Final Year Project 2024

Department of Computer Science

University of Hong Kong

Natural Language-Driven AI Avatar Motion with Video-Based Motion Data

Interim Report

Group members

3035767724	ASJAD, Marzukh Akib	(BEng(CompSc))
3035777975	OLANO, James Daniel Cubillas	(BEng(CompSc))
3035778175	LAU, Ho Yu Mikael	(BEng(CompSc))

Supervised by

Professor KOMURA, Taku

21 January 2024

I. Abstract

Significant advancements have been made in the field of Artificial Intelligence (AI) recently, with the introduction and development of frameworks for various applications, including text-to-speech, text-to-image, text-to-video, and text-to-music. However, when it comes to text-to-video specifically, there is still a lack of a reliable product that can be confidently utilized in various video production scenarios. This report introduces AniGEN, a powerful software solution for text-to-animation conversion. AniGEN has been specifically developed to tackle the challenges associated with creating highly customizable character animations for game development and 3D animation videos. Leveraging cutting-edge findings in Computer Graphics and utilizing modern and efficient AI models for natural text processing, AniGEN aims to revolutionize the creation of animations from text prompts. The development of AniGEN not only benefits individuals seeking to quickly prototype videos and incorporate them into their projects, but it also provides a foundation for further modifications and serves as a catalyst for innovation in the field. We envision AniGEN to showcase the incredible possibilities that emerge when different fields intersect in a captivating and enjoyable manner. It is our hope that our product inspires others to build upon this idea, scaling and improving it in the future, thereby establishing AniGEN as a pioneering software at the intersection of computer graphics and natural language processing. The report subsequently dives into the specific problem that AniGEN aims to solve. Our solution is based on multiple academic and technological successes in the field of computer graphics and artificial intelligence, which serves as the technical foundation for AniGEN.

II. Acknowledgement

Special thanks are given to Professor Taku Komura for supervising the project, Professor Yu Yizhou for serving as second examiner of the project, and Mr. Mingyi Shi for providing additional resources for the project.

III. Table of Contents

IV. List of Figures

V. List of Abbreviations

- Artificial intelligence (AI)
- Three-dimensional (3D)
- Natural Language Processing (NLP)
- Non-Linear Animation (NLA)
- User interface (UI)
- Cascading Style Sheet (CSS)
- Application programming interface (API)
- Real-time Stylized Motion Transition (RSMT)

1. Introduction

This section will discuss the background, motivation, objectives and deliverables of the progress report.

1.1 Background

Video game developers and visual effects artists nowadays rely on motion capture technology to give their audience a sense of realism and immersion through the way the characters move. Robotic and laggy movements in games or movies can do the opposite as it may look unnatural to the audience. Thus, motion capture technology addresses that problem by simply recreating human movements from real life into the digital realm with the use of mocap suits fitted with clear markers and multiple camera setups.

1.2 Motivation

As the demand for AAA video games and realistic visual effects in movies increases, independent video game creators and filmmakers have limited choices in terms of creating smooth and realistic animations since high-quality motion capturing costs are extraordinarily expensive [1]. Other than cost, the technical knowledge needed for setting up and using this technology poses a significant technical barrier for these creators. Currently, independent artists are left with a few options such as rotomation, which is a form of three-dimensional (3D) animating on top of a video, and software like DeepMotion [2] where motion data can be directly extracted from videos. Each of these solutions has its own drawbacks such as the process being time-consuming and dependency on video quality.

1.3 Objectives and Deliverables

Thus, for our project, the goal is to tackle these issues by introducing a web application that is like ChatGPT where users can input commands to control a 3D model utilizing natural language processing and artificial intelligence. This web application is called AniGEN, which combines two words “Animation” and “Generative” together. By keeping this app free and easy to use, it might be able to be the go-to alternative for indie creators. The project will comprise 4 components. Firstly, AniGEN will be coded using React as the front end and Python as the back end which will use local storage (MySQL) as the database for the motion datasets. Secondly, a natural language processing system from OpenAI will be implemented to allow users to generate their own animations. Lastly, personal laptops will be used as cloud hosting for retrieving data from the databases. The remainder of this report will include the following: the methodology and technologies used for developing the web app, the initial design made using Figma, the implementation and back end developments, the upcoming project schedule, and future improvements.

2. Literature Review

In this section, we analyze several academic reports in our field that have inspired our final year project, along with other technological platforms that have influenced the development of AniGEN. Section 2.1 focuses on the accomplishments in the field of natural language processing, while section 2.2 highlights the state of the art technology behind the motion blending process.

