the data collected from motion capture suits, the first step was to decide what development platform should be used. Ultimately, the platform needed multiple things. It needed to have a system in place for utilizing models and rigging their body parts in line with the motion capture sensors. Rigging a model is the process of creating multiple points on the model that can be measured and manipulated. The platform

also needed to be integrated within a Swift environment. This is because the company's app was being developed in Swift and the model would need to be placed in the app. Ultimately, the decision was made to use the Unity development engine.

Unity was chosen for this project because it fits all the requirements listed in the previous paragraph, as well as having extensive documentation, a strong physics engine, and many tools for future improvements on the model. In addition to Unity's development benefits, the class, CS 4730: Game Design, helped teach how to develop code in Unity and C#. It also taught how the Unity physics engine worked. One aspect of this system, translation and rotation functions were heavily utilized later in the development process.

The next step was to create an actual model for physical movement. At first, the team attempted to manually create a model in Blendr, a 3D modeling software, but quickly realized that the models being made were both ugly and incompatible for the movement simulation that the company desired. The team ultimately decided to use a premade model called Ybot.

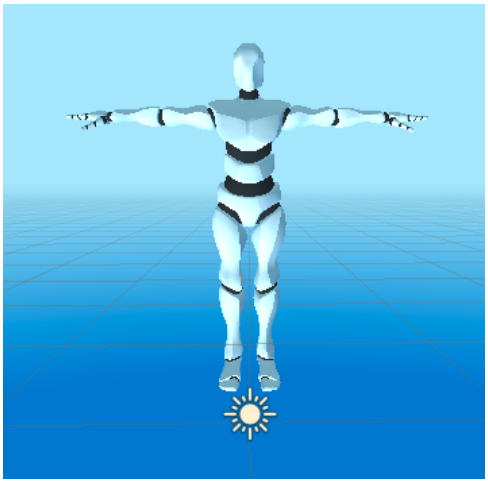

**Figure 1: Ybot Model in Unity Development Scene**

Ybot is a 3D model that looks like a human-shaped robot [4]. It is composed of multiple, separate models for the feet, legs, hands, arms, torso, and head. This separation of body parts turned out to be essential for modeling. This is because it was possible to move the left arm without moving any other body parts. In addition to the benefits of the model, Ybot also came completely rigged. The model was rigged logically so that certain body parts were children of other body parts. An example of this is the hand. There are separate models for the hand and the fingers; however, the fingers are children of the hand, so they also move whenever the hand is moved. Ybot's modeling and rigging allowed the team to focus less time on artistically creating a model and instead on creating the animation system.

Now that all the modeling and software decision making was over, actual integration of the animated model was able to be prioritized. The company's app was already able to grab session JSON files from the AWS S3 buckets, so a function was developed to send that data to a Unity instance. Once inside Unity, the JSON file was able to be parsed by premade functions to single out position data for each of the body parts at each interval. These data points were stored arrays and accessed each frame to acquire the next set of data points. The ideas taught in Data Structures and Analysis 1, specifically list and array data structures, were useful in this process. The physics functions discussed earlier, translate and rotate, were called each frame on all the body parts and the direction to move was calculated by determining the distance between the current point and the point to move to. In order to reach child body parts and move them, a depth-first-search algorithm was utilized, as was taught in Data Structures and Analysis 2. This eventually led to the team successfully creating a model that moves based on an athlete's movement data.

After creating the model, it was determined that there were multiple areas

that the model may be improved upon. One of these areas was that athletes wanted to be able to view their successful sessions alongside the current session to see discrepancies in their movements. This addition was created by adding in a system with multiple models and sessions being stored in lists. Each frame would animate the model at the x position in the model list with the session data at the x position in the session data list. Therefore, athletes were now able to visualize multiple sessions at once.

The next addition to the modeling system was to be able to track the path of specific body parts. This addition utilized the trail renderer tool in Unity [5]. Whenever the user desired to track a certain body part, the trace tool was used on that body part to draw a line based on its movements.

Finally, the last addition to the modeling system created was the creation of "live play." Live play was a system by which an athlete could wear the motion capture suit and the model would move in real time with the athlete. This was created by using the movement code for session-based animations, but instead of reading data from a session, the suit sent data to the app, which would transfer that data to Unity every frame. This allowed the athlete to record and visualize their movements in real time, reducing the time they needed to spend uploading and waiting for session data.

**5. Results**

The creation of an animated model was a success. The team successfully created a model that takes data from the motion capture suit and utilizes it to manipulate a model. Furthermore, multiple quality-of-life advancements were made to that model, such as being able to model multiple sessions at the same time and the "live play" mode.

The customers were also very impressed with the new system. Previously, they were required to upload sessions via the suit and analyze graphs and data points to understand their actions. Now they are able to instantly record and visualize their movements, as well as compare their current movements with previous sessions. Users were very impressed with the model.

In one case, the company displayed their suit to a professional long-distance drive golf player. His goal is to hit a golf ball as far as possible. After trying out the suit, he was able to analyze his swing with the visuals and pinpoint and inaccuracy with it. Within ten minutes he was able to hit the ball ten yards further, a feat which he said should take multiple months of training. The model was seen as a success by him and the rest of its users.

**6. Conclusions**

Motion capture technology is revolutionary in its capabilities to analyze an athlete's movements; however, it is sometimes too daunting for its users. Being presented with vast amounts of data can be confusing and unhelpful. The creation of a visual, moving model simplifies the data and allows users to focus on actually improving themselves as athletes. Moving forward, this company plans to heavily focus on this modeling system and become a central company in the sports analysis industry.

**7. Future Work**

There are multiple ways in which the model could be further improved or new additions added to it. The main focus of the company moving forward is to personalize the model. The model was originally bought from outside the company and its design does not specifically represent the company. The next step is to actually go in and build a personalized model for the company, as well as multiple more models for individual users. These personalized models serve to please the user's sense of style, but they will

also provide better data. All people are shaped differently and have different proportions. Using a single model does not encapsulate the fact that one person's forearm may be 18 inches long and another's could be 20. There are small inaccuracies in the movement of the model due to humans having different shapes and adding personalized models would help fix them.

**References**


1. "Smart Technology to help you transform your health," *Fitbit* . [Online]. Available: https://www.fitbit.com/global/us/technology. [Accessed: 27-Apr-2022].
2. "Unity forum," *Unity Forum*. [Online]. Available: https://forum.unity.com/. [Accessed: 27-Apr-2022].
3. "Intermediate scripting," *Unity Learn*, 13-Aug-2019. [Online]. Available: https://learn.unity.com/project/intermediate-gameplay-scripting. [Accessed: 27-Apr-2022].
4. "Characters, Ybot," *Mixamo*. [Online]. Available: https://www.mixamo.com/#/?page=1&query=ybot&type=Character. [Accessed: 27-Apr-2022].
5. U. Technologies, "Trail Renderer Component," *Unity*, 23-Apr-2022. [Online]. Available: https://docs.unity3d.com/Manual/class-TrailRenderer.html. [Accessed: 27-Apr-2022].