Mymensingh Engineering College

Project Report on

Visual AI Assistant: Softsasi AI

CSE-3112: Software Engineering Lab

Supervisor

Engr. Rawnak Ara Chowdhury

MSc in CSE

Assistant Professor & Head of the Department

Department of Computer Science & Engineering

Mymensingh Engineering College

Submitted By

Md Shakil Anower Samrat | Roll: 307 | Reg. No:2850

Izaz Uddin Mahmud | Roll:308 | Reg. No:2851

Nuruzzaman Ashrafi | Roll: 333 | Reg. No:2875

Abstract

This project, "Softsasi AI," is a Visual AI Assistant developed using the Unreal Engine 5 game engine and Blueprint, a visual scripting system, Open AI API as well as C++, JavaScript programming language. The project was undertaken as a partial fulfillment of the requirements for the course "Software Engineering Lab" at Mymensingh Engineering College. The primary objective of the project was to enhance the skills of the team members in game development and create an engaging and Visual AI Assistant for people.

The AI Assistant features high-quality graphics, challenging mechanics, and industry-standard tools and technology. The project team aimed to create AI Assistant that is user-friendly, easy to navigate and can be enjoyed by users on any browser. The assistant may offer advanced functionality for users, and it is a successful demonstration of teamwork and dedication in the development of AI assistant using industry-standard tools and technology.

Contents

Introduction

“Softsasi AI” is a Visual AI Assistant that we developed using the Unreal Engine 5 game engine, Blueprint(a visual scripting system), OpenAI api as well as C++, JavaScript programming language. It’s an assistant that will help you with various tasks and queries using natural language and high-quality graphics. This project was undertaken as a partial fulfillment of the requirements for the course “Software Engineering Lab” at Mymensingh Engineering College by a team of enthusiastic students.

The primary objective of this project was to create an assistant that would be both user-friendly and engaging for people. And we think we’ve achieved that. The assistant features high-quality graphics that are visually appealing and a natural language interface that is easy to use. But the real highlight of the project is the functionality, it’s versatile, intelligent, and it can handle a variety of tasks and queries. We wanted to make sure that “Softsasi AI” is accessible and convenient, so we made sure it can be enjoyed by users on any browser. We understand that some of you may have specific needs and preferences, but we’re confident that you’ll find the assistant useful and entertaining.

This project was a true collaboration and we couldn’t be happier with the outcome. We’ve used our skills and knowledge of programming, game development, and artificial intelligence to create an assistant that is helpful and engaging for people of all backgrounds and interests.

In conclusion, “Softsasi AI” is a Visual AI Assistant that provides users with an interactive and helpful experience as they use natural language and high-quality graphics. It’s a product of teamwork, dedication, and the use of industry-standard tools and technology. If you’re looking for an assistant that will assist you with various tasks and queries, “Softsasi AI” is the assistant for you.

Purpose of the Project

The purpose of the project was to create a Visual AI Assistant that can help users with various tasks and queries using natural language and high-quality graphics. The project team wanted to apply their skills and knowledge of programming, game development, and artificial intelligence to create an assistant that is user-friendly, engaging, and versatile. The project also aimed to demonstrate the use of industry-standard tools and technology such as Unreal Engine 5, Blueprint, Open AI API, C++, and JavaScript. The project was intended to fulfill the requirements for the course "Software Engineering Lab" at Mymensingh Engineering College and to showcase the team's creativity and innovation.

Description of the project

Content: “Softsasi AI” is a Visual AI Assistant that helps users with various tasks and queries using natural language and high-quality graphics. The assistant features versatile functionality, intelligent responses, and engaging interactions.

Some of the tasks and queries that the assistant can handle are:

- Web search: The assistant can search the web for any topic or keyword that the user provides and display the results in a concise and structured way. The assistant can also answer factual questions based on the web search results.
- Graphic art: The assistant can create graphic art based on the user’s prompt or request. The assistant can generate images of various objects, animals, landscapes, or abstract concepts using artificial intelligence and Unreal Engine 5.
- Code generation: The assistant can generate code snippets or scripts based on the user’s specifications or requirements. The assistant can use C++, JavaScript, or other programming languages to create code that can be used for various purposes.
- Content creation: The assistant can create content such as poems, stories, essays, songs, or parodies based on the user’s input or theme. The assistant can use natural language generation and creativity to produce original and entertaining content.

Intended Audience: The assistant is intended for users of all backgrounds and interests who need help with various tasks and queries using natural language and high-quality graphics.

The assistant is designed to be easy to use and understand for anyone who wants to interact with it. The assistant does not require any prior knowledge or experience of programming, game development, or artificial intelligence. The assistant is also designed to be adaptable and customizable for different users and contexts. The assistant can adjust its tone, style, and language according to the user's preferences and needs.

Content Features: The assistant includes multiple features, each designed to help the user with a specific task or query. It also includes a variety of options, settings, and modes to customize the assistant according to the user's preferences. The assistant's graphics are designed to be visually appealing and create a captivating and interactive experience for users.

Some of the features that the assistant includes are:

- Voice recognition: The assistant can recognize the user's voice and convert it into text. The assistant can also speak back to the user using text-to-speech technology.
- Language translation: The assistant can translate the user's input or output into different languages. The assistant can support English Language.
- Mode selection: The assistant can switch between different modes such as Balanced, Creative, and Precise. Each mode has a different level of accuracy, speed, and creativity for the assistant's responses.
- Feedback system: The assistant can receive feedback from the user about its performance and quality. The assistant can use the feedback to improve its functionality and user satisfaction.

Effect/Result to be expected: Users can expect to be helped and entertained by the assistant, impressed by the high-quality graphics, and satisfied by the intelligent responses. The assistant will also provide an opportunity for the users to learn new information, improve their skills, and explore their creativity.

The assistant will help users with various tasks and queries that they may have in their daily lives or work. The assistant will also entertain users with its graphic art and content creation features. The assistant will impress users with its high-quality graphics that are realistic and immersive. The assistant will satisfy users with its intelligent responses that are accurate and relevant. The assistant will also provide users with an opportunity to learn new information about various topics or domains that they may be interested in or curious about. The assistant will also help users improve their skills in programming, game development, or artificial intelligence by providing them with code generation or feedback features. The assistant will also allow users to explore their creativity by providing them with graphic art or content creation features.

Platform: The assistant will be available for users on any browser.

The assistant will be accessible and convenient for users on any browser that supports Unreal Engine 5 technology. The assistant will not require any installation or download from the user's side. The user will only need an internet connection and a microphone (optional) to interact with the assistant.

Additional Features: The assistant also includes a simple and user-friendly interface, making it easy for users to communicate and understand the assistant. The assistant also includes an option to switch between different languages, allowing users to use the assistant in their preferred language.

