Integration of Cross-Lingual Voice-Enabled NLP with Gaming Platform

Enrollment No. (s) - 19103034, 19103044, 19104053
Name of Student (s) - Roshni Singh, Tanishk Gupta, Ujjwal Sachdeva
Name of supervisor - Dr. Ankit Vidyarthi

May - 2023

**Submitted in partial fulfillment of the Degree of
Bachelor of Technology
in
Computer Science Engineering / Information Technology**

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)`

TABLE OF CONTENTS

(II)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: JIIT, Noida

Signature:

Date: 09/05/23

Name: Ujjwal Sachdeva, Tanishk Gupta, Roshni Singh

Enrollment No.: 19104053, 19103044, 19103034

2

CERTIFICATE

This is to certify that the work titled " **Integration of Cross-Lingual Voice-Enabled NLP with gaming platform** " submitted by " **Roshni Singh, Tanishk Gupta, and Ujjwal Sachdeva** " in partial fulfillment for the award of the degree of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor:

Name of Supervisor: Dr. Ankit Vidyarthi

Designation: Assistant Professor(Snr. Grade)
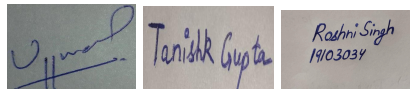
Date: 09/05/23

(IV)

ACKNOWLEDGEMENT

We convey our heartfelt thanks to Dr. Ankit Vidyarthi, our mentor for making every session interactive and interesting. We also thank him for being patient and guiding us all through the project. We thank our institution for providing us with an opportunity to develop this major project, which will play a significant part in our career and any future interviews we are to face.

Signature of the Students:

Name of Students: Ujjwal Sachdeva, Tanishk Gupta, Roshni Singh

Enrollment Number: 19104053, 19103044, 19103034
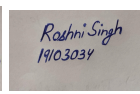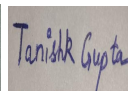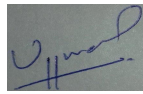
Date: 09/05/23

(V)
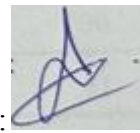
<u>SUMMARY</u>

This project provides a virtual playground where players can connect and interact with each other without being held back by language barriers. With its advanced language translation feature, the project allows players to communicate in real-time, translating messages between different languages instantly. This not only enhances the gaming experience but also helps build a sense of community and fosters global friendships. Additionally, the project's use of machine learning ensures that translations are accurate and reliable, making communication smoother and more efficient. Overall, the Unity project offers a unique solution to the challenge of language barriers in online gaming and opens up new avenues for players to connect and engage with each other.

Signature of Students: Signature of Supervisor:

Name: Ujjwal Sachdeva, Tanishk Gupta, Roshni Singh   Name: Dr. Ankit Vidyarthi

Date: 09/05/23   Date: 09/05/23

(VI)

LIST OF FIGURES

(VII)

LIST OF TABLES

(VIII)

LIST OF ACRONYMS

1. CLIR - Cross-Lingual Information Retrieval
2. SMT - Statistical Machine Translation
3. HWTD - Hybrid Word Translation Disambiguation
4. SMVS - Semantic Morphological Variant Selection
5. OOV - Out Of Vocabulary
6. MMT - Multi-task Multi-stage Transitional
7. NCT - Neural Chat Translation
8. NMT - Neural Machine Translation
9. IDE - Integrated Development Environment
10. SDK - Software Development Kit
11. HTTPS - Hypertext Transfer Protocol
12. DNS - Domain Name Server
13. API - Application Programming Interface
14. UI - User Interface

Chapter 1

Introduction

1.1   General Introduction

Connecting with individuals from around the world is simpler than ever in the global world today. However, when it comes to contact and communication between speakers of other languages, language boundaries can still be a major obstacle. Language hurdles can be a major barrier in the context of a virtual game when participants from different regions of the world gather to play and interact.

By offering a virtual playground where players may gather to play a virtual game and interact and speak with each other using the app's language translation feature, the Unity app was created to address this issue. By leveraging the power of machine learning and natural language processing, the project aims to enable seamless communication between players who communicate in different languages.

1.2   Problem Statement

The problem being addressed is the language barrier that exists between players who communicate in different languages in a virtual game. This language barrier can limit the ability of players to communicate effectively, which can lead to frustration, misunderstandings, and a lack of engagement in the game.

Moreover, existing translation tools and services may not be well-suited for the context of a virtual game, where real-time communication and interaction are critical. The latency and accuracy issues associated with these tools can further compound the language barrier problem, making it even more difficult for players to communicate and interact with each other effectively.

1.3 Significance of the Problem

The problem being addressed in this statement is significant because it speaks to the potential negative impact that language barriers can have on the gaming experience. Effective communication is critical for successful gameplay and collaboration, and language barriers can hinder players' ability to communicate effectively, leading to frustration, misunderstandings, and disengagement from the game.

Furthermore, the statement highlights the limitations of existing translation tools and services in the context of a virtual game. These tools may not be well-suited to handle the real-time communication and interaction required in gaming, leading to latency and accuracy issues that can further exacerbate the language barrier problem.

Addressing the problem is important because it can improve the gaming experience for players by enhancing their ability to communicate effectively and collaborate with each other. Additionally, the development of a solution that is specifically tailored to the unique context of virtual gaming could have implications beyond the gaming world, potentially improving communication and collaboration in other contexts where language barriers exist.

1.4 Empirical Study

There have been several studies in recent years that have explored the impact of language barriers on communication and collaboration in virtual gaming contexts.

A study conducted by researchers at the University of Illinois at Urbana-Champaign and the University of California, Irvine found that language barriers can significantly impact players' ability to communicate and collaborate in virtual game environments. The study involved observing players in a virtual game and analyzing their communication patterns. The researchers found that players who spoke the same language tended to communicate more frequently and effectively, while those who spoke different languages had more difficulty communicating and were less likely to collaborate with each other.

Another study conducted by researchers at the University of British Columbia and Simon Fraser University in Canada explored the use of translation tools in virtual gaming contexts. The study found that existing translation tools were not well-suited for the context of virtual gaming, as they often resulted in latency and accuracy issues that further hindered communication and collaboration between players who spoke different languages.

These studies highlight the importance of addressing the problem of language barriers in virtual gaming contexts and the need for tailored solutions that can enable effective communication and collaboration between players who speak different languages. The app's language translation feature, which leverages machine learning and natural language processing, is one such solution that has the potential to overcome the limitations of existing translation tools and improve communication and collaboration in virtual gaming environments.

1.5   Brief Description of the Solution Approach

The proposed solution involves building a Unity app that utilizes machine learning and natural language processing to provide real-time chat translation between players who communicate in different languages during gameplay. The project will offer a virtual playground where players can come together, play a game, and chat with each other using their native language.

To achieve this, the chat translation feature of the application will utilize machine learning algorithms to analyze the text input of each player and translate it into the language of the other player in real time. Additionally, the translation accuracy will be continually improved by training the algorithms with user feedback and data.

To ensure effective communication and interaction between players, the application will also include in-game tools and features such as voice chat, text chat, game-specific commands, and other tools that enhance the gameplay experience and enable effective communication and collaboration.

1.6   Comparison of existing approaches to the problem framed

Existing approaches to address the language barrier problem in virtual games include:

Using pre-built language packs: Some games offer pre-built language packs that players can install to allow them to communicate with other players in different languages. However, these packs have limited translations, and players may still encounter communication problems, especially with more complex interactions.

Using third-party translation tools: Players can use third-party translation tools like Google Translate to communicate with players who speak different languages. However, these tools can be inaccurate and may not provide real-time translations, leading to delayed and frustrating communication.

Learning new languages: Players can learn new languages to improve their communication skills in virtual games. While this may be effective, it is time-consuming, and not everyone may have the resources or interest to learn a new language.

Compared to these existing approaches, the proposed solution offers real-time chat translation using machine learning algorithms that will improve accuracy with user feedback and data. The solution also provides in-game tools and features that enhance communication and collaboration between players, offering a more comprehensive solution to the language barrier problem in virtual games.

Chapter 2

Literature Survey

2.1   Summary of papers studied

Exploring Web-based Translation Resources Applied to Hindi-English Cross-Lingual Information Retrieval [1] (ACM Transactions on Asian and Low-Resource Language Information Processing, 2022)

It analyzes the problems with cross-lingual information retrieval (CLIR) on the internet and suggests several web sources and methods of online dictionary-based translation to solve them, particularly for Hindi-English CLIR. The suggested methods provide roughly comparable mean average precision to the traditional Statistical Machine Translation (SMT) technique, as is discovered when they are compared to SMT.

Three modules are included in the suggested Wikipedia-based translation technique to deal with the problems of incorrect inter-wiki linkages, partially matched terms, and confusing articles. To find translations, internet dictionaries, ConceptNet, Hindi WordNet & Indo WordNet, and WordNet are used. Translations are clarified using WordNet path similarity.

Semantic morphological variant selection and translation disambiguation for cross-lingual information retrieval [2] (Multimedia Tools and Applications, 2021)

It talks about a study that addresses two major problems with cross-lingual information retrieval (CLIR) for the Hindi language: the problem of translation mis-mapping brought on by the morphological richness of the language and the problem of word translation disambiguation brought on by multiple translations of a word in a dictionary.

To overcome these difficulties, the study suggests two techniques: hybrid word translation disambiguation (HWTD) and semantic morphological variant selection

(SMVS). On the FIRE ad-hoc datasets, the proposed algorithms are assessed, and it is discovered that they outperform the baseline Statistical Machine Translation in terms of assessment measures at the word level.

