Amsterdam University of Applied Sciences

A study on increasing Non Player Character realism by leveraging Large Language Model capabilities

by Chris Huider

500844542

+31(0)637602465

chrishuider@gmail.com

Bachelor Thesis

Game Development

Faculty of Digital Media and Creative Industry

Amsterdam University of Applied Sciences

Supervising teacher: Coline Pannier

Academic year 2023-2024

Version 4

Contents

Abstract

This paper examines whether implementing large language models (LLMs) into non-player characters (NPCs) can increase their experienced realism, which will be tested by implementing an LLM into the passenger NPCs in the KLM extended reality center of excellence's virtual reality communication training.

First, how can the realism of NPCs be defined? There is no agreed upon definition of NPC realism, however in this paper, perceived believability will be used to judge the realism of the NPCs.

The passenger NPCs have to adhere to certain criteria to be perceived as believable, these include being able to deceive the player, convey values from the historical column of the game agent matrix and the NPC should be perceived as possibly existing in their own social environment.

The next question to be answered is whether an LLM can make an NPC adhere to these criteria. At this moment, LLMs vary widely in their conversational abilities, making it difficult to determine whether an LLM will be able to achieve the high complexity of social context required for it to be perceived as believable, however, an NPC might still benefit from the emergent behaviour introduced by the LLM.

To answer this question, an experiment was conducted where an NPC whose responses are generated using an LLM is compared to an NPC whose responses are predetermined.

This experiment's results show that the biggest influence actually came from the addition of being able to speak to NPCs through voice. In addition, being able to choose freely what to say, instead of having to choose between multiple options also had a positive influence.

The introduction of the LLM did cause some unrealistic behaviour in the NPC's responses, which was a negative influence. Additionally, the extra time it takes for the LLM to respond does not affect the experienced realism in a significant way, however, further research will be necessary to support this claim.

Definition of terms

Application Programming Interface (API)
An application programming interface, or API, is a mechanism that enables two software components to communicate with each other using a set of definitions and protocols.

Artificial intelligence (AI)
Artificial intelligence, or AI, is technology that enables computers and machines to simulate human intelligence and problem-solving capabilities.

Extended Reality Center of Excellence (XRCoE)
The Extended Reality Center of Excellence, or the XRCoE, is a team that's part of the KLM IT department that specializes in creating and maintaining extended reality (XR) experiences centered around subjects related to aviation.

Hosting
Hosting describes the act of running a server on your device and allowing other users who are connected to the internet to connect to your server or in other words, lobby.

Koninklijke Luchtvaart Maatschappij (KLM)
The KLM Royal Dutch Airlines is the oldest operating airline in the world, and has three core activities: passenger transport, freight transport and aircraft maintenance.

Large language model (LLM)
Large language models, or LLMs are machine learning models that can comprehend and generate human language text.

Lobby
In a multiplayer game, players can connect to lobbies or in other words, servers, which allows them to play together.

Machine learning (ML)
Machine learning, or ML, is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Non-player character (NPC)
A non-player character, or NPC, is any character in a game that is not controlled by a player.

Virtual reality (VR)
Virtual reality, or VR, is a simulated experience that employs position tracking and 3D near-eye displays to give the user an immersive feel of a virtual world.

Introduction

Issue

The research into the use of a Large Language Model (LLM) in the KLM's virtual reality (VR) communication training opens the door to possibilities that the XR Center of Excellence (XRCoE) may not have fully explored yet. There is a chance that an LLM can create interactions that are much closer to the complexity and variability of real human conversations than can currently be simulated by a trainer pushing buttons. This offers a unique opportunity to make the training much more realistic and interactive.

Such an LLM can lead to a more personalized and dynamic learning environment, where scenarios adapt in real-time to the actions and reactions of the pilot. This can help develop crucial skills for mastering unpredictable situations, something invaluable in the real world.

The deployment of an LLM can also improve the cost efficiency by reducing the need for continuous trainer involvement and providing a scalable solution adaptable to various training needs. This would not only benefit the quality of training, but also increase its long-term efficiency and effectiveness.

Furthermore, exploring the use of an LLM in the VR communication training could demonstrate that the XRCoE is at the forefront of employing advanced technologies to provide the most challenging and realistic training experiences. This could enable KLM to better prepare pilots for the complexities of their role, allowing them to act with more confidence and competence in difficult situations.

In short, researching new technologies like an LLM offers an exciting opportunity to push the boundaries of the current training methods employed by the KLM and achieve significant improvements in the education of future pilots, making the training methods more dynamic, effective, and cost-efficient.

Assignment

This issue led to the following assignment from the KLM XRCoE:

"Conduct research on how LLMs can enhance the unpredictability and autonomy of NPCs within the KLM's XRCoE's VR 'communication training', aiming to enhance the realism, quality, and efficiency of the training."