The interface of the assistant will be simple and intuitive for users to navigate and use. The interface will include a text box for the user to type or speak their input, a button to send their input, and a display area for the assistant's output. The interface will also include a menu bar for the user to access the options, settings, and modes of the assistant. The interface will also include a feedback button for the user to rate or comment on the assistant's performance and quality.

Overall, "Softsasi AI" is designed to provide users with an interactive and helpful experience as they use natural language and high-quality graphics. The assistant is intended for users of all backgrounds and interests and offers versatile functionality, intelligent responses, and engaging interactions.

Platforms & Requirements

Platform Used:

- Unreal Engine 5
- Visual Stdio 2019
- Metahuman Creator
- Epic Games
- Windows 11 Pro 64 bit
- Metahuman SDK
- Blender
- Scandy Pro

Requirement to running the software:

- Windows 10 64 bit and above
- Dual-core Processor with clock speed at least 3 Ghz
- Dedicated GPU memory 2 GB
- 8 GB Ram
- Availability of Device’s Storage (2 GB) or more
- Need Cloud Services to running in other devices.
- IPhone 11 pro for Scanning Metahuman

Design & Methodology

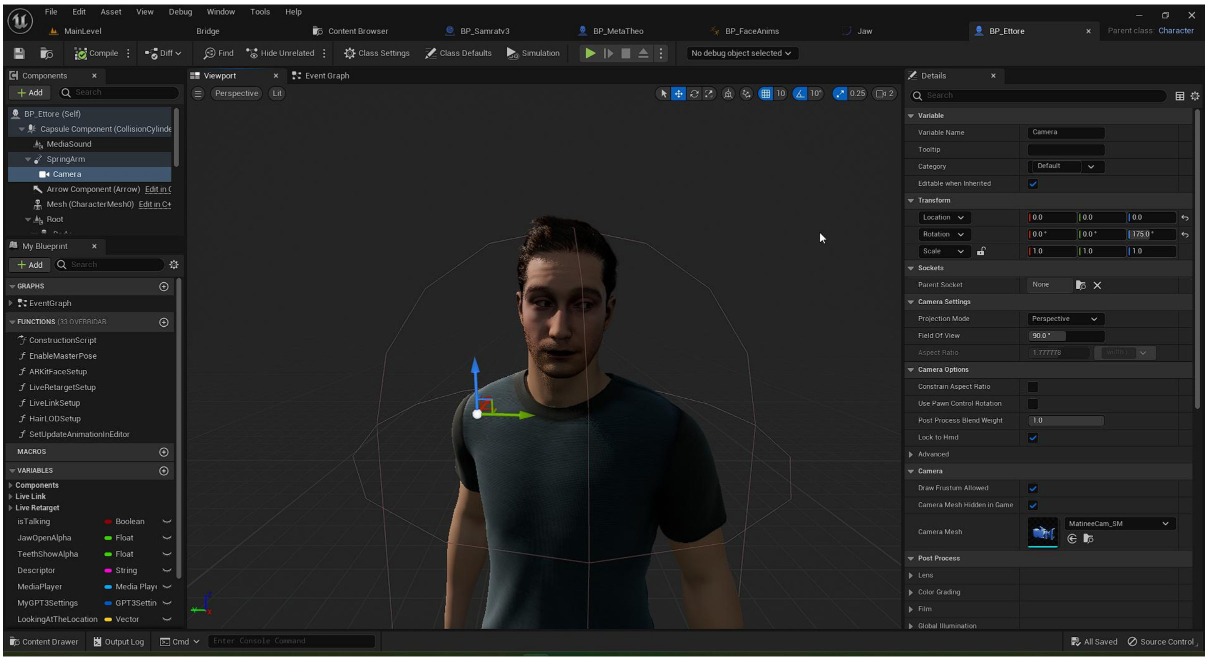
Design:

The design of the Visual AI Assistant “Softsasi AI” was carefully crafted to create an interactive and helpful experience for users. The assistant’s world was created with a futuristic and minimalist aesthetic, providing users with a sleek and elegant environment to interact with. The assistant in the game is a custom-designed 3D humanoid character, developed by the team member Md. Shakil Anower Samrat, which adds a unique element to the assistant’s world.

One of the key elements of the assistant’s design is the inclusion of a natural language interface that allows users to communicate with the assistant using voice or text. This feature is designed to make the assistant easy to use and understand, as well as add an element of personality and intelligence to the assistant.

The assistant’s user interface was designed to be simple and easy to navigate, with clear and concise instructions. The assistant’s UI design is minimalist and intuitive, allowing users to access the various features and options of the assistant with ease.user-friendly, making it easy for players to understand the controls and navigate the game.

Here are some glimpse of the layouts:



Components:

- BP_Ettore (Self)
  - Capsule Component (CollisionCylinde
    - MediaSound
    - SpringArm
      - Camera
    - Arrow Component (Arrow) Edit in C
    - Mesh (CharacterMesh0) Edit in C+
    - Root

My Blueprint:

GRAPHS

- EventGraph

FUNCTIONS (33 OVERRIDAB

- ConstructionScript
- EnableMasterPose
- ARKitFaceSetup
- LiveRetargetSetup
- LiveLinkSetup
- HairLODSetup
- SetUpdateAnimationInEditor

MACROS

VARIABLES

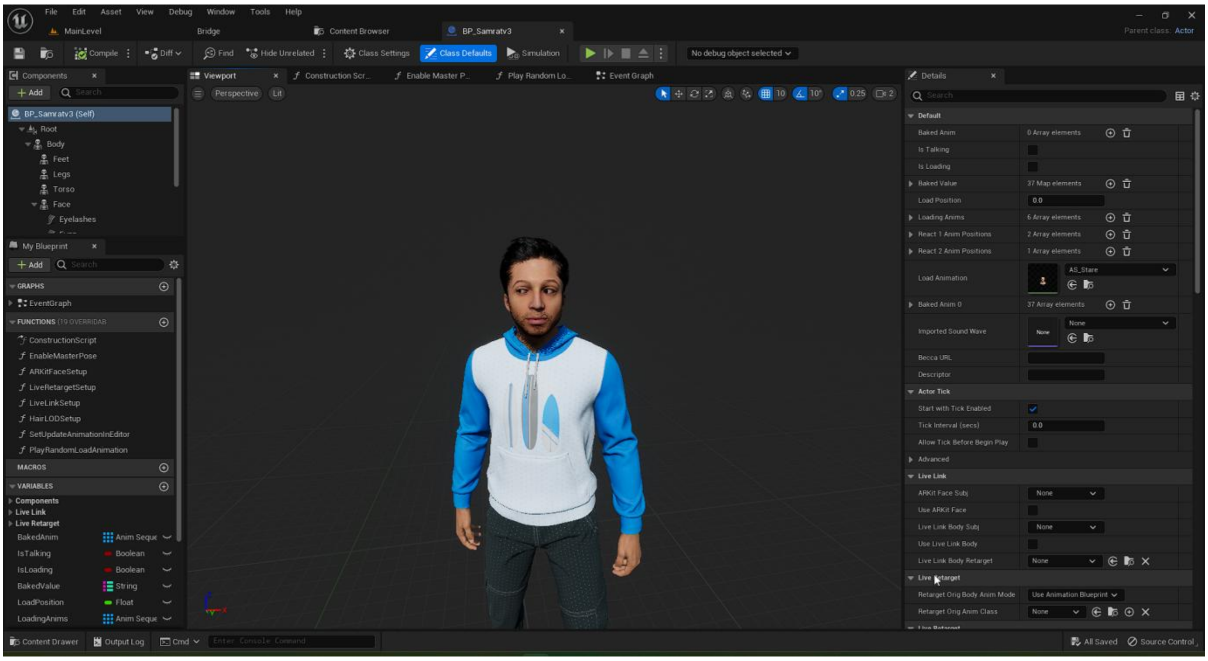
- Components
- Live Link
- Live Retarget