2.1 Leveraging NLP

In the field of Natural Language Processing, OpenAI's GPT (Generative Pre-trained Transformers) model has established itself as a leading large language model (LLM) [3] in the field of natural language processing (NLP). This state-of-the-art model exhibits exceptional proficiency in comprehending intricate natural language queries, making it invaluable across a wide range of applications. As a result, it serves as the underlying technology for many popular NLP applications, including text-to-image [4] and text-to-video technologies [5]. Furthermore, there have been successful implementations of NLP techniques to query and retrieve data from databases [6]. The rationale behind this proposition is straightforward: if a model can be trained to accomplish tasks based on natural language prompts, it can also be trained to retrieve information effectively from a database. This serves as the foundation for AniGEN's innovative input processing technology which involves the recognition of prompts and thus enabling the system to retrieve relevant information from a motion data database. Therefore, AniGEN can effectively respond to user input by fetching the most relevant motion data available.

2.2 Blending Motions with Non Linear Animations

Non-Linear Animation (NLA) [7] in Blender provides a powerful framework for creating complex animations through the layering and blending of multiple animation strips. Blending within the NLA involves the combination of animation values from different strips in the NLA stack. With a variety of blending modes available [7], such as Replace, Combine, Add, Subtract,

and Multiply, each mode utilizes its own unique formula for blending the values. All these methods incorporate keyframe blending while taking account of the f-curve value and the weight of the properties (denoted as influence). An f-curve [8] consists of a set of keyframes that define the values of the property at specific points in time. These keyframes serve as control points that determine how the property changes over the course of the animation. NLA employs a range of these formulas that incorporate the f-curve value, influence factor, and other animation properties to facilitate blending. However, the intricate details of these formulas are abstracted by the NLA Editor in Blender, relieving users from the burden of explicit manipulation. The editor adeptly handles the complexities involved in blending animations, ensuring a seamless integration of animation sequences. Consequently, we can focus on the animation blending relying on the NLA editor's robust functionalities to effectively combine and blend animations without the need for extensive manual intervention. Furthermore, Blender's Python scripting capabilities enable us to access and automate the motion blending process offered by the NLA Editor.

3. Methodology

Now, we will proceed to examine the comparison and methodology sections. Section 3.1 will present a comprehensive analysis of the outputs generated by RunwayML and AniGEN. Subsequently, the methodology will be explored in greater detail, divided into three subsections. Section 3.2 will provide an overview, while Section 3.3 will focus on the front end implementation of our software, and Section 3.4 will delve into the back end implementation.

3.1 Comparative analysis: AniGEN vs. RunwayML

A detailed comparison between the expected output of AniGEN and RunwayML is portrayed below.

A text prompt of "a man doing a backflip" was submitted to RunwayML, and the expected output from AniGEN is depicted in Figure 3.1a and Figure 3.1b. On the other hand, the output from RunwayML is represented in Figure 3.2a and Figure 3.2b.

Comparison of AniGEN's expected output and RunwayML's output with the same input prompt

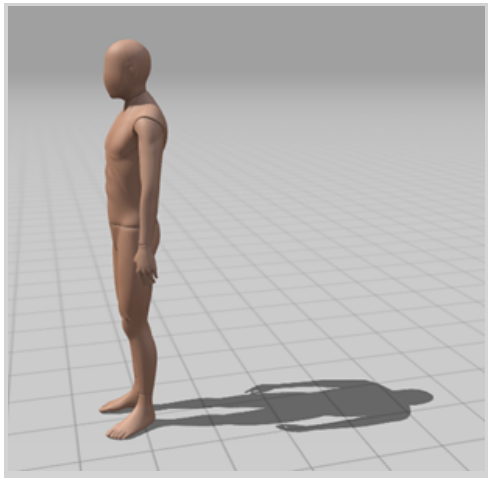

Figure 3.1a: *A 3D model of a mannequin standing at time t = 0 seconds of a clip as expected to be generated by AniGEN.*

Figure 3.2a: *A man doing a backflip at time t = 0 seconds of a clip generated by RunwayML.*