Context-based Translation for the Out of Vocabulary Words Applied to Hindi-English Cross-Lingual Information Retrieval [3] (IETE Technical Review 39, no. 2, 2022)

This paper proposes a context-based translation algorithm to address the Out Of Vocabulary (OOV) words issue in Cross-Lingual Information Retrieval (CLIR) for Indian languages. The algorithm utilizes two large raw corpora (in source and target language) and a small bi-lingual parallel corpus to translate OOV words.

The proposed algorithm achieves better recall and mean average precision in comparison to the baseline Statistical Machine Translation (SMT) for Hindi-English CLIR. The proposed algorithm reduces the number of OOV words more effectively.

A Multi-Task Multi-Stage Transitional Training Framework for Neural Chat Translation [4] (IEEE, 2022)

This paper proposes a multi-task multi-stage transitional (MMT) training framework for neural chat translation (NCT), which aims to translate cross-lingual chats between speakers of different languages. Existing context-aware NMT models face challenges due to limited resources of annotated bilingual dialogues, neglect of conversational properties, and training discrepancy between different stages.

The MMT framework uses an NCT model trained on a bilingual chat translation dataset and additional monolingual dialogues with two auxiliary tasks - utterance discrimination and speaker discrimination - to introduce dialogue coherence and speaker characteristic into the model.

The training process has three stages, with intermediate training using additional monolingual dialogues, serving as a phase to alleviate training discrepancy. The NCT model is trained using a gradual transition strategy, moving from monolingual to

bilingual dialogues, to make the stage transition smoother. The proposed framework is demonstrated to be effective and superior through extensive experiments on two language pairs.

3D Game Development with Unity A Case Study: A First-Person Shooter (FPS) Game [5] (2014)

The project focuses on developing a 3D game application using the Unity 4 game engine for Windows OS. The report is aimed at beginners with basic knowledge of the Unity 3D game engine. The game is a first-person shooter game, and the report discusses the theories of game design, introduces and demonstrates the Unity game engine, and discusses the project's implementation, future development, and feasibility of the market. The game is not developed with ways of generating revenue, and it's more suitable for recreating and teaching purposes.

Development of Smart Game Based on Multi-Platform Game Engine [6] (International Journal of Multimedia and Ubiquitous Engineering, 2016)

The study focuses on designing and developing a smart game using the Unity3D game engine, which supports various functions such as shader, physics engine, network, terrain manipulation, audio, video, and animation. The study aims to plan, design, and develop a smart game based on the multi-platform game engine to make the game development process more efficient.

2.2   Integrated Summary of the literature studied

It talks about a study that addresses two major problems with cross-lingual information retrieval (CLIR) for the Hindi language: the problem of translation mis-mapping brought on by the morphological richness of the language and the problem of word translation disambiguation brought on by multiple translations of a word in a dictionary.

To overcome these difficulties, the study suggests two techniques: hybrid word translation disambiguation (HWTD) and semantic morphological variant selection (SMVS). The proposed algorithms are assessed, and it is discovered that they outperform the baseline Statistical Machine Translation in terms of assessment measures at the word level.

Another paper aims to develop a 3D game application using the Unity 4 game engine for Windows OS. The report targets beginners with basic knowledge of the Unity 3D game engine and discusses the theories of game design, the Unity game engine, project implementation, future development, and market feasibility. The game is a first-person shooter game and is more suitable for recreating and teaching purposes rather than generating revenue.

Overall, the papers highlight the limitations of existing CLIR techniques for Indian languages, such as the non-absoluteness of the parallel corpus, the morphological richness of the Hindi language, and the word translation disambiguation issue. The proposed techniques improve evaluation measures such as Recall and Mean Average Precision (MAP) by effectively addressing these limitations.

The papers show that utilizing various web resources, such as Wikipedia and WordNet, can improve translation accuracy. Additionally, addressing the OOV words using context-based translation algorithms can significantly improve the effectiveness of CLIR techniques for Indian languages.

Research papers	Dataset used	Findings
Sharma Et al. [1]	FIRE 2010 and 2011	● The paper proposes using web-based translation techniques such as Wikipedia, HWN & IWN, CON, and online dictionaries to address this issue. ● The Wikipedia-based technique is found to be effective in translating user queries using exact or partial matches.
Sharma Et al. [2]	FIRE 2010 and 2011	● The paper proposes a new technique (SMVS) for cross-lingual information retrieval that outperforms other techniques and a new word-level disambiguation technique using association scores.
Sharma Et al. [3]	FIRE 2010 and 2011	● The paper proposes a context-based translation algorithm to translate OOV words, addressing both morphological variants and missing words. ● The algorithm uses SC and SA components and is varied based on four models, with CBOW_Vec performing the best on the FIRE ad-hoc dataset.
Zhou Et al. [4]	Bilingual Chat Translation Dataset	● The paper proposes a multi-task multi-stage transitional (MMT) framework for neural chat translation (NCT) to address challenges faced by existing context-aware NMT models. ● The proposed framework is demonstrated to be effective and superior through experiments on two language pairs.
Peng Et al. [5]	Manual	● The project creates a 3D first-person shooter game using Unity 4 for Windows OS, with a report covering game design theories, Unity engine, project implementation, and market feasibility. ● The game is for recreation and education, not revenue.
Hwan Et al. [6]	Manual	● The study aims to design and develop a smart game using Unity3D game engine with various functions, including shader, physics engine, network, terrain manipulation, audio, video, and animation, to make the game development process more efficient.

Table 1: Comparison of papers

17

Chapter 3

Requirement Analysis and Solution Approach

3.1 Overall Description of the Project

The project aims to provide an application that is designed to create a more inclusive and connected gaming experience for players around the world. The language translation feature of the project, powered by machine learning algorithms and natural language processing, allows players to communicate in real time without needing pre-built language packs or third-party translation tools, which can be inaccurate and frustrating to use. By enabling players to converse in their native languages, the project helps build a sense of community and encourages global friendships.

The project's advanced technology ensures accurate and reliable translations, creating a smoother and more efficient communication experience. The app offers a unique solution to the problem of language barriers in online gaming, providing a virtual playground where players can connect and engage with each other on a deeper level.

3.2 Development Requirements

Requirements for the project can be as follows:

Functional requirements:
- Application must support real-time communication between players in different languages.
- Application must translate messages accurately and reliably.
- Application must support multiple languages.
- Application must provide an intuitive and user-friendly interface.
- Application must be compatible with different devices and operating systems.

Non-functional requirements:

- Application must have high performance and low latency.
- Application must be secure, with user data and privacy protected.
- Application must have a scalable architecture to accommodate increasing numbers of users.
- Application must be easily maintainable and upgradable.

Technical requirements:

- Application must use machine learning algorithms and natural language processing for translation.
- Application must support C# and other object-oriented scripting languages.
- Application must have access to a large dataset of multilingual texts for training the machine learning models.
- Application must be able to handle different accents, dialects, and cultural nuances of different languages.

User requirements:

- Application must be easy to use and intuitive, even for non-technical users.
- Application must provide clear instructions on how to use the translation feature.
- Application must have a microphone enabled.

About the Game

The game integrated is a First Person Perspective shooting game where players across the world can create a lobby and other players can either join already created lobbies or create another. After joining the game, players get spawned at different locations in a specified area. Each player gets equipped with a gun and health. To win the game, players need to aim and shoot at other players. Players can move across the area with the 'w', 'a', 's', and 'd' keys, and shoot with the left mouse button and jump with the spacebar. If a player dies in the game, they get respawned at a random position in the gaming area and can continue playing. Each player can view the scoreboard which includes the player's kills and deaths by pressing the tab button.
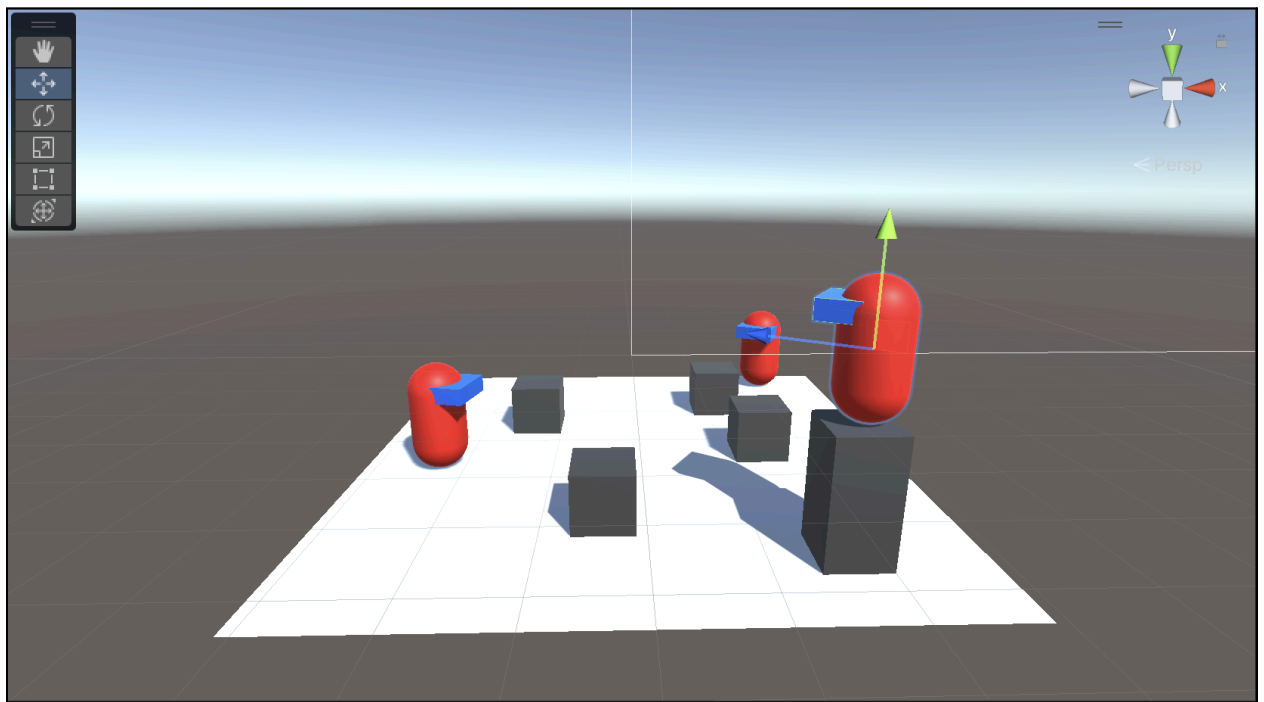

Figure 1: The Game

About Unity

Unity is a cross-platform game engine that is widely used by game developers to create 2D and 3D games for various platforms, including mobile devices, PCs, consoles, and virtual reality devices. It provides an integrated development environment (IDE) that includes a visual editor, scripting tools, a physics engine, an audio engine, and many other features that make game development easier and more efficient.