isTalking	Boolean
JawOpenAlpha	Float
TeethShowAlpha	Float
Descriptor	String
MediaPlayer	Media Playe
MyGPT3Settings	GPT3Settin
LookingAtTheLocation	Vector

Details:

Variable			
Variable Name	Camera		
Tooltip			
Category	Default		
Editable when Inherited	☑		
Transform			
Location	0.0	0.0	0.0
Rotation	0.0 °	0.0 °	175.0 °
Scale	1.0	1.0	1.0
Sockets			
Parent Socket	None		
Camera Settings			
Projection Mode	Perspective		
Field Of View	90.0 °		
Aspect Ratio	1.777778	width ı	
Camera Options			
Constrain Aspect Ratio	☐		
Use Pawn Control Rotation	☐		
Post Process Blend Weight	1.0		
Lock to Hmd	☑		
Advanced			
Camera			
Draw Frustum Allowed	☑		
Camera Mesh Hidden in Game	☑		
Camera Mesh	MatineeCam_SM		
Post Process			
Lens			
Color Grading			
Film			
Global Illumination			

Fig: Ettore AI Model Metahuman



Components:

- BP_Samratv3 (Self)
  - Root
    - Body
      - Feet
      - Legs
      - Torso
      - Face
        - Eyelashes

My Blueprint:

GRAPHS

- EventGraph

FUNCTIONS (19 OVERRIDAB

- ConstructionScript
- EnableMasterPose
- ARKitFaceSetup
- LiveRetargetSetup
- LiveLinkSetup
- HairLODSetup
- SetUpdateAnimationInEditor
- PlayRandomLoadAnimation

MACROS

VARIABLES

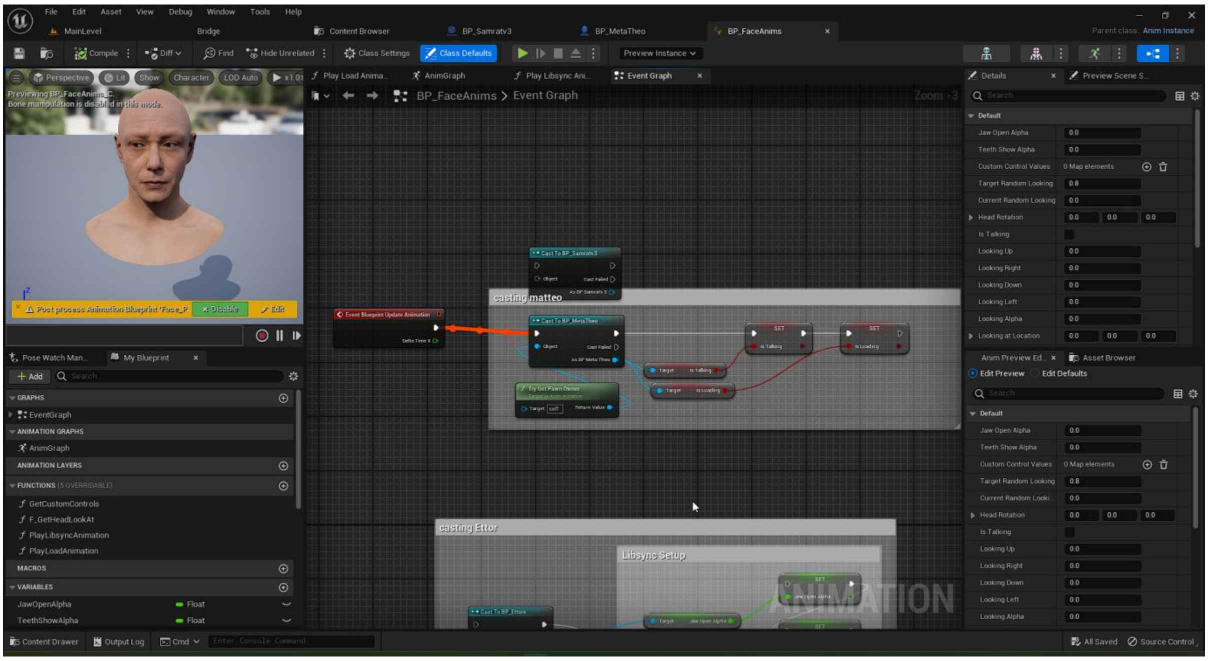
- Components
- Live Link
- Live Retarget

BakedAnim	Anim Seque
IsTalking	Boolean
IsLoading	Boolean
BakedValue	String
LoadPosition	Float
LoadingAnims	Anim Seque

Details:

Default	
Baked Anim	0 Array elements
Is Talking	☐
Is Loading	☐
Baked Value	37 Map elements
Load Position	0.0
Loading Anims	6 Array elements
React 1 Anim Positions	2 Array elements
React 2 Anim Positions	1 Array elements
Load Animation	AS_Stare
Baked Anim 0	37 Array elements
Imported Sound Wave	None
Becca URL	
Descriptor	
Actor Tick	
Start with Tick Enabled	☑
Tick Interval (secs)	0.0
Allow Tick Before Begin Play	☐
Advanced	
Live Link	
ARKit Face Subj	None
Use ARKit Face	☐
Live Link Body Subj	None
Use Live Link Body	☐
Live Link Body Retarget	None
Live Retarget	
Retarget Orig Body Anim Mode	Use Animation Blueprint
Retarget Orig Anim Class	None

Fig: Samrat AI Model Metahuman



Previewing BP_FaceAnims_C.
Bone manipulation is disabled in this mode.