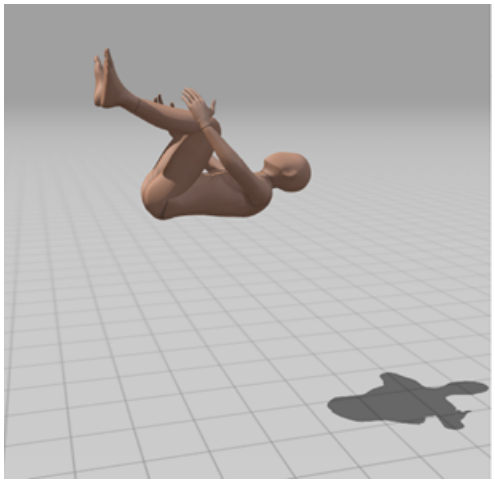

Figure 3.1b: *The same 3D model in the middle of a backflip at time t = 1 second of the same clip as expected to be generated by AniGEN.*

Figure 3.2b: *The same man doing a backflip at time t = 1 second of the same clip as generated by RunwayML.*

The clip in Figure 3.1a and Figure 3.1b is from MIXAMO [9] which is a cloud-based animation library maintained by Adobe. The output from MIXAMO is similar to our expected output from AniGEN (the reasons for the similarity are explained in the section 3.4). The clip is roughly 4 seconds and we have shown the first frame (time t = 0 seconds) of the expected output in Figure 3.1a and a frame at time t = 1 second in Figure 3.1b. The output video from RunwayML application is also a clip of 4 seconds. Figure 3.2a is the first frame of the output video at time t = 0 seconds and Figure 3.2b is the frame of the video at time t = 1 second. It can be noticed that there are stark differences in the outputs in comparison to the consistency between the frames for the output in AniGEN. It can also be observed that the model does not get distorted and there is actual relevance with the prompt, although the model itself is a mannequin so the character design does not correspond to the part of the “man” as mentioned in the prompt. The output video generated by RunwayML demonstrates the figure's adherence to the character design, maintaining relevance to the "man" aspect of the prompt, as well as performing the backflip action. However, it lacks the depiction of the motion's beginning and end. Furthermore, the figure itself is distorted from the beginning with his hand being pointed in another direction. These videos cannot be used in music videos or other implementations because of the lack of consistency between the frames. It is precisely to solve these problems that we decided to create AniGEN.

3.2 Overall System Design

The methodology employed in our work encompasses the consideration of user input, the intermediate processes involved, and the output. To illustrate this workflow, Figure 3.3 outlines the separate components forming AniGEN's system design and their connections.

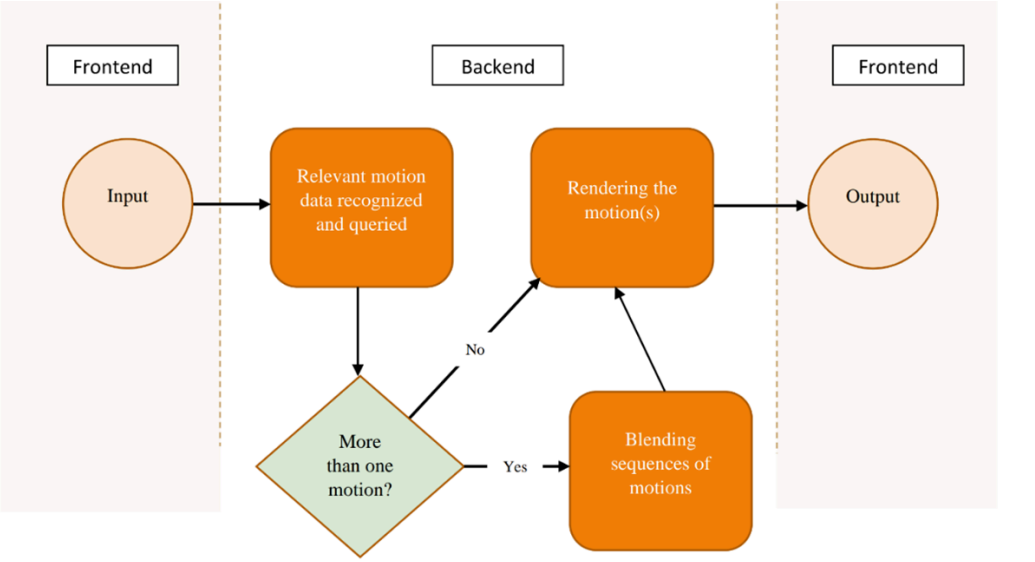

Figure 3.3: *The chart shows the workflow of the software. A user gives their input which is carried to the back end from the front end. The relevant motion data is recognized using NLP from the user prompt and then retrieved from the database. The motions are blended using Blender NLA if there is more than one motion. Afterwards, the sequence is rendered in a 3D animation software and the output video is then returned to the user.*

Figure 3.3 illustrates our system design, which comprises both the front-end and back-end components. The front end is responsible for gathering information from the user and providing them with relevant outputs. On the other hand, the back end is involved in the creation of videos using the provided information.