It supports multiple programming languages, including C#, JavaScript, and Boo, and it also supports popular game development tools such as Visual Studio and MonoDevelop. With Unity, developers can create games with stunning graphics, realistic physics, and immersive audio, and deploy them to a variety of platforms with ease.

It has a large and active community of developers and users, who provide support, resources, and plugins that can extend the engine's capabilities. Unity is widely used in the game development industry, as well as in other industries such as architecture, engineering, and film-making.
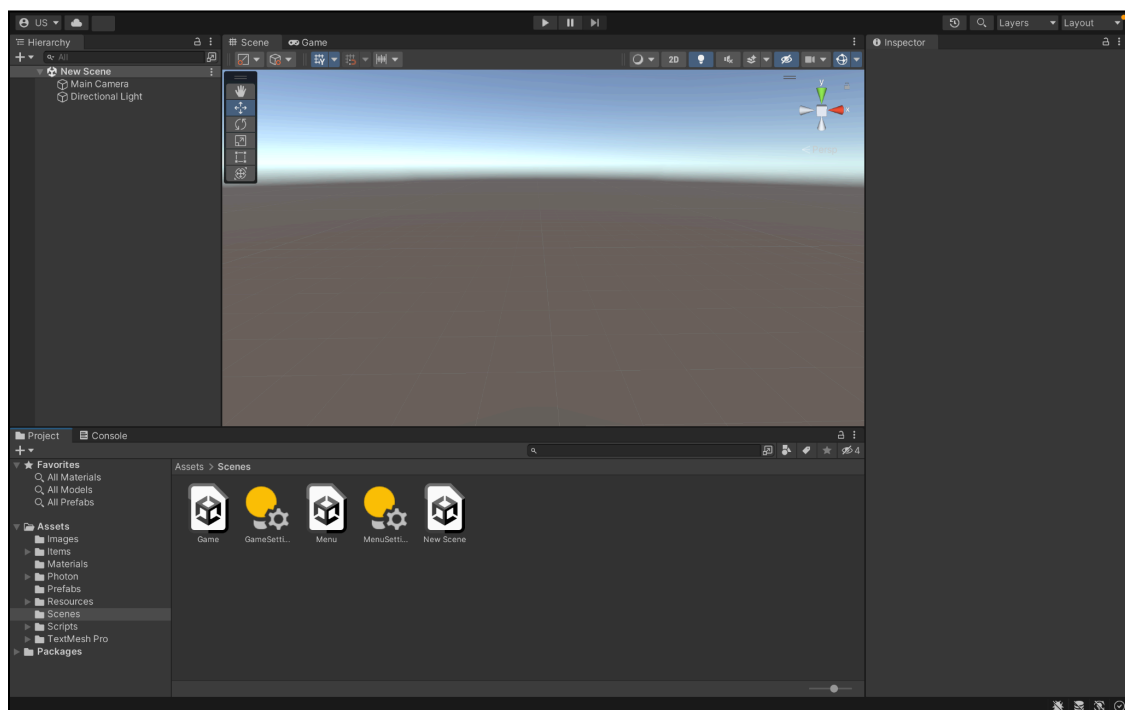
Figure 2: A Unity scene

About Photon Engine

Photon Engine is a real-time multiplayer game development engine and cloud-based service that provides a high-performance server architecture for online games. It was developed by Exit Games and is used by game developers to create multiplayer games for various platforms, including mobile devices, PCs, consoles, and virtual reality devices.

It provides a scalable and reliable infrastructure for multiplayer games, including features such as matchmaking, in-game communication, player authentication, and synchronization of game state across multiple devices. It supports multiple programming languages, including C#, JavaScript, and provides a wide range of client SDKs.

Photon Engine is a powerful and versatile tool for creating multiplayer games that can handle high player counts, provide low latency gameplay, and ensure a seamless and enjoyable gaming experience.

About Netlify (Deployment)

Netlify is a cloud-based platform for web developers that provides a range of tools and services to simplify the process of building and deploying websites and web applications. It offers features such as continuous deployment, serverless functions, HTTPS-enabled custom domains, and integrated form handling, among others.

With Netlify, developers can easily connect their code repository from platforms like GitHub, GitLab, or Bitbucket, and set up continuous deployment, which automatically builds and deploys changes to the live site every time they push code to the repository. Netlify also provides serverless functions that enable developers to run code on-demand without managing server infrastructure.

It also offers a range of other features, such as integrated form handling, asset optimization, A/B testing, and performance monitoring, among others. It also provides a user-friendly web interface that makes it easy to manage website settings, configure DNS records, and monitor site performance.

Overall, Netlify is a powerful and flexible platform that simplifies the process of building and deploying websites and web applications, enabling developers to focus on their code and deliver better experiences to their users.

Natural Language Processing

Computer science and artificial intelligence's Natural Language Processing (NLP) branch focuses on how computers and human language interact. NLP works with a variety of activities, including sentiment analysis, speech recognition, machine translation, and text processing. The ultimate objective of NLP is to make it possible for robots to comprehend and produce human language, which is a dynamic and complicated system.

NLP's reliance on dictionaries and other lexical resources to comprehend language is one of its drawbacks, though. Dictionaries have a limited number of words and their definitions, therefore NLP systems cannot recognize any term that is not in a dictionary. This restriction can be particularly troublesome in languages like Hindi, where there are several terms that are often used in ordinary conversation but are not included in dictionaries or shabdkosh (a Hindi dictionary).

Many Hindi terms, for instance, are derived from local dialects but have not yet received official recognition in dictionaries or shabdkosh. Because of this, it may be challenging for NLP systems to correctly analyze and comprehend natural language content in Hindi. In addition, Hindi has intricate grammar and syntax, which makes NLP jobs more challenging.
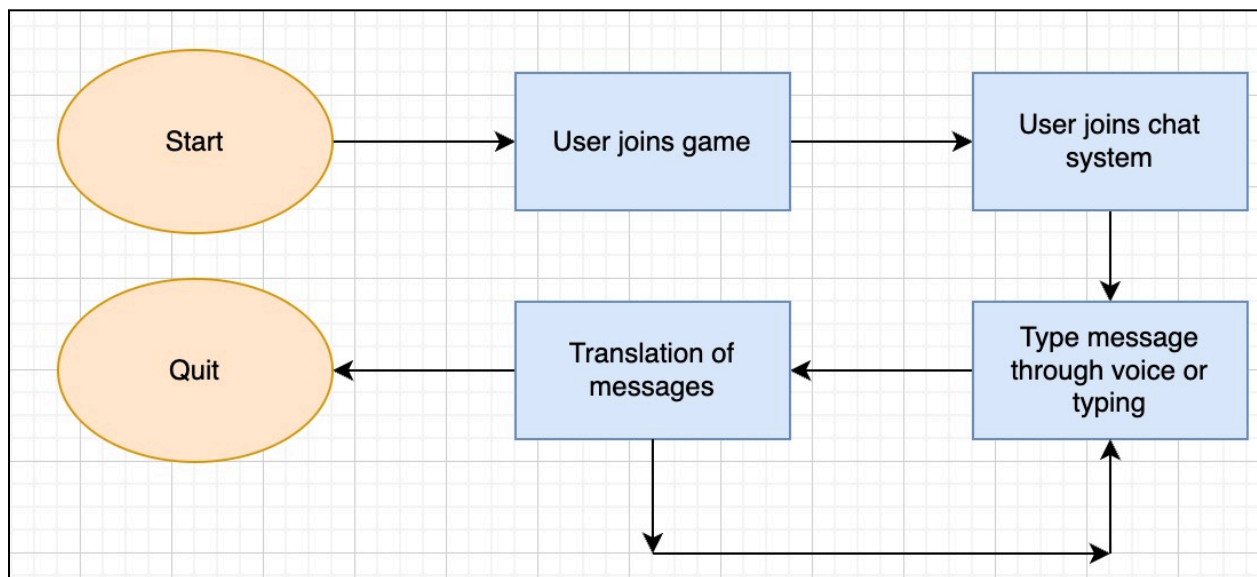
3.3   Solution Approach

Innovative design and state-of-the-art technology were combined as part of the Cross-Lingual Voice-Enabled Information Processing in the Game Environment project's solution strategy. Multiplayer functionality, voice-enabled communication, and cross-lingual translation were the three main areas of concentration of the strategy.

The Photon engine was used to offer multiplayer capabilities, allowing players from all over the world to connect to the game and speak with one another in real-time. Players may input messages in their original tongue, which are then instantly translated into the recipient's language, using a specially designed chat system.

JavaScript was used to integrate the webkitSpeechRecognition API, allowing players to talk in their local language and have their words instantly translated into text. This enabled voice-enabled communication. Through the chat system, the transcribed text is subsequently relayed to other players in real-time, enabling seamless multilingual dialogue. Finally, the Google Cloud Translation API was utilized to provide real-time, automated translation of speech and texts between languages.