Post process Animation Blueprint 'Face_P

Default

Jaw Open Alpha 0.0
Teeth Show Alpha 0.0
Custom Control Values 0 Map elements
Target Random Looking 0.8
Current Random Looking 0.0
Head Rotation 0.0 0.0 0.0
Is Talking
Looking Up 0.0
Looking Right 0.0
Looking Down 0.0
Looking Left 0.0
Looking Alpha 0.0
Looking at Location 0.0 0.0 0.0

Fig: Face Animation

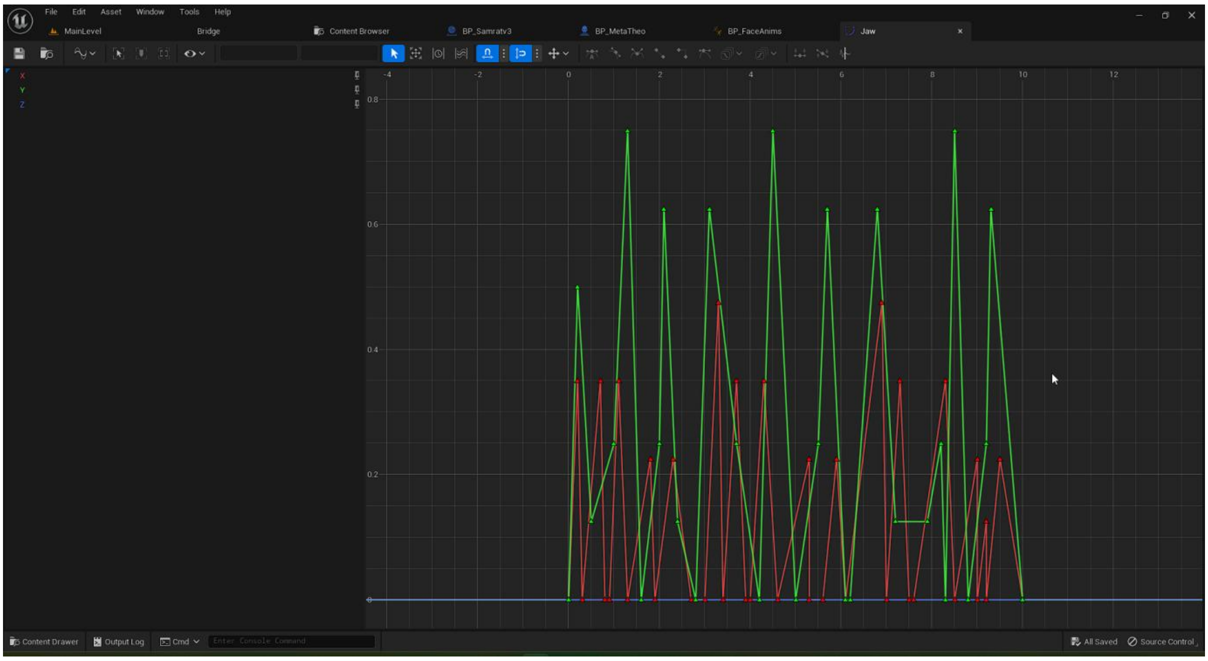


Fig: Jaw carves Animation

This way we have designed the game. Now let’s move on to the methodology part.

Methodology:

The team followed a systematic and iterative approach to develop the Visual AI Assistant. This included conceptualizing, developing, testing, and evaluating the assistant. The team used Unreal Engine 5 as the game engine, Blueprint as the visual scripting system, OpenAI api as the artificial intelligence platform, and C++ and JavaScript for programming.

Conceptualization: The team began by conceptualizing the assistant, discussing ideas for the assistant's functionality, graphics, and interaction. The team also researched similar assistants and identified what features they wanted to include in "Softsasi AI" and what features they wanted to avoid.

Development: Once the concept was finalized, the team began the development process. This included creating the assistant's world, designing and implementing the assistant's functionality, graphics, and interaction.

Here are some glimpse of Blueprint and C++ program:

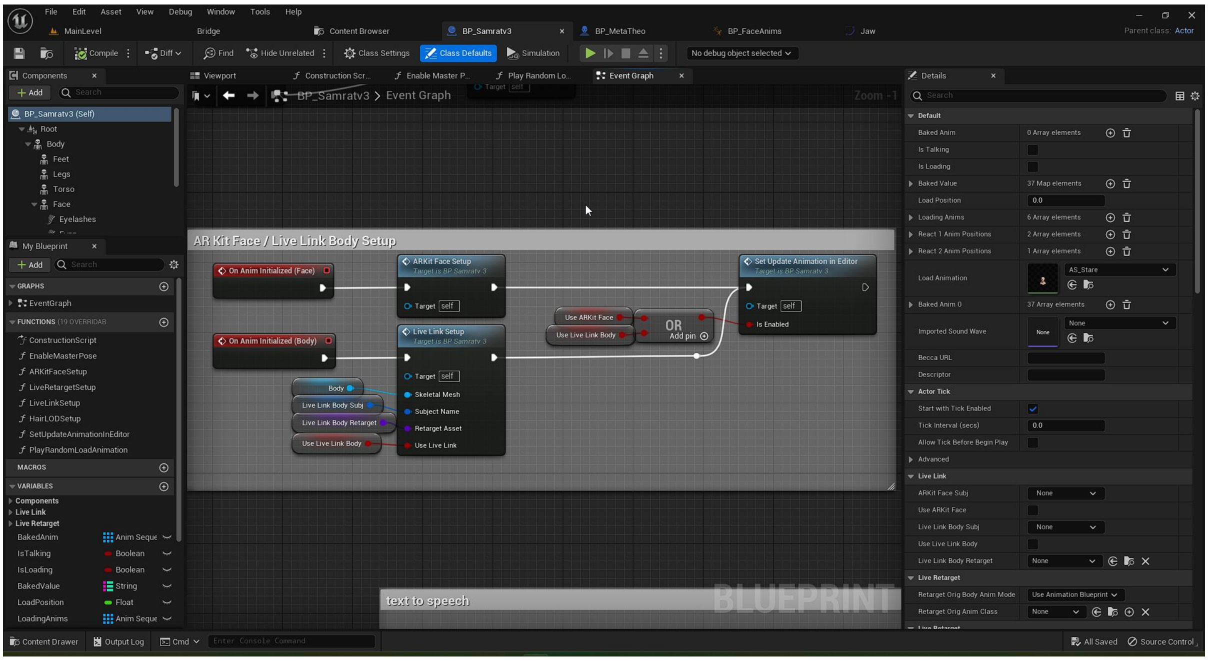
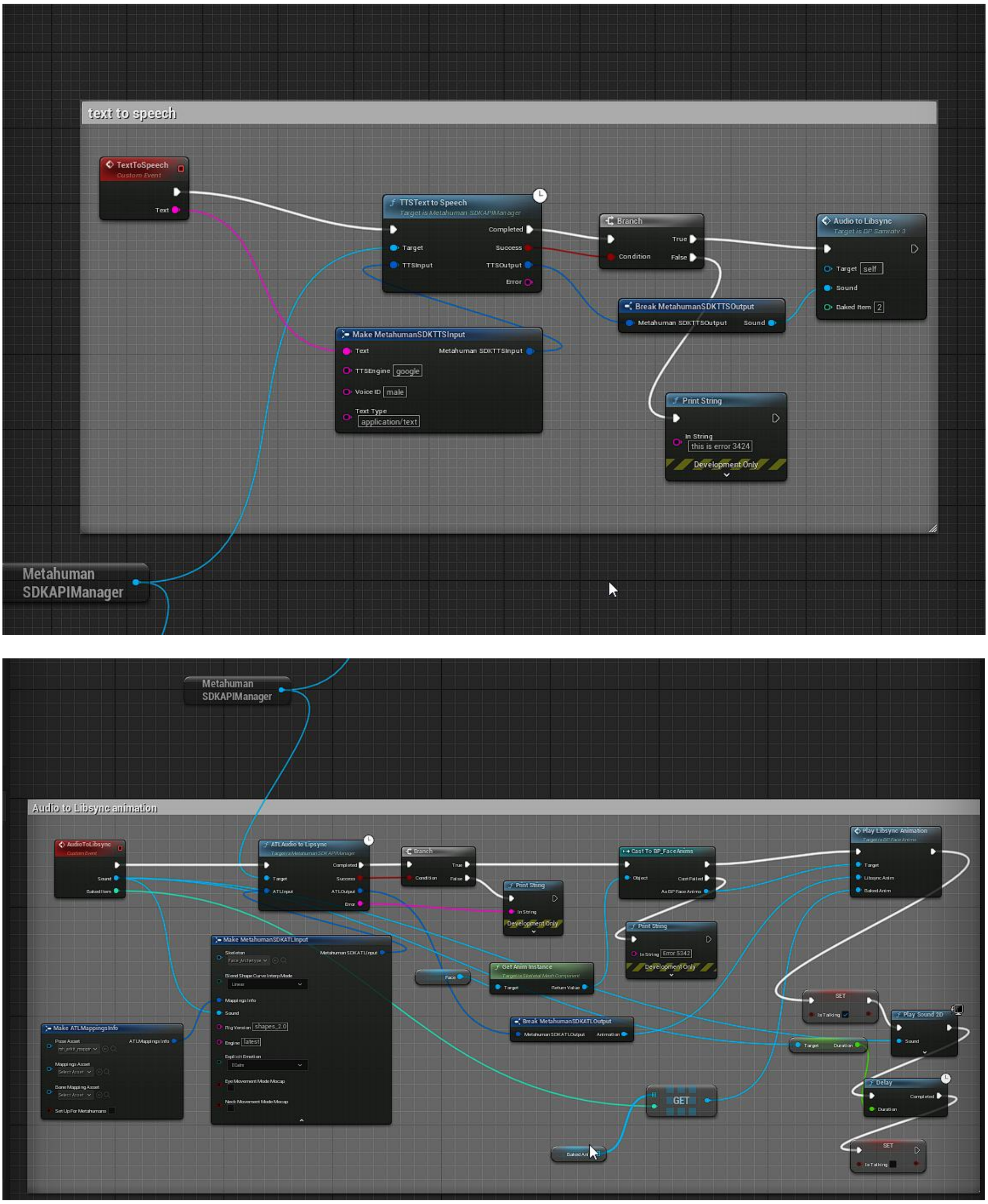