3.3 Front End Methodology Implementation

Our software AniGEN is to be embedded in a web-based application. A snapshot of our application's User Interface (UI) is displayed in Appendix A, created using Figma, a rapid UI prototyping software. The decision to utilize Figma was primarily influenced by the expertise of one of our team members in its usage. This UI design prototype will subsequently be transformed into an actual web page, employing React and Tailwind CSS. These two widely adopted tools are renowned for their ability to develop modern webpages, ensuring seamless implementation of our UI design. React was chosen as our framework for several reasons. It boasts a vast and supportive community [10], making it easier to find resources and assistance when developing our front-end UI. Additionally, React offers an easy setup experience [10] and provides access to numerous libraries and tools that facilitate the creation of visually appealing interfaces. Tailwind CSS was selected because it allows us to write Cascading Style Sheets (CSS) directly within our React code, eliminating the need for separate CSS files [11]. This streamlined approach enhances development efficiency and simplifies the management of styles. In the UI displayed in Appendix A, two key sections should be noted: the Avatar Selection part and the Text Input part. These sections encompass the user input elements.

To enhance the user's creative freedom, we will upload various Avatars in the form of 3D characters (FBX or OBJ file formats) to the web application. The Avatar Selection feature allows users to choose from these available Avatars for their content creation. Initially, for the Avatar selection part, we will publish a few pictures of avatars which the user can scroll through in slideshow format. The Text Input part enables users to enter their prompts or instructions, which will then be submitted to AniGEN for video generation. We will also design a simple popup message to open when the user submits the text to show confirmation of the data passed in. After confirmation, the input text and the selected Avatar information are sent to the back-end server for processing and subsequent video creation.

3.4 Back End Methodology Implementation

The back-end server consists of various Python files that facilitate communication with the front end, the MySQL hosted database server containing motion data files, and Blender, the 3D animation software. We mainly chose Python as the main programming language for two main reasons. Firstly, it is a very popular programming language [12] with sufficient community support in the field of NLP and AI. Thirdly, Python is also the main scripting language for Blender [13].

For our database server, we have opted to utilize MySQL. The choice of MySQL is based on its advantageous features, such as the highly generous storage limitation of 256 TB given by its storage engine InnoDB [14], and its sole dependence on our local storage. Given our current focus on maintaining the software within the local environment, MySQL aligns well with our objectives. Furthermore, our team possesses significant expertise in working with MySQL, making it a suitable and familiar choice for our database needs. To identify relevant motions based on user-input text, we need to perform two essential tasks. Firstly, we require the annotation of the motion data. Fortunately, the motion data from Adobe Mixamo [9] already contains annotated high quality motion data, providing valuable information about each corresponding motion data entry. Secondly, we will employ an AI-powered large language model (LLM) capable of recognizing these annotated motion data and establishing connections with the input text, thereby determining their relevance. For our second task, we will use the GPT-3.5 Turbo Model from OpenAI [15]. It will be used to train on the database and then fetch the most relevant motion data(s) from the database according to the input text.

To render our scenes and animations, we have chosen to leverage Blender, a widely recognized open-source 3D animation software. It is precisely because of this step that there is a considerable similarity of the expected output with the animation clip from MIXAMO. Blender's open-source nature [16] enables easy integration with various third-party services, expanding its capabilities. Although Blender has a steep learning curve, its flexibility and extensive features make it an ideal choice. As the members of the project team are personally familiar with Blender, it further solidifies our decision to use it as our primary rendering software. It supports multiple

forms of rendering, allowing for the selection between high-quality and low-quality rendering based on the number of motions to be rendered. Higher performance is required for a larger number of motion sequences, potentially sacrificing the quality factor. For the purpose of retrieving data from MySQL database, Python and MySQL connectors will be used. Once the rendering process is finished, the resulting video in the MP4 format will be sent back to the front-end UI, specifically to the designated Video section as depicted in Figure 4.1. From there, users will have the ability to play or pause the video and even download it as desired.

4. Results and Difficulties Encountered

Currently, the design of the UI has been completed since November and implementation of some of the UI features has been done as of January.

