Generating In-Game 3D Self-Expression using Motion Diffusion Model

Daniel Cha[1], Haechan Je[2], Jaehoon Lee[3], Suho Park[4], and Yonghyeon Jo[5]

[1] College of Engineering, Advanced Materials Sciences and Engineering
[2] College of Engineering, Systems Management Engineering
[3] College of Engineering, Civil, Architectural and Environmental System Engineering
[4] College of Engineering, Mechanical Engineering
[5] College of Science, Physics

Sungkyunkwan University, Seoul, South Korea
Capstone Design Project 2023

Abstract. In this project, we present a novel approach to in-game 3D self-expression using a modified motion diffusion model in the Unreal Engine. Our goal is to create a plugin that enables users to easily and intuitively customize their character's animations in real-time gaming by only writing some text. We describe our modified motion diffusion model and its implementation in Unreal Engine. Our approach allows users to generate diverse and natural-looking animations with minimal effort, enhancing their immersive gaming experience. We present simple demo game that demonstrate the effectiveness and efficiency of our approach, showing that users can quickly create personalized animations using our plugin. This work provides a significant contribution to the field of in-game character animation and self-expression, and we believe it has the potential to revolutionize the way users interact with their characters in real-time gaming.

Keywords: Motion diffusion model · Unreal engine · real-time self-expression.

1 Introduction

Video games have evolved significantly in recent years, offering players more immersive and personalized experiences. One crucial aspect of these experiences is character animation, which plays a vital role in conveying personality, emotions, and actions of game characters. Traditional approaches to character animation in video games involve manual key-framing, motion capture, or procedural animation techniques. However, these methods have some limitations. Manual key-framing is a time-consuming and labor-intensive process that requires skilled animators, while motion capture can be expensive and require specialized equipment. Moreover, the number of animations that can be provided through these methods is often limited, leading to repetitive and less diverse animations for characters in the game.

In contrast, our proposed approach to in-game 3D self-expression using a modified motion diffusion model provides a new way for users to customize their character's animations in real-time gaming without the need for any specialized equipment. Our plugin enables users to create diverse and natural-looking animations by simply writing some text. This approach is intuitive and easy to use, allowing for a more personalized and immersive gaming experience.

Our approach builds on the motion diffusion model, which is a powerful tool for generating motion using only text. By modifying this model, we aim to provide a more efficient and effective way to create personalized animations for characters in the game.

Also, this work provides a significant advancement in the field of in-game character animation and self-expression, overcoming many of the limitations of traditional animation techniques. We believe that our project has the potential to revolutionize the way users interact with their characters in real-time gaming, providing a more intuitive and personalized gaming experience.

2 Background

2.1 Human Pose Estimation

Human pose estimation is a task in computer vision that involves detecting the locations of a person's joints and estimating their posture from an input image or video. It has many practical applications, including activity recognition, motion analysis, and virtual reality. In the context of video games, human pose estimation can be used to create more realistic and natural-looking character movements.

Fig. 1. Human pose estimation

However, traditional approaches to human pose estimation can be computationally expensive, especially when dealing with complex movements and interactions between characters in the game. To overcome these challenges, researchers have proposed various approaches that leverage machine learning techniques, such as convolutional neural networks (CNNs), to detect and estimate human poses in real-time. These techniques have shown promising results and have been applied in several video games to create more immersive and dynamic character animations.

2.2 MDM: Human Motion Diffusion Model

Motion Diffusion Model (MDM), is a carefully adapted classifier-free diffusion-based generative model for the human motion domain. MDM is transformer-based, combining insights from motion generation literature. A notable design-choice is the prediction of the sample, rather than the noise, in each diffusion step. This facilitates the use of established geometric losses on the locations and velocities of the motion, such as the foot contact loss. MDM is a generic approach, enabling different modes of conditioning, and different generation tasks. This model is trained with lightweight resources and yet produce excellent results on leading benchmarks for text-to-motion and action-to-motion. The MDM framework has a generic design enabling different forms of conditioning.[1]

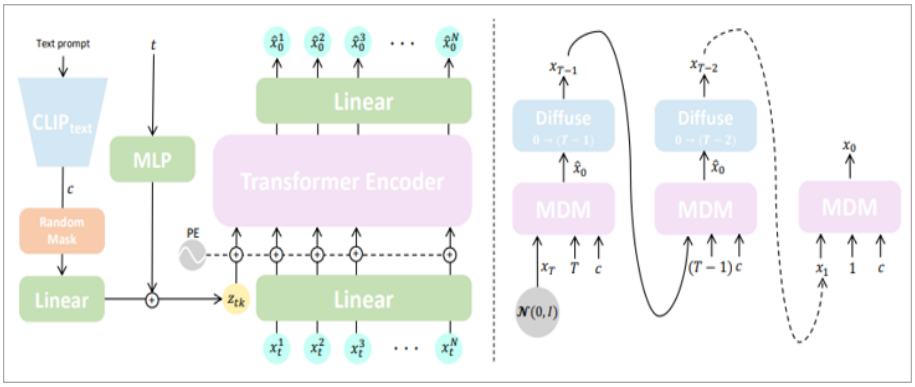

Fig. 2. MDM overview[1]