Fig: Live Link Body



Connecting Text to speech and Audio Libsync animation through Metahuman SDK

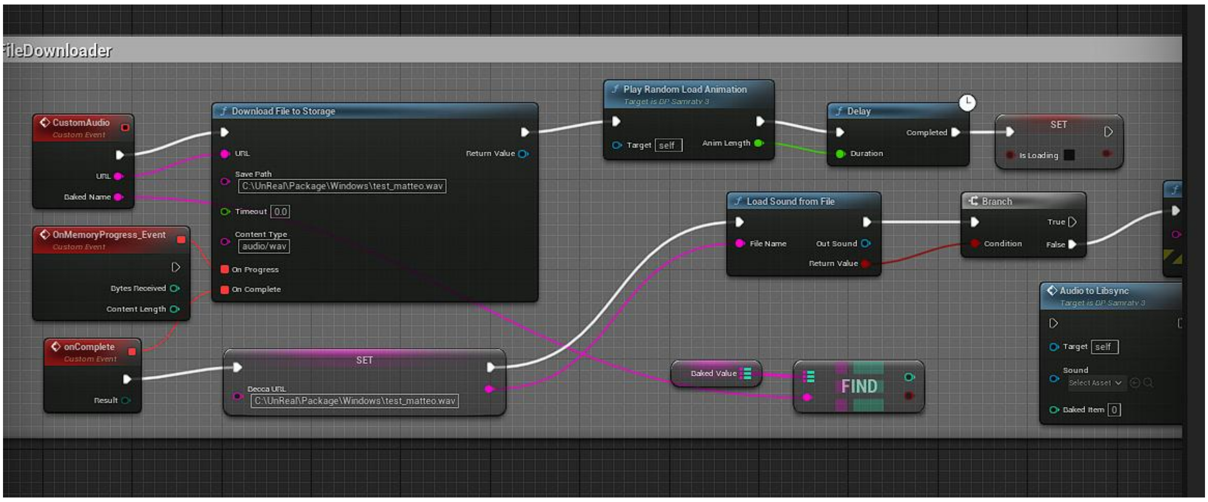
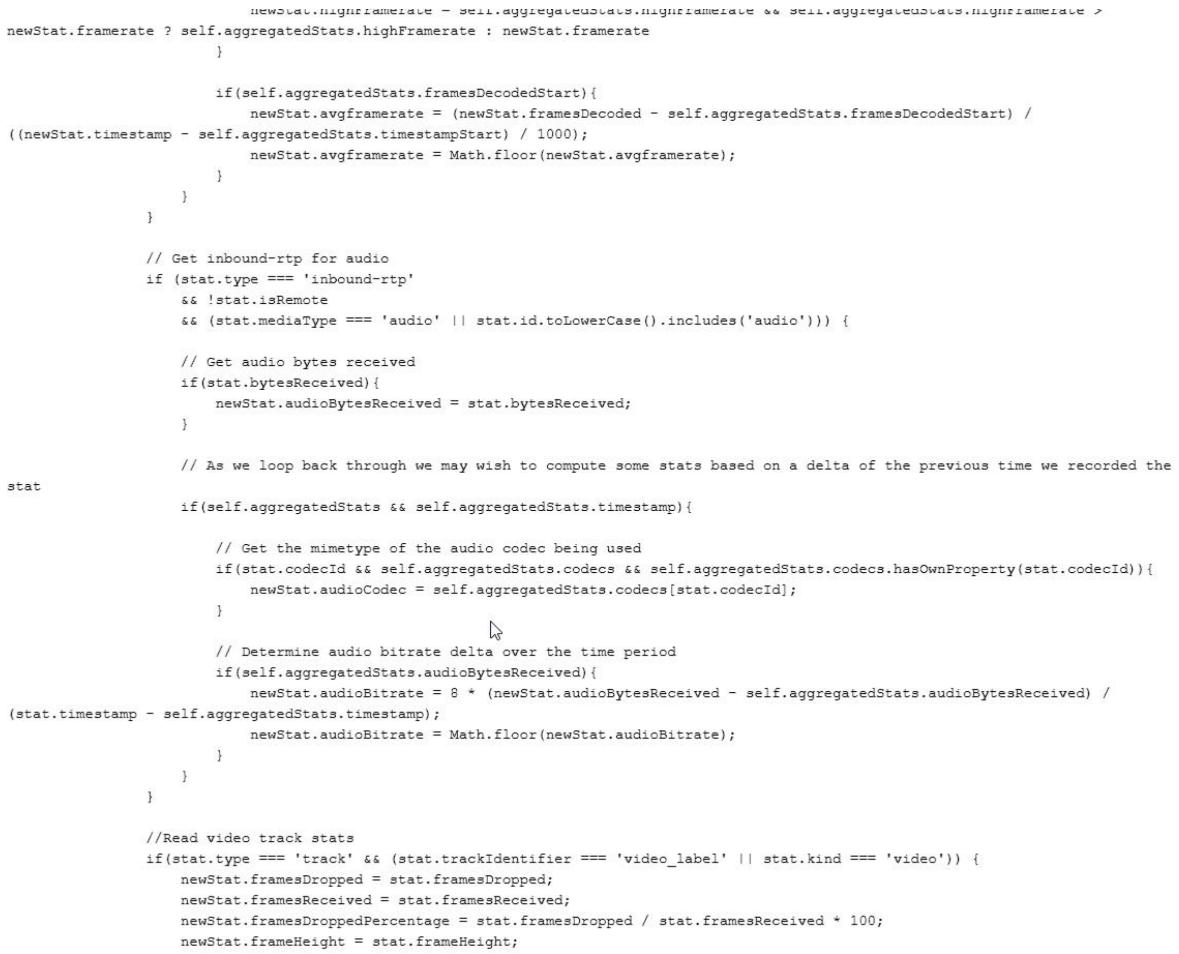
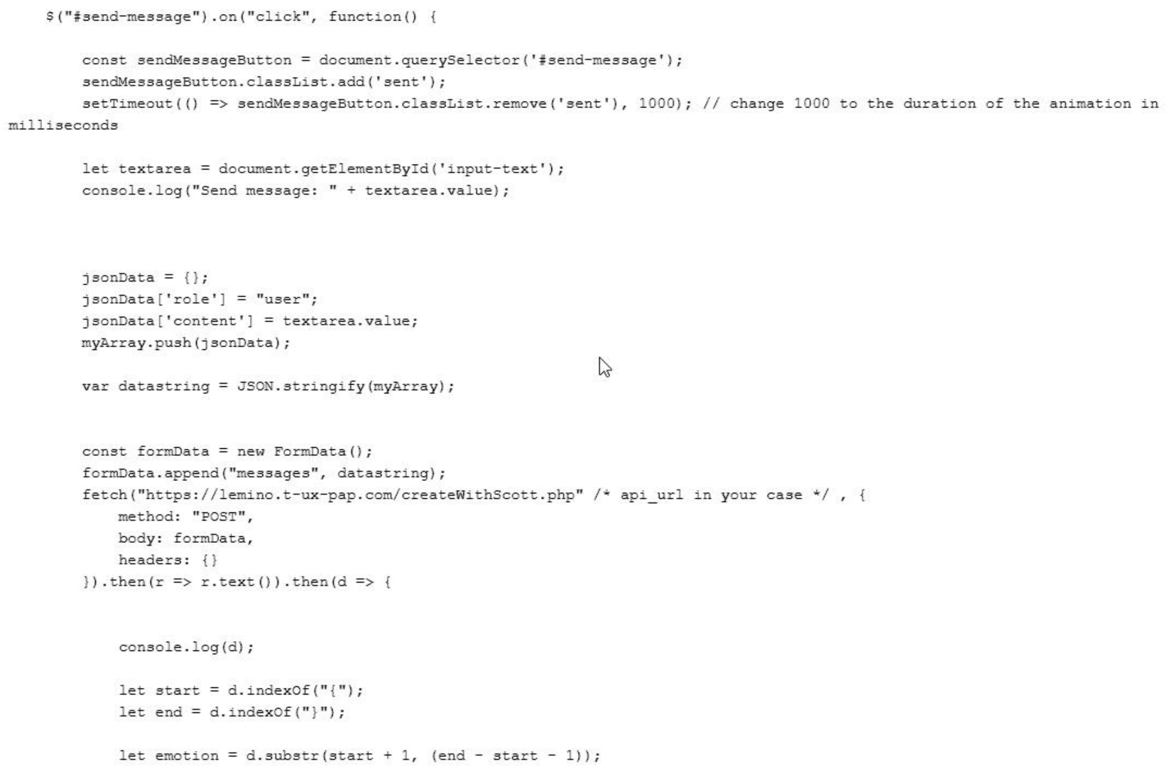


Fig: Text to speech Audio Downloder



```
                            newStat.highFramerate = self.aggregatedStats.highFramerate && self.aggregatedStats.highFramerate >
newStat.framerate ? self.aggregatedStats.highFramerate : newStat.framerate
                        }

                        if(self.aggregatedStats.framesDecodedStart){
                            newStat.avgframerate = (newStat.framesDecoded - self.aggregatedStats.framesDecodedStart) /
((newStat.timestamp - self.aggregatedStats.timestampStart) / 1000);
                            newStat.avgframerate = Math.floor(newStat.avgframerate);
                        }
                    }
                }

                // Get inbound-rtp for audio
                if (stat.type === 'inbound-rtp'
                    && !stat.isRemote
                    && (stat.mediaType === 'audio' || stat.id.toLowerCase().includes('audio'))) {

                    // Get audio bytes received
                    if(stat.bytesReceived){
                        newStat.audioBytesReceived = stat.bytesReceived;
                    }

                    // As we loop back through we may wish to compute some stats based on a delta of the previous time we recorded the
stat
                    if(self.aggregatedStats && self.aggregatedStats.timestamp){

                        // Get the mimetype of the audio codec being used
                        if(stat.codecId && self.aggregatedStats.codecs && self.aggregatedStats.codecs.hasOwnProperty(stat.codecId)){
                            newStat.audioCodec = self.aggregatedStats.codecs[stat.codecId];
                        }

                        // Determine audio bitrate delta over the time period
                        if(self.aggregatedStats.audioBytesReceived){
                            newStat.audioBitrate = 8 * (newStat.audioBytesReceived - self.aggregatedStats.audioBytesReceived) /
(stat.timestamp - self.aggregatedStats.timestamp);
                            newStat.audioBitrate = Math.floor(newStat.audioBitrate);
                        }
                    }
                }

                //Read video track stats
                if(stat.type === 'track' && (stat.trackIdentifier === 'video_label' || stat.kind === 'video')) {
                    newStat.framesDropped = stat.framesDropped;
                    newStat.framesReceived = stat.framesReceived;
                    newStat.framesDroppedPercentage = stat.framesDropped / stat.framesReceived * 100;
                    newStat.frameHeight = stat.frameHeight;
```