Figure 3: Solution Approach

Text to Speech flow

The player sends their message in their native language and the server returns the message in the receiver's native language. The received text message is then converted into speech and the player can hear it.
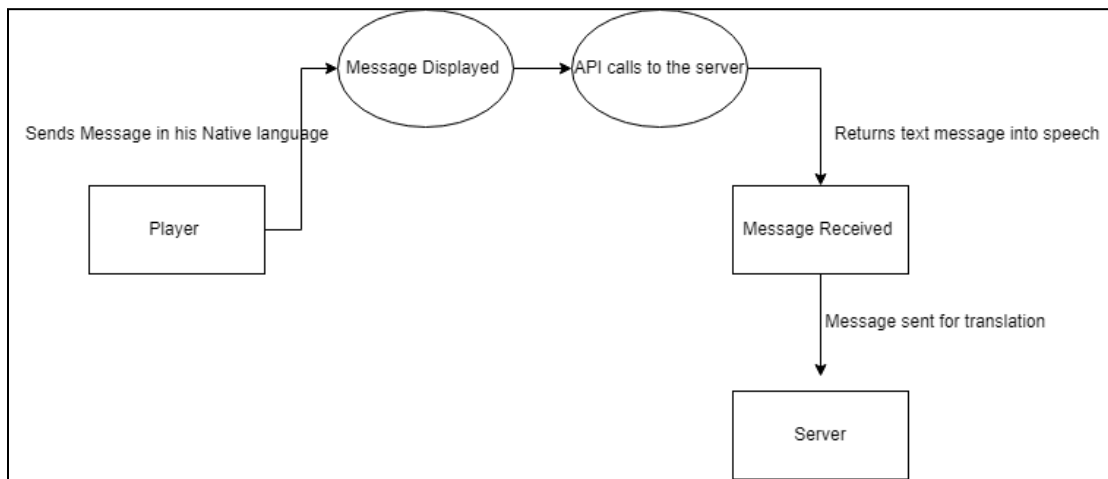
Figure 4: Text-to-Speech Flow

Speech-to-Text Flow

The player can speak any message they wish to deliver in their native language and other players can receive it in their native language.
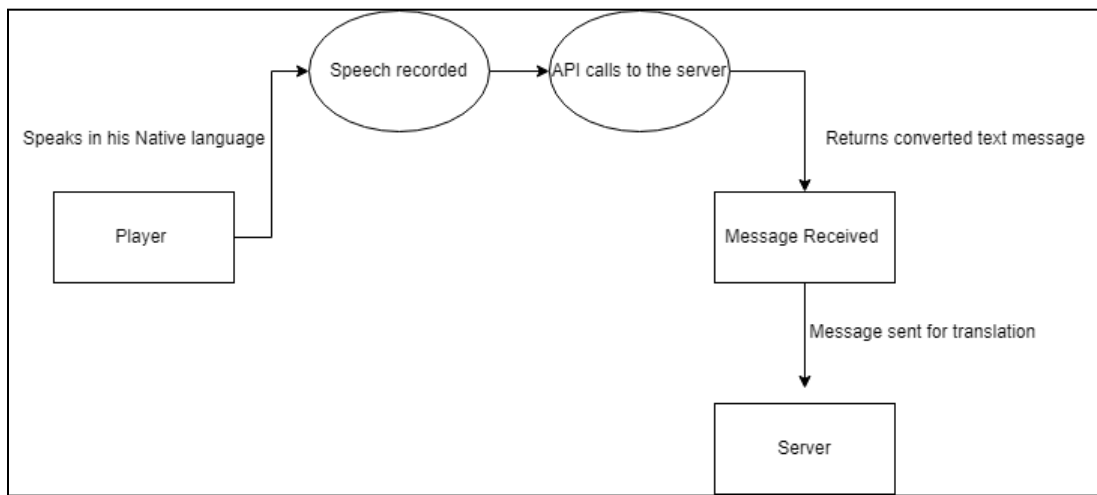
Figure 5: Speech-to-Text Flow

Translation Flow

The player sends a message in the chat window and it is then translated to the receiver's native language through a Python script running in the backend of the application.
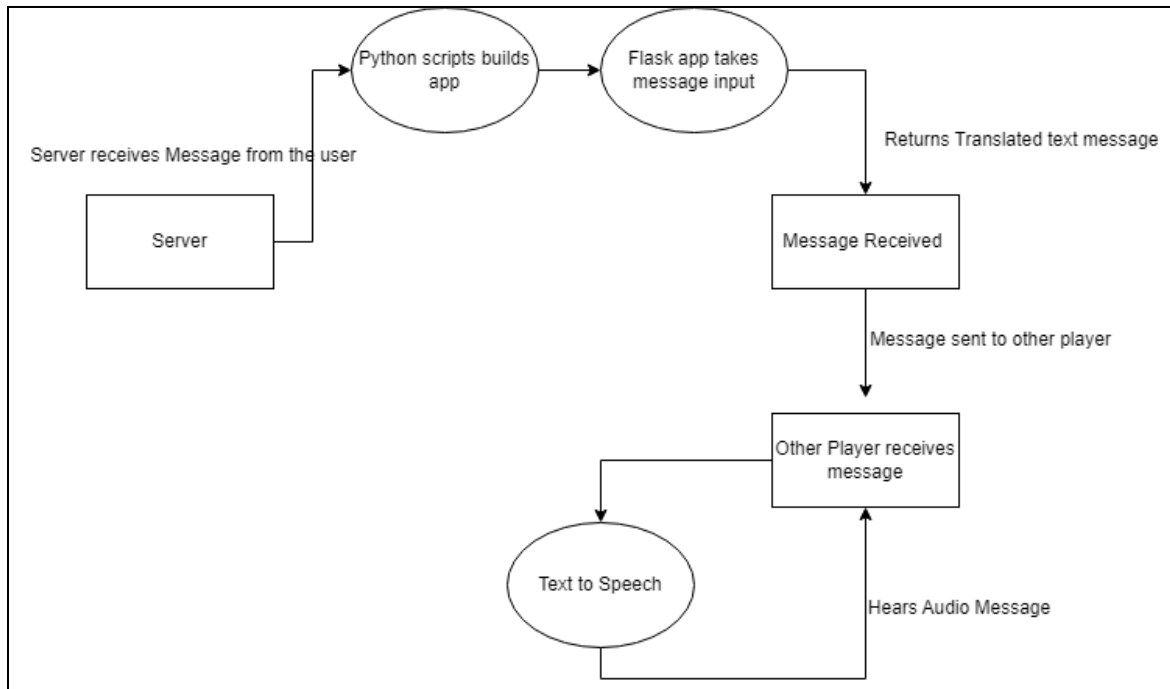
Figure 6: Translation flow

Game Joining Flow

The user starts the game and can enter their username. The user can either start their own lobby or can join some other lobby where they can get connected with the users around the world. Once the user has joined/started a lobby, they can simply start the game.
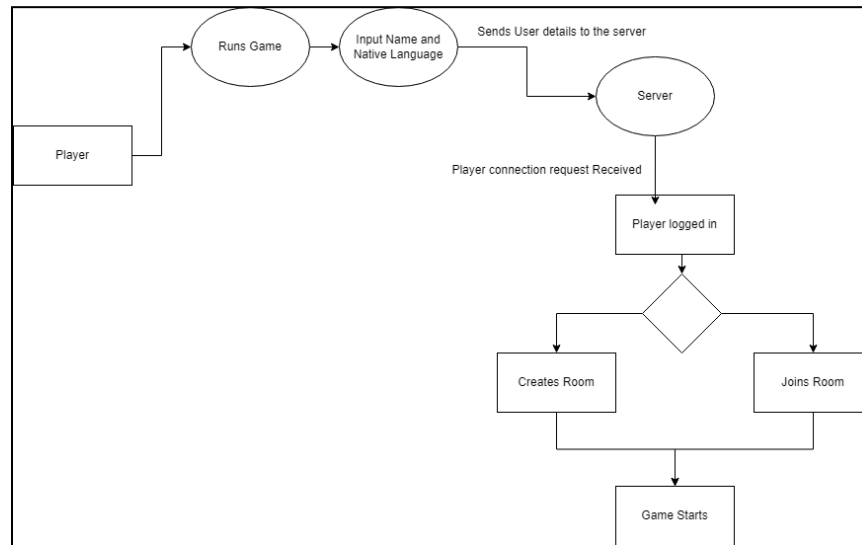
Figure 7: Game Joining flow

Chat Panel Flow

Player 1 sends the message in the chat panel and it's then passed onto the server where the message gets translated into Player 2's native language and is displayed in the same format in the chat panel.
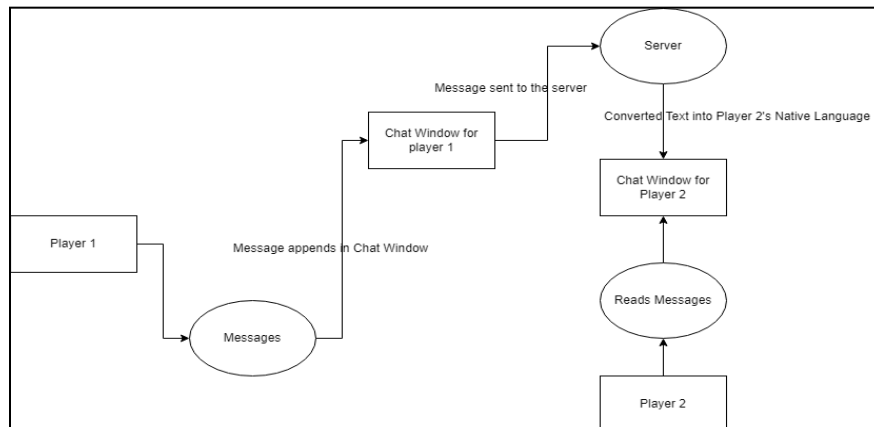
Figure 8: Chat Panel Flow