2.3 FLAME: Free-form Language-based Motion Synthesis & Editing

FLAME, Free-from Language-based Motion synthesis and Editing is the model that integration of diffusion-based generative models into the motion domain. FLAME can generate high-fidelity motions well aligned with the given text. Also,

it can edit the parts of the motion, both frame-wise and joint-wise, without any fine-tuning. FLAME involves a new transformer-based architecture, devised to better handle motion data, which is found to be crucial to manage variable-length motions and well attend to free-form text. FLAME showed outstanding generation performances on three text-motion datasets: HumanML3D, BABEL, and KIT. Editing capability of FLAME can be extended to other tasks such as motion prediction or motion in-betweening, which have been previously covered by dedicated models.[2]

2.4 Unreal Engine

Unreal Engine is a 3D computer graphics game engine developed by Epic Games, first showcased in the 1998. Since the development, it has been used in a variety of genres of games and has seen adoption by other industries, most notably the film and television industry. Unreal Engine is written in C++ and features a high degree of portability, supporting a wide range of desktop, mobile, console, and virtual reality platforms. The latest generation, Unreal Engine 5, was launched in April of 2022, and its source code is available on GitHub and commercial use is granted based on a royalty model.

3 Related Work

3.1 Deep Motion

Deep motion is an one example of a web application which uses machine learning and AI techniques for converting video to 3D animation into a game character. It uses AI motion capture technique, which captures and reconstructs full-body motion, including face and hand tracking. Deep Motion's AI-based approach to motion capture allows for highly accurate and natural-looking animations, making it an attractive option for developers looking to create high-quality character animations. With the increasing demand for immersive and realistic gaming experiences, the use of AI-based animation technology like Deep Motion is likely to become more prevalent in the game development industry.

3.2 Roblox

Roblox is an online game platform and game creation system developed by Roblox Corporation that allows users to program games and play games created by other users. Recently, Roblox integrated the Deep Motion's 3D animating technology and Vroid VRM Custom Character support which is a massive UI overhaul for easier and quicker animation creation. Users can utilize this in order to create a Roblox default game character by uploading .VRM file format character files which is used for AR/VR and diverse live streaming apps.

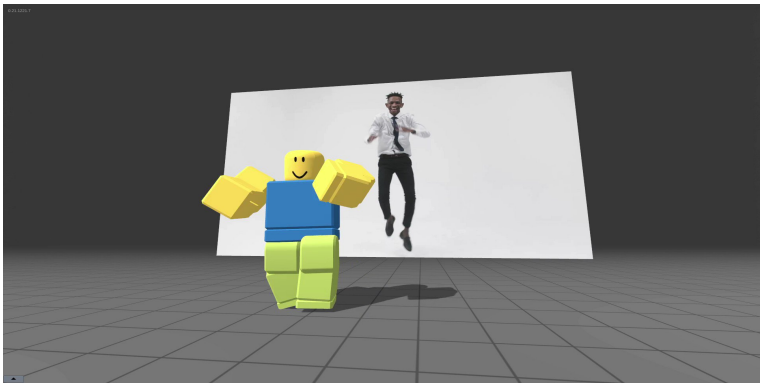

Fig. 3. Deep Motion applied in Roblox[3]

3.3 Kinetix

Kinetix is a company which developed and launched a no-code 3D creation tool powered by AI. It assures a democratized access to 3D animation for all creators by the user-friendly platform enabling User-Generated Content(UGC). The company aims to empower creator's self expression through emotes that can be used on any avatar in every virtual world and further interoperability with their developed game engine package.

4 Problem Statement

Creating 3D animations and applying them to game characters is a complex and challenging process that requires the use of advanced software and techniques. While there have been significant advancements in animation technology over the years, there are still several major problems that must be addressed to create high-quality, natural-looking animations that accurately represent the intended motion.

4.1 Necessity of human posing

One of the primary challenges of creating 3D animations is the necessity of human posing. Traditionally, animators have had to create poses for characters manually, which can be time-consuming and require a high level of skill and expertise. For example, to create a motion for a game character to do b-boy dance, a real human must first do a b-boy dance in advance. If the developer cannot find a person who can actually do a b-boy dance, creating the same motion will be difficult. This process is particularly challenging for complex or dynamic movements, and can result in animations that appear stiff or unrealistic.