4.1 Front End Design

AniGEN Main Page UI Design

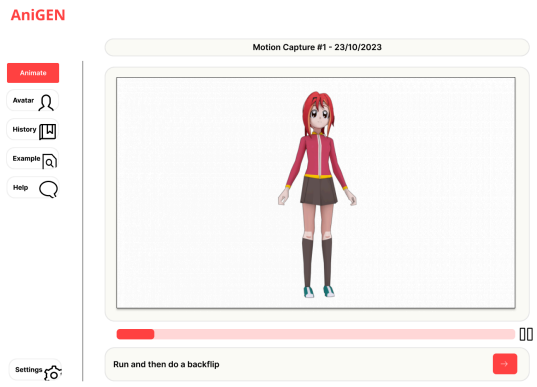

Figure 4.1. *The initial main page UI design in Figma. This will be what the users will first see when entering the page. Currently this is just a design and is not functional.*

Popup Avatar Selection Window UI Design

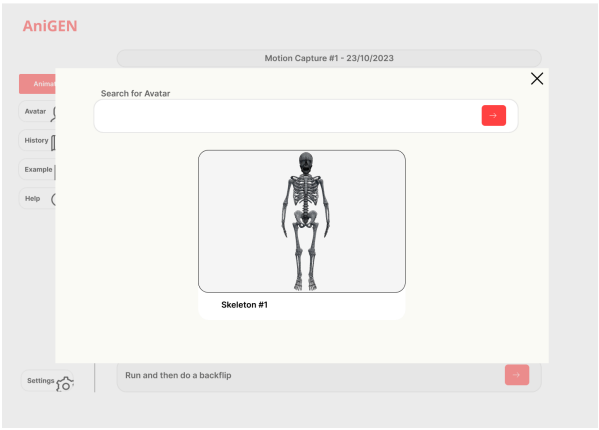

Figure 4.2. *The popup window when clicking the "Avatar" button.*

In Figure 4.1, Figma was used to design the initial UI design of the AniGEN. The main colors used in the design are white and red with an off-white accent on certain places such as the text box and animation display section. To input a prompt, users would need to type in the text box at the bottom of the animation panel. Once the animation is finished processing and rendering, the user can use the playback UI feature to play and pause the motion and rewind to a specific timestamp. On the side of the page, there are buttons for other features and information. Starting from the top, the "Animate" button takes the user back to the main page. Below the "Animate" button, the user can click the "Avatar" button to open the popup window shown in Figure 4.2.

Users can select from a preset selection of avatars. Due to the difference in anatomy size, the animation will have to be re-rendered to be able to view the changed avatar. Custom OBJ and FBX file formats can be dragged and uploaded to the web application in order to use the user's own 3D avatar models.

In addition to these two buttons, there are the "History", "Example" and "Help" buttons. The "History" button shows the user's previous prompts used. Furthermore, the "Example" button displays some sample animations to give the user some ideas as to what can be done using this web application. Lastly, the "Help" button will show a popup of contact information such as email address so as to allow users to ask for help or report any bugs.

4.2 Front End Implementation

AniGen Main Page Implementation

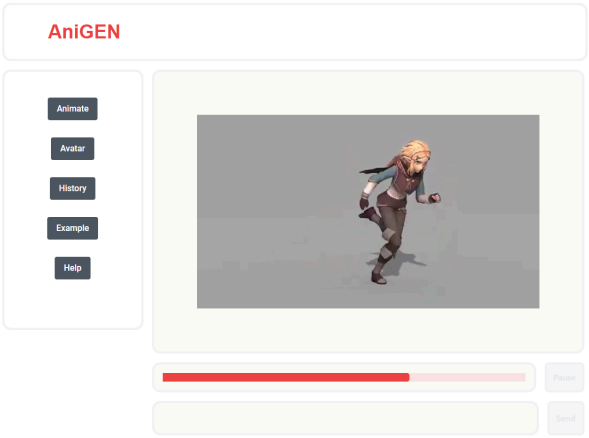

Figure 4.3. *The implementation of the UI design of the main page using React.*

Popup Avatar Selection Window Implementation

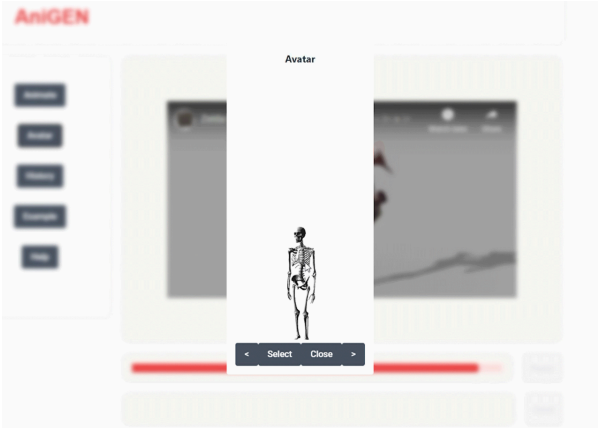

Figure 4.4. *The implementation of the Popup avatar selection window design of the main page using React with the help of react-3d-viewer - a React library.*