Chapter 4

Modeling and Implementation Details

4.1   Design Diagrams

4.1.1 Control Flow Diagram
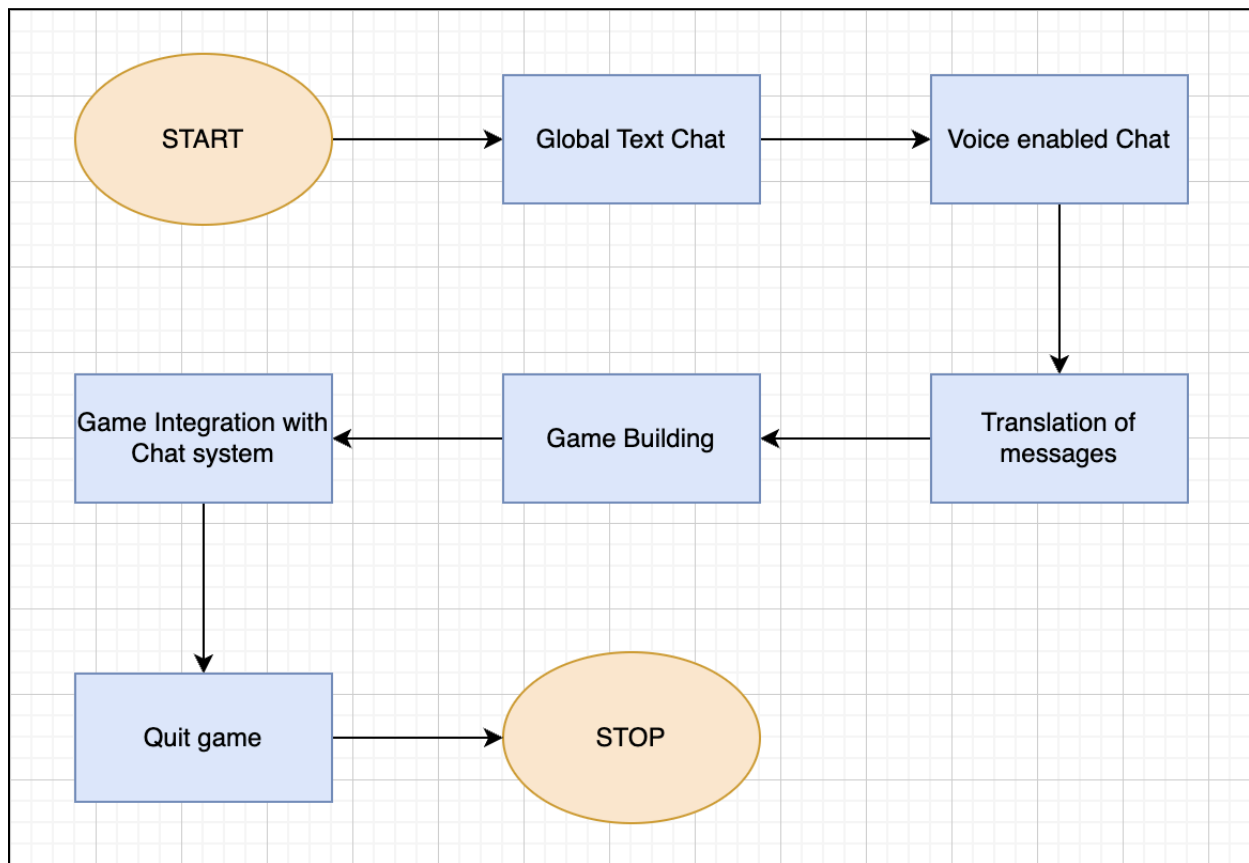
Figure 9: Control Flow Diagram

The user is given the opportunity to text chat with other players once they have started the game. This is achieved by using a unique chat platform that lets users compose messages in their own language. Cross-lingual communication is made possible by the automatic translation of these messages into the recipient's native tongue.

Using JavaScript's webkitSpeechRecognition API, the game also allows voice-activated communication. When a player uses their native tongue, the API converts what they say into text, which is then communicated to other players through the in-game chat system in real-time. Regardless of their language, this enables players to converse with one another in real-time using both text and speech.

All chat messages are automatically translated into the recipient's language to facilitate multilingual dialogue. The Google Cloud Translation API is used for this, which offers extremely accurate translations in real-time. The Unity engine, which offers a strong and adaptable basis for developing rich, immersive game worlds, was used to build the game itself. Regardless of their language, players from all over the world will find the game to be interesting and fun.

The game's proprietary chat system is also smoothly incorporated into it, allowing players to interact with one another in-game in real-time. The Photon engine, which offers strong messaging capabilities and high-performance multiplayer capability, is the foundation around which the chat system is constructed.

Overall, the control flow diagram provides a high-level overview of how the various components of the Cross-Lingual Voice-Enabled Information Processing in Game Environment project work together to enable seamless cross-lingual communication in a game environment.

4.1.2 Activity Diagram
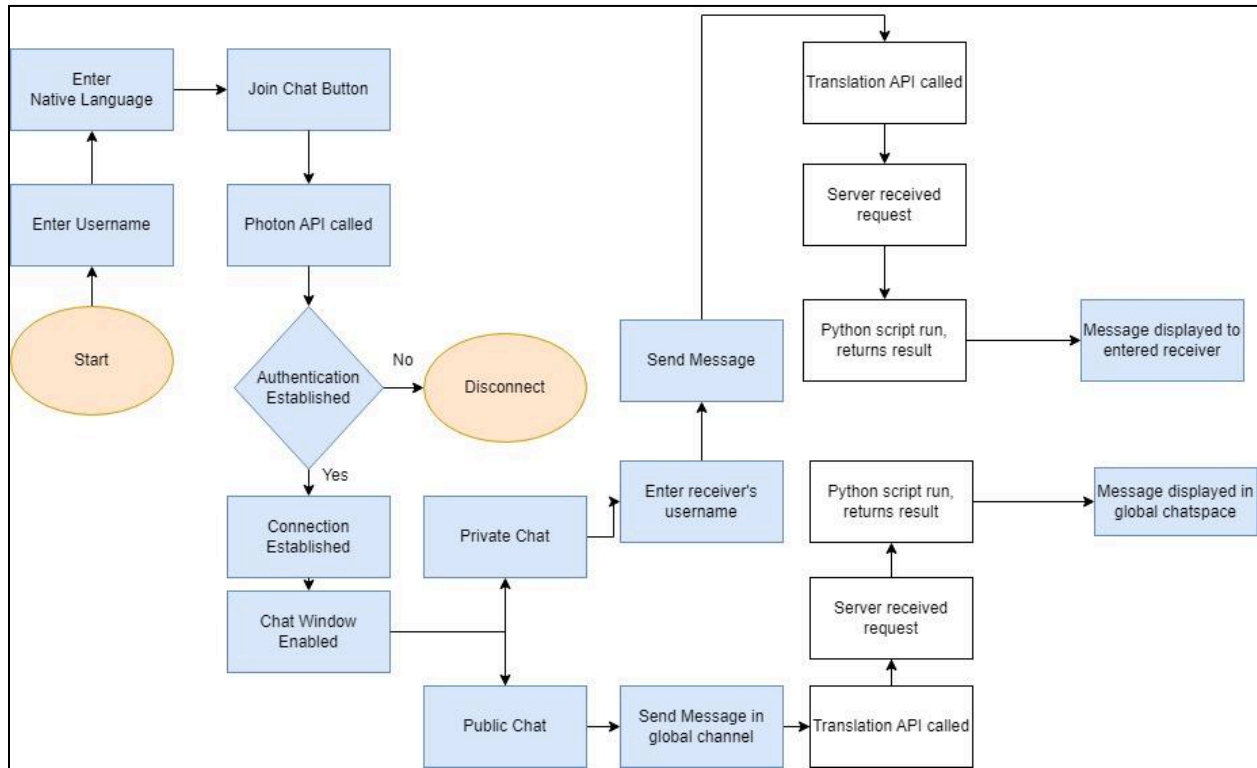
Figure 10: Activity Diagram

The activity diagram shows the process flow of the Cross-Lingual Voice-Enabled Information Processing in Game Environment project, starting with the user entering their username and native language, and clicking on the "Join Chat" button. If the connection is established, the chat window is opened and the microphone for voice communication is turned on. From there, the user can choose to type a message in the public chat or send a private message to a particular user.

In both cases, the translation API is called on a backend Python script hosted at netlify.com, which returns the translated message. The translated message is then displayed in the chat window, either for all users or only for the recipient in the case of a private message. If the connection is not established, the user is disconnected from the game.