Fig. 4. Example of human posing while making emote in real game[4]

Furthermore, animators must ensure that the movements are anatomically correct and that the character's motions align with the intended narrative or game-play mechanics. This can require significant iteration and refinement to achieve the desired result.Just like the former example, the difficult it is for the human to pose the motion, the harder to create the 3D animation for the same motion. Furthermore, it will be impossible for a developer to create an in-game motion that humans are unable to pose which is a big limitation.

4.2 Mismatch of skeleton framework between motion data

For a 3D motion data to be applied to an in-game character model, the 3D motion data must be converted to a file format that is compatible in the specific game development software. For example, usual 3D motion data is in a form of .bvh file format, while the game engine this proposal is about to use, the Unreal Engine, uses .fbx file format. For the Unreal Engine to be able to use the .bvh motion data, the conversion of .bvh file to .fbx file is mandatory. However, this process is not easy. Both motion data .bvh file and .fbx file consists information about skeletons and joints for the moving character and there is a high possibility that there is a mismatch of the two skeleton frameworks between two motion data. For this reason, the process of retargeting the skeleton framework of .bvh motion data to match the framework of .fbx is required and this process very complex.

4.3 Usage of only predefined 3D animations

The third challenge is the limitation of predefined 3D animations. In traditional animation techniques, animators must create predefined animations for every possible movement, which can be limiting and may not capture the full range of movements that a character may perform in a game. This can result in repetitive or unnatural-looking animations, and may limit the expressiveness

and self-expression of the character. Moreover, creating bespoke animations for each game can be a prohibitively time-consuming process, particularly for larger games with many different characters.

5 Proposed Solution

We propose the creation of a plugin for the Unreal Engine software that can generate 3D animations for game characters in real-time, without the need for manual posing or predefined animations. This innovative plugin can be considered as a function that offers a specific feature or capability to the Unreal Engine, rather than a service that needs ongoing maintenance and support. As a result, game developers can benefit from this powerful tool to create more expressive and natural-looking animations for their characters, without the requirement for complex motion capture data or bespoke animations.

The plugin will be made publicly available as an open-source library or API, enabling developers to customize and modify the functionality to suit their specific needs. This approach allows for a high degree of customization and flexibility, enabling game developers to create unique and immersive experiences for their players. The potential applications of this technology are immense, and we anticipate that it will revolutionize the way game developers approach animation creation.

Our plugin utilizes advanced machine learning techniques, motion diffusion model, to create animations in real-time. This model takes text as an input, which means that humans do not have to act themselves to create an animation for the game character. By leveraging the deep learning capabilities of the Human Motion Diffusion Model, game users can create their own customized animation using only in-game text system. This will enable the creation of animations that are hard for humans to pose themselves, resulting in a wide variety of animations for the game character without the need for computer graphics or human actors. Ultimately, this will lead to higher user satisfaction and a higher level of communication between users through a variety of emotions and animations for their game characters.

In conclusion, our proposed method of creating a plugin for the Unreal Engine software that generates 3D animations for game characters in real-time, using motion diffusion model, has the potential to revolutionize the way game developers approach animation creation. By using our plugin, we aim to empower developers to customize and modify the functionality to suit their specific needs. By utilizing advanced machine learning techniques such as motion diffusion modeling, we provide infinite scalability to the games that use our technology, enabling more immersive and responsive gameplay.

6 Project Plan

6.1 Weekly Plan

	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
Retargetting Animation (Skeleton)	■	■	■						
Model performance and speed improvement		■	■	■	■	■			
Adjustment model to Unreal Engine			■	■	■	■	■		
Implement plug-in function library						■	■		
Implement simple demo							■	■	
Packaging and deploy								■	■

Fig. 5. Weekly Schedules Details

6.2 Role Assignment

CV/AI	Suho Park	Motion diffusion model performance improvement & Model training
	Daniel Cha	Data collection & Model training
Unreal Engine	Jaehoon Lee	Adjustment of model to Unreal Engine
	Yonghyeon Jo	Plugin function library implementation
	Haechan Jae	Building demo project and animation post-processing

Fig. 6. Role Assignment Details

References

1. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model (2022).

2. Kim, J., Kim, J., Choi, S.: Flame: Free-form language-based motion synthesis editing (2022).
3. DeepMotion youtube (2021.09), https://youtu.be/hPIO7DPLnUs. Last accessed 4 Oct 2017
4. PUBG:BATTLEGROUNDS youtube (2020.06), https://youtu.be/qFrl20Eg0qM.