Prompt Input and History Window Implementation

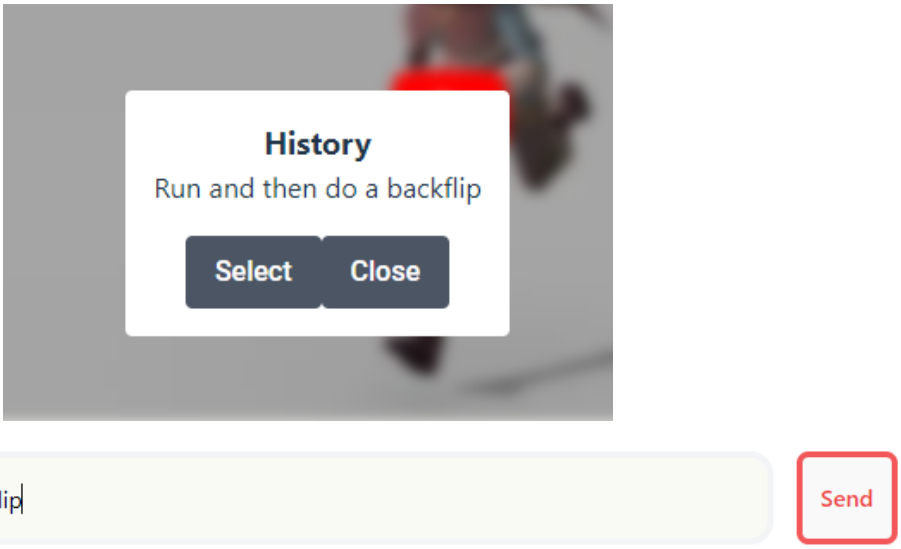

Figure 4.5. *The implementation of the prompt input box and history popup window.*

In Figure 4.3, The React implementation was made to resemble the UI design as close as possible. This page utilizes one main React library called React Player, this library is used for the animation video section and it allows for React to display videos from sources such as local storage and YouTube. In order to control this video, a playback slider is implemented alongside to allow users to go to specific points of the animation with the help of the React library called React Slider. Below the slider is the implemented prompt input box as shown in Figure 4.5, as of right now, it can only print to console and to the history list for now due to the absence of the back end. The history tab can be accessed through one of the side navigation buttons. Another feature that is implemented and can be accessed through them is the “Avatar” button. Clicking the button will open up the implemented popup window shown in Figure 4.4. Utilizing the React 3D Viewer library, it was possible to display OBX files onto the front end. In this implementation, the skeleton OBX model is used to recreate the Figma design of the window; Users are able to drag the model around to view every part of it. Lastly, as shown in Figure 4.5, if a user types in an input like “Run and then do a backflip” inside of the prompt, it will then be displayed in the history window where users can view their previous animations once sent. Currently, the window only displays the previous prompts and is also stored during runtime. Overall, the current features implemented are mostly able to replicate what the intended features are supposed to do.

4.3 Difficulties Encountered

One of the difficulties encountered trying to animate 3D models is the use of motion capture data, which is not standardized. This poses an issue as trying to assign motion data to an incompatible 3D model may encounter compatibility issues when trying to use mocap data from different sources or studios. However, there are alternative solutions available to overcome this difficulty. One option that came up was to use Mixamo, which is a library of pre-made animations specifically made for games [9]. These animations are standardized and can be easily integrated into most of their 3D models, saving time and effort in rigging motion data to incompatible models. Mixamo provides a wide range of high-quality animations that can be applied to various 3D characters in a game or video. By using Mixamo, the quality of rendered animations will be much smoother as the motion data are specifically made for video games, ensuring that they are of high quality and suitable for the intended purpose.

Another problem that impeded development was implementing certain React libraries that caused import and styling issues. Certain libraries used were incompatible with the code base, moreover, some of them required extra steps in order to fully work. One instance of this is when using the React 3D Viewer library, a separate dependency file was needed to make the library work smoothly. Without this, all of the functions would be unable to be utilized. After realizing this, extensive research on more compatible React libraries will be done in order to avoid such issues from happening again.