4.1.3 Data Flow Diagram
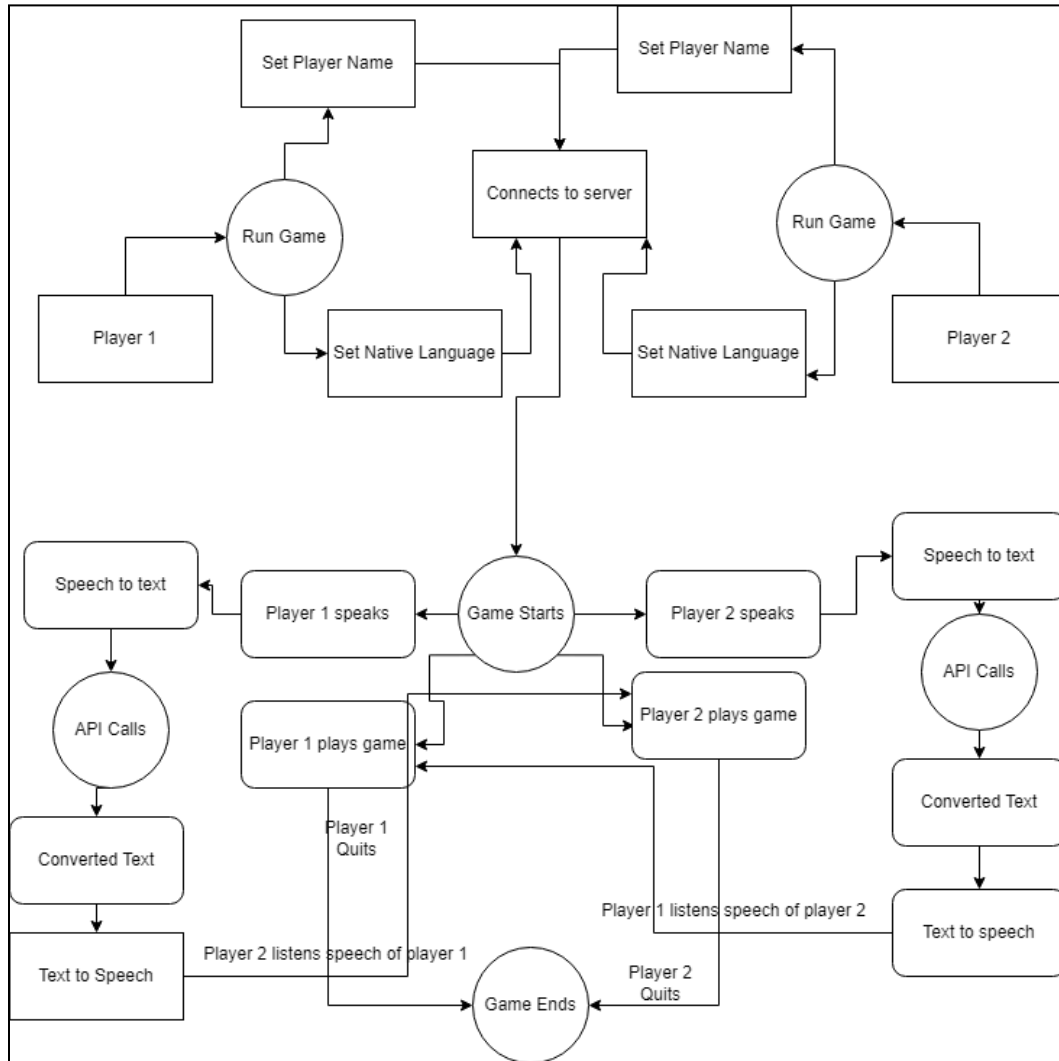
Figure 11: Data Flow Diagram

When a player starts this application, they have two options to fill in - 'Set Player Name' and 'Set Native Language'. After entering these details, the player gets connected to the online server where they go through authentication. Now the integrated game starts, in which players are spawned in the same gaming instance where they can play and talk. Every individual player can speak in their own native language and their speech is converted into text an API is called which converts that text into another player's native language and then that translated text is converted into speech at the receiver's end.

4.2   Implementation details and issues

The goal of the project is to develop an app that will provide users everywhere with a more connected and inclusive gaming experience. Users may speak in real-time using the app's language translation. The software fosters international friendships and a feeling of community by allowing users to communicate in their original tongues.

4.2.1   Join Chat Panel

The Join Chat Panel function is a part of the user interface that gathers crucial user data required for logging into the game's chat system. The user inputs their preferred username and their native language in the feature's two input areas. The ability to seamlessly and effectively communicate with other players in the game depends on having access to this information. The user may rapidly submit their details and join the game's chat system using the Join Chat Panel's straightforward and user-friendly interface.

The Photon engine API is called to create a connection to the chat system once the user has supplied their details and clicked the "Join Chat" button. The user can start speaking to other players in their own language if the connection is successful. Global players may join and speak with one another easily thanks to the Join Chat Panel function, which is a crucial part of the game's multilingual communication system.

Figure 12: Join Chat Panel

4.2.2 Chat Room

The Chat Room is a key component of the cross-lingual communication system in the game. It appears after the user has successfully connected to the Photon chat server through the Join Chat Panel feature. The Chat Room allows users to communicate with other players in their native language.

It contains a message input field where the user can type in their message in their native language and press the send button to send it to the other players in the chat room. It is a central hub for communication in the game and is designed to be simple and user-friendly.
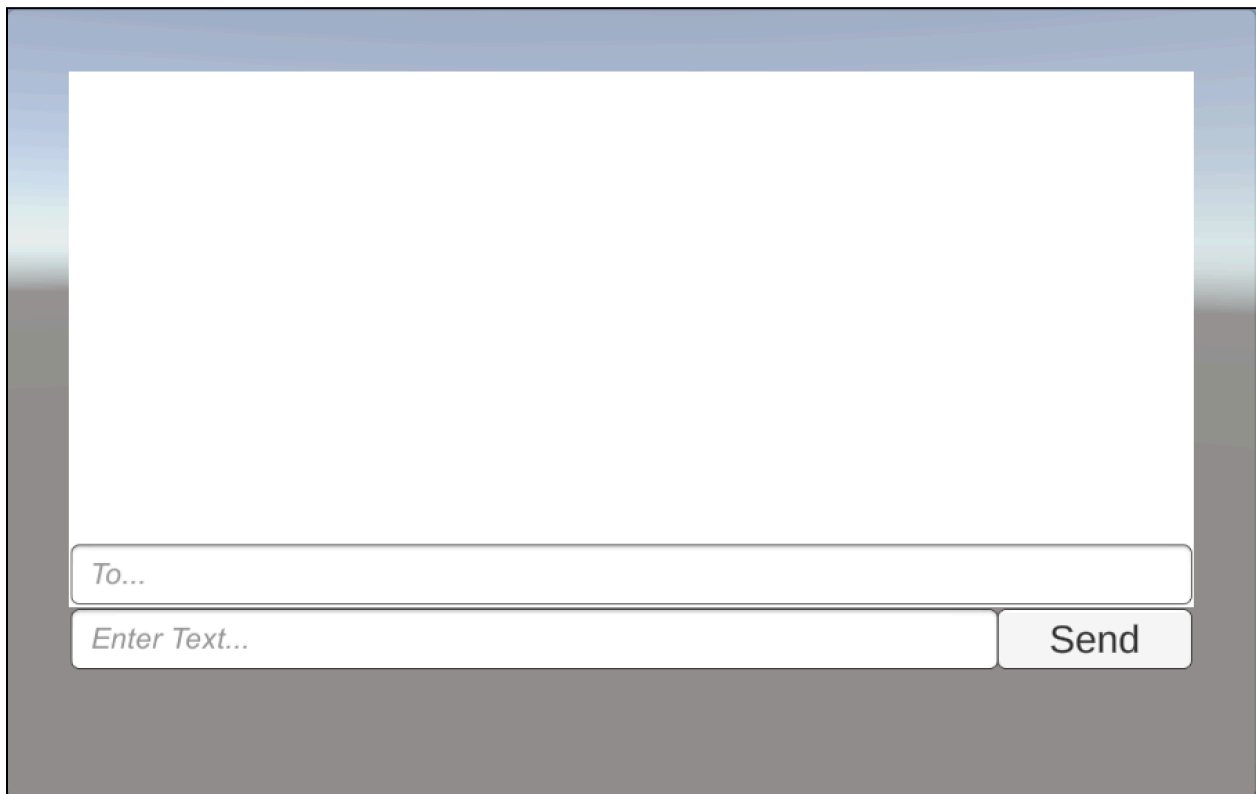
Figure 13: Chat Room

4.2.3 Private Messages

In addition to the Chat Room, the cross-lingual communication system also includes a Private Messages feature. This feature allows users to send private messages to a specific player in the game without other players seeing the message. To use this feature, the user can simply enter the username of the player they want to send the message to in the Input field UI provided. Once the username is entered, the user can then type in their message in their native language and press send to send the message directly to that player.

The Private Messages feature is an essential tool for players who want to communicate privately with another player in the game. It allows them to communicate in their native language and avoid any language barriers that might exist.

Figure 14: Private Messages

4.2.4 Real-time translation

One of the key features of the cross-lingual communication system is Real-Time Translation. This feature enables players to communicate with each other in their own native language without any language barriers. Whenever a player types a message in their native language, the message is automatically translated into the recipient's native language in real-time. This is made possible by a Python script hosted on netlify.com that performs the translation function. The script uses Google Translate API to translate the message from the sender's language to the recipient's language.

Real-Time Translation is an important feature of the game, as it allows players from all over the world to communicate effectively and seamlessly. It eliminates the need for players to learn a new language in order to play the game and ensures that everyone can communicate in a way that is comfortable and natural for them.
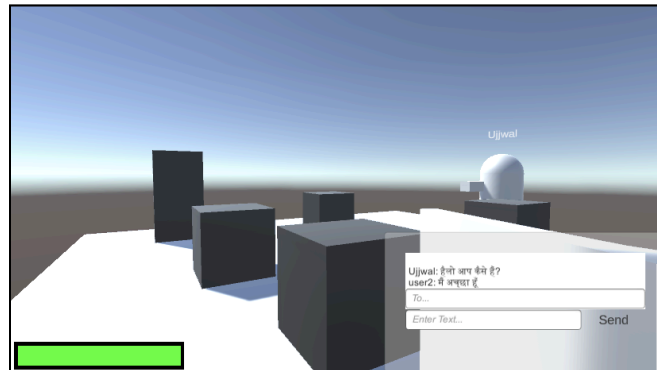
Figure 15: Real-time translation 1
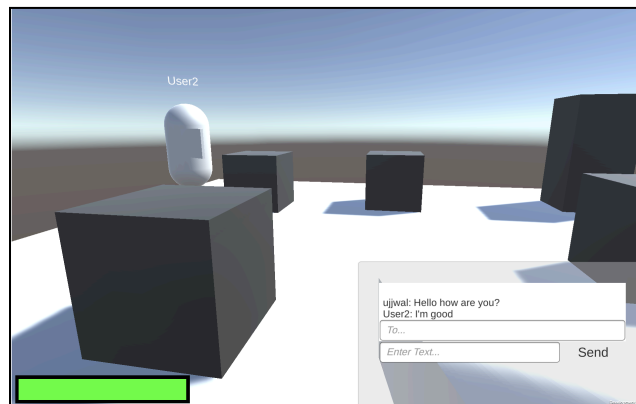
Figure 16: Real-time translation 2