Fig: webRtc connection



```
    $("#send-message").on("click", function() {

        const sendMessageButton = document.querySelector('#send-message');
        sendMessageButton.classList.add('sent');
        setTimeout(() => sendMessageButton.classList.remove('sent'), 1000); // change 1000 to the duration of the animation in
milliseconds

        let textarea = document.getElementById('input-text');
        console.log("Send message: " + textarea.value);



        jsonData = {};
        jsonData['role'] = "user";
        jsonData['content'] = textarea.value;
        myArray.push(jsonData);

        var datastring = JSON.stringify(myArray);


        const formData = new FormData();
        formData.append("messages", datastring);
        fetch("https://lemino.t-ux-pap.com/createWithScott.php" /* api_url in your case */ , {
            method: "POST",
            body: formData,
            headers: {}
        }).then(r => r.text()).then(d => {


            console.log(d);

            let start = d.indexOf("{");
            let end = d.indexOf("}");

            let emotion = d.substr(start + 1, (end - start - 1));
```

Fig: User Message Sent to server

Testing: The team tested the assistant using various tools and methods such as unit testing, integration testing, system testing, usability testing, etc. The team also debugged and fixed any errors or bugs that they encountered during the development process.

Evaluation: The team evaluated the performance and quality of the assistant using various criteria such as functionality, usability, reliability, efficiency, maintainability, portability, etc. The team also collected feedback and suggestions from users and instructors using online surveys and interviews. The team also analyzed the results and findings from the testing and evaluation phase and made some improvements and modifications to the assistant based on them.

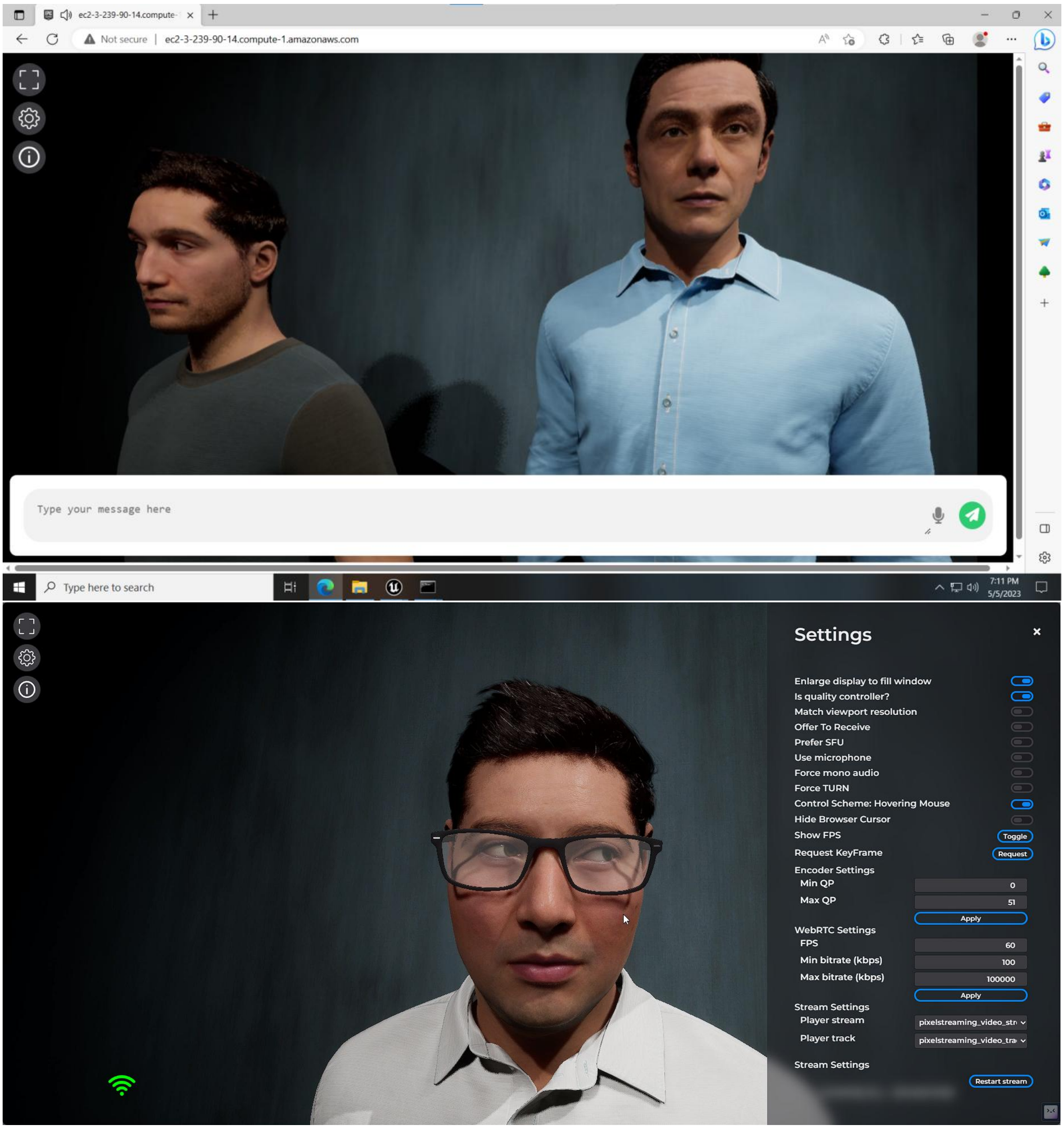
The development process was not linear or sequential but rather iterative and incremental. The team revisited and revised each phase multiple times based on the feedback, suggestions, results, findings, challenges, difficulties, etc. that they encountered during the development process. The team also communicated and collaborated with each other regularly using online platforms such as Discord, Zoom, and Slack to share their progress, problems, solutions, ideas, etc.