The Real-time Stylized Motion Transition (RSMT) Python tool was initially used for transitions between motions. The code was created as a deliverable of a research project [17]. However, the project team found the implementation of the code difficult for the purposes of this project. Therefore, for the purposes of blending different motions the project team decided to use Non Linear Animations in Blender instead. Compared to RSMT, NLA is easier to use due to more relevant documentation being available, and NLA is also recognized as an industry standard tool.

5. Proposed Upcoming Schedule

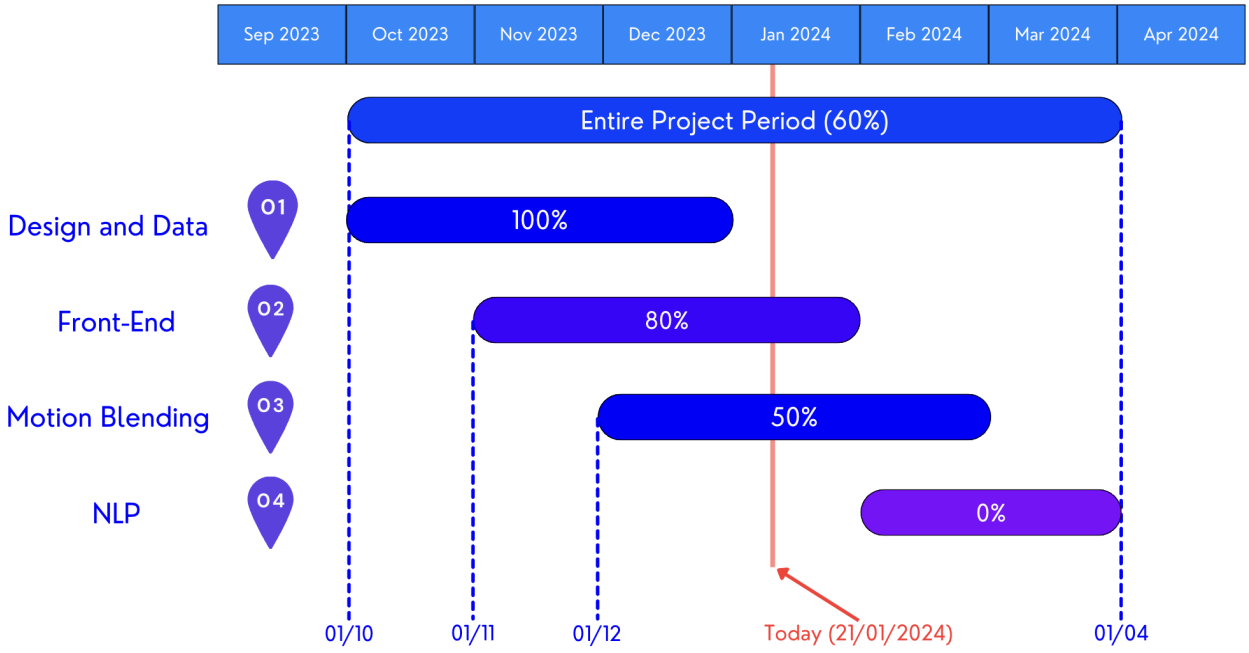

Figure 5. *The upcoming project schedule, up to the expected project completion in April 2024.*

Upon completion of initial tasks, the project is split into three development tracks: one for the software development side, the Blender side and the NLP side. It has been found that developing the transition between motions is complex and time-consuming. Therefore, two-thirds of the development group will be working on researching this topic. Moreover, one-third of the development group will focus on developing the front end and the back end of the web application. All members will work on the NLP side once it is ready, which is anticipated to be after the completion of the Blender NLA setup. This ensures all tasks are done in a timely manner according to the project timeline. Currently, the project is slightly behind schedule. Therefore, the distribution of tasks and focusing on specific areas is necessary to keep up with the original schedule. This section will describe the schedule plan and what will be done during these periods.

The development of the web application, which started in the middle of October last year, will continue until the beginning of February this year. Further improvements such as completing features and CSS styling fixes will be done. For example, the history window will allow the users to download and view previous animations instead of just viewing prompts. The avatar window will also allow for the selection of different 3D character models. Lastly, the most important step is to integrate the frontend with the backend such as being able to send the prompt to the backend and receiving the rendered animation onto the animation panel.

From February to March, there will be more attention on the NLP model implementation. Since OpenAI's GPT-3.5 Turbo LLM will be used, integrating it into the current back end is the only required step for setup. The LLM will take in the input from the textbox in Figure 4.1, and use it to scan over the database of motion data. It will look for the label corresponding closest to the prompt. The accuracy of the NLP model will be tested, looking for at least a success rate of over 80%. To ensure the correctness of tests, the labels themselves will be used as prompts. In the end, synonyms and non-label words will be tested with the NLP model.