4.2.5 Voice-enabled communication

The voice-enabled communication feature allows users to communicate with each other using their voice instead of typing. This feature is built using the Web Speech API, which enables speech recognition and synthesis in web browsers. When the user speaks, the speech is recognized using the Web Speech API and converted into text, which is then sent as a chat message to other users in the chat room. Other users can also speak, and their speech will be recognized and converted into text in real time. This feature helps users who may not be able to type or who prefer to use their voice to communicate, improving accessibility and user experience.
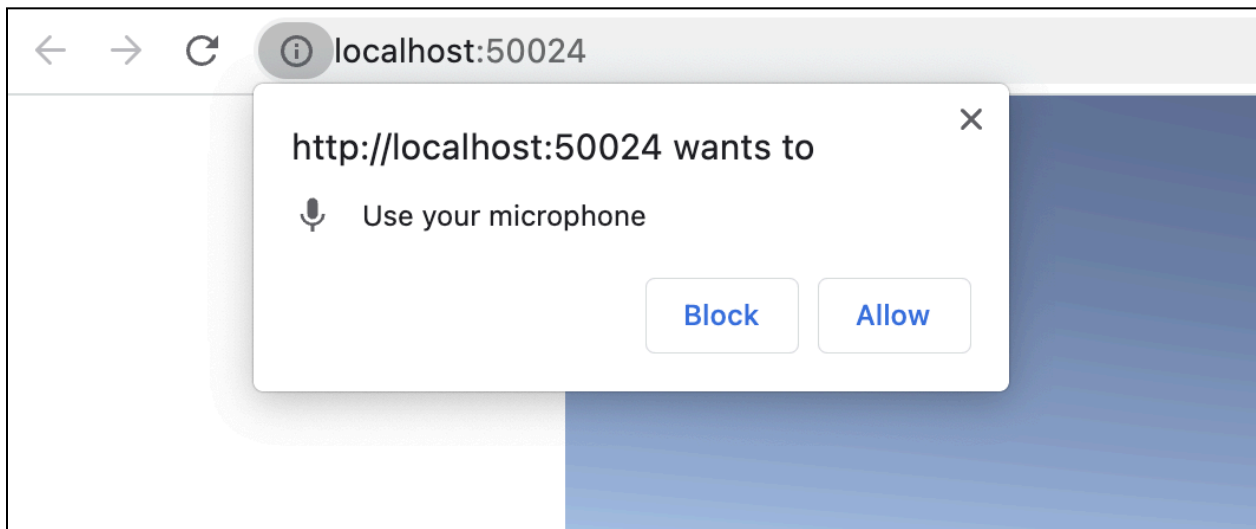
Figure 17: Voice-enabled communication

4.2.6 Text-to-Speech description

Users may listen to text messages instead of reading them thanks to the text-to-speech technology, which turns them into speech. When a user sends a text message, the text is first translated into the recipient's native tongue, and then, using the Web Speech API, the translated text is turned into speech. Following then, the speech is played via the recipient's speakers so they may hear it in their own language. The accessibility of this feature benefits users who might have trouble reading or would rather listen to communications. Additionally, it improves the gameplay experience by enabling more effective and efficient communication between players, particularly in hectic playing conditions when typing messages may not be feasible.

4.2.7 Game integration with Cross-Lingual Voice enabled module

The Cross-Lingual Voice enabled module is integrated into the game which makes the game more interesting. The voice enabled feature helps the players to just speak their message and it gets automatically typed in the chat panel. The cross-lingual feature translates the message to the recipient's native language and the text-to-speech feature converts the received message and delivers it to the player using the device's speaker. All these features combined with the game provide an efficient and effective way of communication while playing with players from around the world.
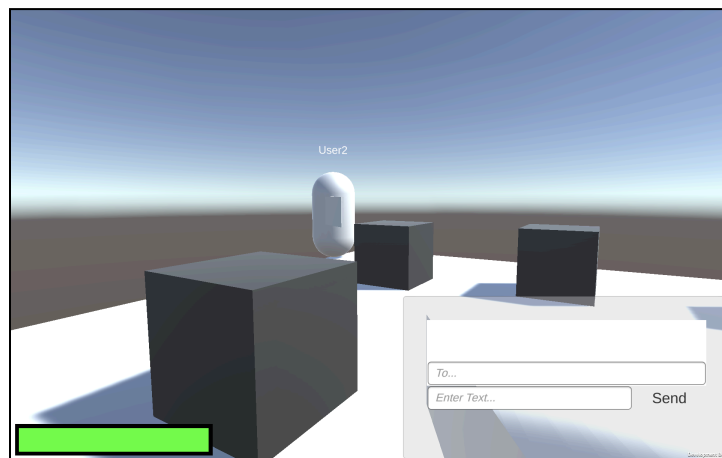
Figure 18: Game integration with Cross-Lingual Voice enabled module

4.3 Risk Analysis and Mitigation

S. No.	Risk Area	Statement	# of risk statements	Weights	Total Weight	Priority
1.	Data privacy and security	The app will be processing user data, which may contain sensitive information. To mitigate the risk of data breaches, the app should be designed with robust security features such as encryption.	1	10	10	1
2.	Translation accuracy risks	Despite using machine learning algorithms, there is a risk of translation inaccuracies that could impact the user experience. To mitigate this risk, the app should undergo rigorous testing and validation before release.	1	10	10	2
3.	Technical risks	The app relies heavily on technology, and any technical issues such as server downtime, network failures could impact the user experience. To mitigate this risk, the app should be designed with redundancies and failover mechanisms.	1	10	10	3

Table 2: Risk Analysis

Chapter 5

Testing

5.1   Testing Plan

Test strategy - The test strategy for the multiplayer game with language translation functionality, built using Unity engine and Flask API, can be structured as follows:

1. Unity Testing: This testing will ensure that the game logic and features are working as expected in the Unity engine. Unit tests can be written to test the different components of the game, such as player authentication, player name and native language input, and gameplay mechanics. These tests will ensure that the game's logic is working as intended, and the functionality is not broken during any future updates.

2. API Testing: Testing of the Flask API can be done using tools like Postman or cURL to verify that the API calls are being made correctly and that the responses are as expected. These tests will ensure that the API is sending and receiving the data correctly and that there are no errors or unexpected behavior in the API.

3. Performance Testing: Performance testing will ensure that the game and API can handle the expected load and provide a seamless experience to the players. This testing will involve simulating a large number of players joining the game and checking if the game's performance and the API response time are within acceptable limits.

4. Compatibility Testing: Compatibility testing will ensure that the game and API are compatible with different platforms and browsers. This testing will involve testing the game on various operating systems and browsers to ensure that the game works correctly across all platforms.

5. Localization Testing: Localization testing will ensure that the game is correctly translated into different languages, and the text-to-speech conversion works as intended. This testing will involve testing the game's functionality with different languages and accents to ensure that the translations are accurate and natural-sounding.

By following the above test strategy, the multiplayer game with language translation functionality can be thoroughly tested and ensure that it is robust, and secure, and provides a seamless experience to the players.

Type of Test	Will the test be performed?	Comments	Software Component
Unit Test	Yes	Testing each module on respective systems	Unity and Postman
Regression Test	Yes	Testing the whole game after integrating modules on different Android devices.	Windows/Mac
Beta Test	Yes	Testing is done by the end users from their laptops	Windows/Mac
Alpha Test	Yes	Testing is done by the developers	Windows/Mac

Table 3: Type of Test

Role	Name	Start Date	Completion Date	Hours	Comments
Game Design	Ujjwal Sachdeva, Roshni Singh	01-01-23	05-04-23	30	Built 3D game environment of players in the same plane.
Server Handling	Tanishk Gupta and Roshni Singh	10-02-23	12-04-23	30	Built server, connected with git, made Rest APIs, wrote Python scripts.
Integration	Ujjwal Sachdeva and Tanishk Gupta	15-02-23	20-04-23	30	Integration of the Server with the game environment.

Table 4: Role

5.2   Component decomposition and type of testing required

The app is composed of the following parts:

1. Chat Box
2. Private Chat Box
3. Voice-enabled
4. Speech to text
5. Text to Speech
6. API calls for Language Text Translator.

Testing

1. Test the player name and native language input to ensure that the game can correctly detect and display the chosen language.
2. Test the gameplay mechanics to ensure that players can move, interact with objects, and communicate with each other seamlessly.
3. Test the Flask API's endpoint to ensure that it receives and sends data correctly.
4. Test the language conversion functionality by sending the text in different languages and verifying that the converted text is accurate and natural-sounding.
5. Test the game's performance by simulating a large number of players joining the game and verifying that the game runs smoothly without any lag or crashes.
6. Test the API response time by sending multiple requests simultaneously and ensuring that the API responds within an acceptable time limit.
7. Test the game and API's security by conducting vulnerability assessments and penetration testing to ensure that they are protected against potential threats.
8. Test the code for potential vulnerabilities and ensure that sensitive user data is protected.
9. Test the game on different operating systems and browsers to ensure that it works correctly on all platforms.
10. Test the game's compatibility with different devices, screen resolutions, and network speeds. Test the language conversion functionality by sending the text in different languages and accents to ensure that the translations are accurate and natural-sounding.
11. Test the game's functionality with different languages to ensure that the translated text is correctly displayed on the player's screen.