The development process was challenging but rewarding for the team. The team learned new skills and knowledge in programming, game development, artificial intelligence, project management, etc. The team also experienced teamwork, creativity, innovation, problem-solving, etc. The team also enjoyed working on the project and creating a Visual AI Assistant that can help users with various tasks and queries using natural language and high-quality graphics.

Result

The result of the project is a Visual AI Assistant “Softsasi AI” that can help users with various tasks and queries using natural language and high-quality graphics. The assistant features versatile functionality, intelligent responses, and engaging interactions. The assistant is accessible and convenient for users on any browser that supports Unreal Engine 5 technology.

The result of the project can be demonstrated by showing some examples of the assistant’s output and performance for different tasks and queries. The following are some screenshots of the assistant’s output and performance for web search and content creation tasks and queries.



Settings

Enlarge display to fill window ●
Is quality controller? ●
Match viewport resolution ○
Offer To Receive ○
Prefer SFU ○
Use microphone ○
Force mono audio ○
Force TURN ○
Control Scheme: Hovering Mouse ●
Hide Browser Cursor ○
Show FPS Toggle
Request KeyFrame Request

Encoder Settings
Min QP 0
Max QP 51
Apply

WebRTC Settings
FPS 60
Min bitrate (kbps) 100
Max bitrate (kbps) 100000
Apply

Stream Settings
Player stream pixelstreaming_video_str
Player track pixelstreaming_video_tra

Stream Settings
Restart stream

Fig: Running Screen

Discussion

The project team discussed the results and findings of the project, as well as the challenges and difficulties that they faced during the project development. The team also discussed the implications and limitations of the project, as well as the future work and recommendations for the project.

The team discussed the results and findings of the project, which showed that the Visual AI Assistant “Softsasi AI” was able to help users with various tasks and queries using natural language and high-quality graphics. The team also discussed the feedback and suggestions that they received from users and instructors, which indicated that the assistant was user-friendly, engaging, and versatile. The team also discussed the criteria and methods that they used to evaluate the performance and quality of the assistant, which showed that the assistant was functional, reliable, efficient, maintainable, and portable.

The team also discussed the challenges and difficulties that they faced during the project development, such as:

- Choosing the appropriate tools and technologies for the project, such as Unreal Engine 5, Blueprint, OpenAI api, C++, and JavaScript.
- Integrating the various components of the assistant, such as the world, the graphics, the functionality, and the interaction.
- Testing and debugging the assistant using various tools and methods, such as unit testing, integration testing, system testing, usability testing, etc.
- Collecting feedback and suggestions from users and instructors using online surveys and interviews.

- Communicating and collaborating with each other using online platforms such as Discord, Zoom, and Slack.

The team also discussed how they overcame these challenges and difficulties by using effective solutions and strategies, such as:

- Researching and comparing different tools and technologies for the project, such as Unreal Engine 5, Blueprint, OpenAI api, C++, and JavaScript.
- Using GitHub for version control and collaboration.
- Using various tools and methods to test and debug the assistant, such as unit testing, integration testing, system testing, usability testing, etc.
- Using online surveys and interviews to collect feedback and suggestions from users and instructors.
- Using online platforms such as Discord, Zoom, and Slack to communicate and collaborate with each other.

Limitations and further work of the Project

However, the project also had some limitations, such as:

- The assistant may not be able to handle all types of tasks and queries that users may have.
- The assistant may not be able to provide accurate or relevant responses for some tasks or queries that users may have.
- The assistant may not be able to support all languages that users may prefer to use.
- The assistant may not be compatible with all browsers or devices that users may want to use.

The team also discussed the future work and recommendations for the project, such as:

- Adding more features or functionalities to the assistant to make it more useful or entertaining for users.
- Improving or modifying some features or functionalities of the assistant to make it more accurate or relevant for users.
- Adding more languages or options to the assistant to make it more adaptable or customizable for users.
- Making sure that the assistant is compatible with all browsers or devices that users may want to use.

Future Scope of the Project

The project team has identified some areas for future work and improvement for the Visual AI Assistant “Softsasi AI”. The team has also made some recommendations for the project based on their past projects. After deeply research and analyzed, The project team has decide that the project will integrate with our last semester project Null Void. The future scope of the project includes:

Adding more features or functionalities to the assistant to make it more useful or entertaining for users. For example, the team could add features such as:

- Calendar: The assistant could help users manage their schedules, appointments, events, reminders, etc.
- Games: The assistant could play games with users, such as trivia, quizzes, puzzles, etc.
- Music: The assistant could play music for users, based on their preferences, mood, genre, etc.
- Jokes: The assistant could tell jokes to users, based on their humor, topic, language, etc.

Improving or modifying some features or functionalities of the assistant to make it more accurate or relevant for users. For example, the team could improve or modify features such as:

- Web search: The assistant could provide more filters, options, or sources for the web search results, such as date, location, category, etc.
- Graphic art: The assistant could provide more options or settings for the graphic art generation, such as style, color, size, etc.
- Code generation: The assistant could provide more languages or frameworks for the code generation, such as Python, Java, React, etc.
- Content creation: The assistant could provide more genres or formats for the content creation, such as horror, comedy, speech, letter, etc.

Adding more languages or options to the assistant to make it more adaptable or customizable for users. For example, the team could add languages or options such as:

- Hindi: The assistant could support Hindi as another language that users can use to communicate with the assistant.
- Voice: The assistant could allow users to choose the voice of the assistant from different options, such as male, female, accent, pitch, etc.
- Name: The assistant could allow users to name the assistant according to their preference.
- Theme: The assistant could allow users to change the theme of the assistant’s world and graphics from different options, such as dark mode, light mode, etc.

Making sure that the assistant is compatible with all browsers or devices that users may want to use. For example, the team could ensure that the assistant works well on browsers or devices such as:

- Firefox: The team could test and optimize the assistant for Firefox browser.
- Safari: The team could test and optimize the assistant for Safari browser.
- Android: The team could test and optimize the assistant for Android devices.
- iOS: The team could test and optimize the assistant for iOS devices.

These are some of the possible areas for future work and improvement for the project. The team believes that by implementing these changes and enhancements, they can make the Visual AI Assistant “Softsasi AI” more user-friendly, engaging and versatile.

Conclusion

The project team has successfully developed a Visual AI Assistant “Softsasi AI” that can help users with various tasks and queries using natural language and high-quality graphics. The project team has used Unreal Engine 5, Blueprint, OpenAI api, C++, and JavaScript as the tools and technologies for the project. The project team has followed a systematic and iterative approach to design, implement, test, and evaluate the project. The project team has achieved their objectives and requirements for the project. The project team has also received positive feedback and suggestions from users and instructors. The project team has also learned new skills and knowledge from working on the project. The project team has also identified some areas for future work and improvement for the project. The project team is proud of their work and hopes that the Visual AI Assistant “Softsasi AI” will be useful and entertaining for users.