6. Conclusion

To conclude, the project team has presented a web app processing natural-language user inputs and returning 3D avatar animation sequences as outputs. The web app facilitates the configuration of 3D avatar animations, even for users and artists who may not necessarily have experience in advanced coding.

This paper discussed the background and objectives of the solution, as well as its methodology. Furthermore, difficulties encountered during the project development and the remaining work plan were also discussed.

As of 21 January 2023, the following work have been achieved:

Firstly, the project team has completed a prototype of the user interface and has implemented parts of it on the front end.

Secondly, the project team has also implemented motion blending through Blender, and has achieved results in smooth transitions between multiple motions.

The project team is still not able to character interactions with the surrounding environment, as it involves highly advanced and in-depth concepts and techniques in character animation and computer graphics. As such, the project is limited to animation of the 3D avatar itself, with no regards to its surrounding environments.

Despite the limitations the project team has encountered during development, this project would potentially serve as a first step towards artificial-intelligence-powered solutions aimed to bridge the skill gap between the vast majority of the user base, and the advanced and in-depth coding required to achieve the solution.

This paper has the potential to be built upon and enhanced, and be used in other applications and fields, such as the generation of retro-style chip music with natural-language parameters, or the full-scale generation of motion pictures from natural-language prompts.

VI. References

[1] Rokoko, "The Complete Guide to Professional Motion Capture." rokoko.com. https://www.rokoko.com/insights/the-complete-guide-to-professional-motion-capture (accessed Jan. 21, 2024).

[2] DeepMotion, "DeepMotion - AI Motion Capture & Body Tracking." deepmotion.com. https://www.deepmotion.com/ (accessed Jan. 21, 2024).

[3] OpenAI, “GPT-3.5 Turbo fine-tuning and API updates,” openai.com. https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates (accessed Jan. 21, 2024).

[4] OpenAI, “DALL·E: Creating images from text,” openai.com. https://openai.com/research/dall-e (accessed Jan. 21, 2024).

[5] Synthesia, “Discover Synthesia’s 15+ Unique Features,” www.synthesia.io. https://www.synthesia.io/features (accessed Jan. 21, 2024).

[6] A. Bazaga, N. Gunwant, and G. Micklem, “Translating synthetic natural language to database queries with a polyglot deep learning framework,” Scientific Reports, vol. 11, no. 1, Sep. 2021, doi: 10.1038/s41598-021-98019-3.

[7] Blender, “Source/Animation/NLA - Blender developer Wiki.” wiki.blender.org. https://wiki.blender.org/wiki/Source/Animation/NLA (accessed Jan. 20, 2024).

[8] Blender, “Introduction — Blender Manual.” docs.blender.org. https://docs.blender.org/manual/en/latest/editors/graph_editor/fcurves/introduction.html (accessed Jan. 20, 2024).

[9] Adobe, “Mixamo,” mixamo.com. https://www.mixamo.com/#/ (accessed Oct. 25, 2023).

[10] Honeypot, “6 best React communities to get information and support,” cult.honeypot.io. https://cult.honeypot.io/reads/6-best-react-communities-to-get-information-and-support/ (accessed Oct. 25, 2023).

[11] Tailwind CSS, “Optimizing for production - Tailwind CSS,” tailwindcss.com. https://tailwindcss.com/docs/optimizing-for-production (accessed Oct. 25, 2023).

[12] Python, “Our community,” Python.org. https://www.python.org/community/ (accessed Jan. 21, 2024).

[13] Blender, “Developer intro/environment - Blender developer Wiki,” wiki.blender.org. https://wiki.blender.org/wiki/Developer_Intro/Environment (accessed Oct. 25, 2023).

[14] MySQL, "InnoDB Limits," dev.mysql.com. https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html (accessed Jan. 21, 2024).

[15] A. Peng, M. Wu, J. Allard, L. Kilpatrick, and S. Heidel, “GPT-3.5 Turbo fine-tuning and API updates,” openai.com. https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates (accessed Jan. 20, 2024).

[16] Blender Foundation, “About — blender.org,” blender.org. https://www.blender.org/about/ (accessed Jan. 21, 2024).

[17] X. Tang, L. Wu, H. Wang, B. Hu, X. Gong, Y. Liao, S. Li, Q. Kou and X. Jin, "RSMT: Real-time Stylized Motion Transition for Characters," in *Proc. SIGGRAPH '23*, 2023, doi: 10.48550/arXiv.2306.11970