Permissions - The game needs the following permissions before starting the application:
Audio - The game requests to record and access your mic.

S. No.	List of Various Components that require testing	Type of Testing Required
1.	Players game window	Unit Test
2.	Players Message window	Unit Test
3.	Speech-to-text and Language Text Translator.	Alpha and Beta
4.	Integrating all the modules.	Regression Test.

Table 5: Type of Testing Required

5.3   List of all test cases

For testing the game, there were some test cases on which we tested to check the game:

1. Authentication Testing:
A. Verify that players are asked to enter their credentials before joining the game.
B. Verify that players cannot access the game without entering valid credentials.

2. Player Input Testing:

A. Verify that the game can detect and display the chosen language correctly.
B. Verify that players can change their name and native language as needed.

3. Gameplay Testing:

A. Verify that players can move, interact with objects, and communicate with each other seamlessly.
B. Verify that players can hear the text-to-speech conversion of other players' messages.
C. Verify that the language conversion is accurate and natural-sounding.

4. API Testing:

A. Verify that the Flask API can send and receive data correctly.
B. Verify that the API can convert text to different languages accurately and quickly.

5. Performance Testing:

A. Verify that the game can handle a large number of players without lagging or crashing.
B. Verify that the API can handle a large number of requests without slowing down or crashing.

6. Compatibility Testing:

A. Verify that the game works correctly on different operating systems and browsers.
B. Verify that the game is compatible with different devices, screen resolutions, and network speeds.

7. Localization Testing:

A. Verify that the language conversion is accurate for different languages and accents.
B. Verify that the translated text is correctly displayed on the player's screen.
C. Verify that the translated text is natural-sounding and easy to understand.

Test Case ID	Input	Expected Output	Status
1.	Player 1 and Player 2 logins	Login Successful	PASS
2.	The player sends text globally.	Messages received to all other players	PASS
3.	The player sends a private message	Message received by the right player.	PASS
4.	The player speaks in English	Player with French native language hears the voice in French.	PASS
5.	The player speaks in French	Player with English native language hears the voice in English.	PASS
6.	The player speaks in Hindi	Player with English native language hears the voice in English.	PASS
7.	Player 1 quits	Connection closed for player 1	PASS
8.	Player 2 quits	Connection closed for player 2	PASS
9.	Player 1 and Player 2 fight each other.	Their health reduces.	PASS
10.	Player 1 dies	Player 1 respawns at different locations.	PASS

Table 6: Test Case

5.4   Error and Exception Handling

Following errors were faced while working on the project:

1. Unable to call API to our server. We have to create an access token to connect our game project to the server.
2. Private messages got mixed with global messages, so for solving this issue we have to add a separate option to send a private message.
3. Game crashes/stop responding due to server lag, as it takes time to run Python scripts and return translated result in that API call, so changed the code to work with the time lags.

Test Case Id	Test Case for Components	Debugging Technique
1.	To check players health reduced during game.	Print Test
2.	To check if the input string of the players sent message matches with the received message string	Print Test
3.	To check the voice translated matches the real translated text.	Reverse process Test.

Table 7: Debugging Technique

5.5   Limitations of the solution

1. Language Support: The accuracy and reliability of the language-translation feature depend on the quality and availability of language models and APIs. If the language model or API used is not accurate or reliable, it may result in incorrect or unnatural translations.

2. Network Dependencies: The multiplayer game is dependent on network connectivity and server availability. If the network connection is slow or unstable, or if the server is down, the game may not function properly, resulting in delays or disconnections.

3. Hardware Requirements: The game may require high-end hardware to run smoothly and provide a good user experience. Players with lower-end hardware may experience lag or other issues, which can impact the overall gameplay experience.

4. Compatibility Issues: The game may not be compatible with all operating systems or devices. Some players may have difficulty running the game due to compatibility issues.

5. Security Risks: The multiplayer game requires players to provide sensitive information such as their names and native language. If the game's security measures are not adequate, this information may be at risk of being hacked or stolen.

6. Cost: Hosting the game and the required APIs may require significant resources and may not be financially viable for small-scale developers.

Chapter 6

Findings, Conclusion, and Future Work

6.1   Findings

The app aims to address the issue of language barriers during gameplay by building it through Unity and utilizing machine learning and natural language processing to provide real-time chat translation between players who communicate in different languages. The app will offer a virtual playground where players can come together, play a game, and chat with each other using their native language. To ensure effective communication and interaction between players, the app will also include in-game tools and features such as voice chat, text chat, game-specific commands, and other tools that enhance the gameplay experience and enable effective communication and collaboration.

6.2   Conclusion

The Unity app which is built incorporates natural language processing to offer real-time chat translation between players communicating in different languages during playing the game. The app provides a virtual playground that enables players to participate in games and interact with each other using their preferred language. To accomplish this, the app's chat translation functionality employs machine learning algorithms to evaluate each player's text input and translate it into the other player's language in real time. Furthermore, the app will contain in-game tools such as voice chat, text chat, game-specific commands, and other features to enhance the gaming experience and enable efficient communication and collaboration among players.

6.3   Future Work

1. Automatic language detection: The app would not take input of the user's preferred language rather it would automatically detect when spoken or written in the chat to communicate with other players.

2. Improved translation accuracy: Even with machine learning, the accuracy of translation can sometimes be limited due to some words which are not in the Wordnet. To improve accuracy, the app can use more advanced natural language processing techniques, such as deep learning, to better understand the context of the conversation and provide more accurate translations.

3. Integration with other games: The proposed solution can be integrated with other games beyond Unity, allowing players of different games to communicate with each other in their native language.

4. Customization of translation: The app can allow players to customize the translation to better suit their communication style. For example, some players may prefer informal translations that use slang and idioms commonly used in their language.

5. Integration of additional communication modes: While the proposed solution includes voice and text chat, other modes of communication can be integrated, such as gesture recognition, to enhance the player's experience and allow for more natural communication.

6. Analysis of user feedback: The app can analyze user feedback to identify areas for improvement and to determine what features are most useful to players. This feedback can be used to continuously improve the app and its features.

Deployment on Open-source platform

GitHub repository link - https://github.com/ujjwal-webdev/ChattingTesting

Deployment link - https://36majorproject.netlify.app/

Contributors ids -

1)   https://github.com/tanny99

2)   https://github.com/ujjwal-webdev

3)   https://github.com/roshni27s

References

1) Vijay Kumar Sharma, Namita Mittal, Ankit Vidyarthi, Deepak Gupta. 2022. Exploring Web-based Translation Resources Applied to Hindi-English Cross-Lingual Information Retrieval. ACM Transactions on Asian and Low-Resource Language Information Processing. (accessed on: 09/03/2023)

2) Vijay Kumar Sharma, Namita Mittal, Ankit Vidyarthi. 2021. Semantic morphological variant selection and translation disambiguation for cross-lingual information retrieval. Multimedia Tools and Applications. (accessed on: 12/03/2023)

3) Vijay Kumar Sharma, Namita Mittal, Ankit Vidyarthi. 2020. Context-based Translation for the Out of Vocabulary Words Applied to Hindi-English Cross-Lingual Information Retrieval. IETE Technical Review. (accessed on: 12/03/2023)

4) Zhou, Chulun, Yunlong Liang, Fandong Meng, Jie Zhou, Jinan Xu, Hongji Wang, Min Zhang, Jinsong Su. 2022 "A Multi-Task Multi-Stage Transitional Training Framework for Neural Chat Translation." IEEE Transactions on Pattern Analysis and Machine Intelligence (accessed on: 12/04/2023)

5) Xia, Peng. "3D Game Development with Unity: A Case Study: A First-Person Shooter (FPS) Game." (2014) (accessed on: 15/04/2023)

6) Bae, Jae Hwan. "Development of smart game based on multi-platform game engine." International Journal of Multimedia and Ubiquitous Engineering 11, no. 3 (2016): 345-350 (accessed on: 17/04/2023)

7) "The Leading platform for real-time 3D creation", https://unity.com/ (accessed on: 10/02/2023)

8) "Machine Translation using Transformers in Python", https://www.thepythoncode.com/article/machine-translation-using-huggingface-transformers-in-python (accessed on: 25/02/2023)

9) "How to translate languages in Python with Google Translate and DeepL", Ilya Krukowski,https://lokalise.com/blog/how-to-translate-languages-in-python-with-google-translate-and-deepl-plus-more/ (accessed on: 27/02/2023)

10) "Voice dubbing services are key to video game production",
https://www.languageconnections.com/voice-dubbing-services-are-integral-to-the-video-game-translation-process/ (accessed on: 09/04/2023)

11) Matt Strach, "5 Tips for Powering Up Your Video Game Voices in Any Market",
https://www.getblend.com/blog/video-game-voice-over/ (accessed on: 11/04/2023)

12) "Game development with Azure Cognitive Service for Speech",
https://learn.microsoft.com/en-us/azure/cognitive-services/speech-service/gaming-concepts (accessed on: 13/04/2023)

13) Anastasia Dimitriadou, "Game Translation and Localization – A Definitive Guide",
https://www.pangea.global/blog/game-translation-and-localization-a-definitive-guide/ (accessed on: 16/04/2023)

14) Rob Speer, "conceptnet5", https://github.com/commonsense/conceptnet5/wiki/Languages (accessed on: 18/04/2023)

15) "The Future of Language: How Natural Language Processing is Changing the Game",
https://devnote.in/the-future-of-language-how-natural-language-processing-is-changing-the-game/ (accessed on: 20/04/2023)

16) "Create a Real Time Voice Translator using Python",
https://www.javatpoint.com/create-a-real-time-voice-translator-using-python (accessed on: 21/04/2023)

17) "Voice Controlled Games: The Rise of Speech Technology in Gaming",
https://summalinguae.com/language-technology/voice-controlled-games/ (accessed on: 22/04/2023)