Paper subject

The research question of this paper is: "*How can the implementation of a large language model improve non-player character realism?"*; and will discuss the following points:

1. What is the KLM's virtual reality communication training?
2. When is a non-player character considered realistic?
3. How should the passenger character(s) behave to achieve the desired result?
4. How can the unpredictability and autonomy of the passenger characters be enhanced?
5. What is a large language model?
6. Designing an experiment for assessing perceived non-player character believability
7. Potential ways of implementing a large language model into the experiment application
8. Creating the experiment application
9. Conducting the experiment
10. Experiment results

1. What is the KLM's virtual reality communication training?

The KLM's VR communication training is a training done in VR in order to train pilots in making announcements to the passengers while being present in the cabin. It trains the pilot to handle anxiety when making the announcement and helps the pilot adapt to different situations which can occur when making an announcement.

The pilot or trainee enters the VR environment and is placed in the cabin of an Embraer 190 airplane. The Embraer 190 has around 100 seats of which in the training most are filled with passengers.

Currently the training requires an instructor to be present to maximize its effectiveness. This instructor sets the scenario for the training. For example, the pilot has to explain to the passengers why they have a two-hour delay. The instructor also controls the passengers' behaviour, for example, the instructor can instruct them to be angry. This is done through the training panel in the desktop application.

How does the virtual reality communication training application work?

The application is broken up into two applications, the VR headset application and the desktop application. The VR headset application is used by the trainee and runs on a Meta Quest 3. The desktop application is used by the instructor and runs on any laptop or desktop operating on the Windows operating system. Using the desktop application, the instructor can change passenger behaviour and play sound effects. It can also start a recording used for reviewing the training and can change camera positions to allow for different viewing angles. The final functionality the desktop application has is the ability to host lobbies. The VR headset application has to connect to a lobby in order for the desktop application to be able to influence the training.

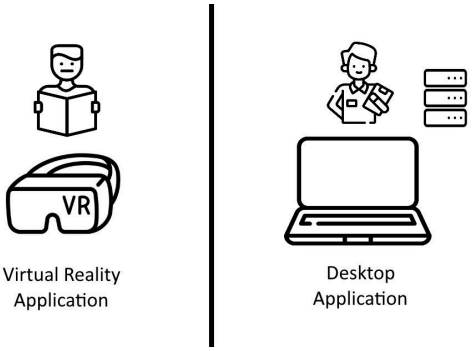

Image 1. Communication training application (image by Chris Huider, 29-04-2024)

The connection between VR headset and desktop applications is done using Photon. It allows for data to be sent over the network when the two applications are connected in the same lobby.

2. When is a non-player character considered realistic?

To increase passenger character realism by enhancing unpredictability and autonomy it is imperative to know how an NPC should behave to be perceived as realistic. As such, a literature study on this subject has been conducted.

What is a non-player character?

An NPC is a character in a computer game that is not controlled by someone playing the game ("NPC," 2024). NPCs are used to serve a multitude of purposes and can have multiple functions. Bartle (2003/2004) produced a typology of NPCs based on the roles they represent and functions they provide within their environment. They are:

- *Buy, sell and make stuff*
- *Provide services*
- *Guard places*
- *Get killed for loot*
- *Dispense quests (or clues of other NPCs' quests)*
- *Supply background information (history, lore, cultural attitudes)*
- *Do stuff for players*
- *Make the place look busy*

Warpefelt and Verhagen (2016; as cited in Warpefelt and Verhagen, 2017) produced a typology of NPCs based on the one produced by Bartle (2003/2004), seen in Figure 1. This typology iterates on the typology produced by Bartle, by taking into account the context in which the NPC is found in addition to their roles and functions within their environment.

Metatype	Type	Subtype
Functions	Vendor	
	Services	
	Questgiver	
Adversaries	Enemy	Boss
	Opponent	Manipulator
Friends	Sidekick	
	Ally	
	Companion	
	Pet	
	Minion	
Providers	Storyteller	
	Loot provider	

Figure 1. The NPC typology of (Warpefelt and Verhagen, 2016; as cited in Warpefelt and Verhagen, 2017)

Warpefelt and Verhagen iterated on this NPC typology once more by refining it using the Game Agent Matrix (GAM), seen in Figure 2, producing results containing short descriptions of the types in the typology. These descriptions encapsulate the essence of every type of NPC and will henceforth be referred to as the 'updated NPC typology' (see appendix 1 or Warpefelt and Verhagen, 2017, for more). This updated NPC typology will be relevant later.

How is non-player character realism defined?

NPCs also contribute to upholding the believability of a game world (Loyall, 1997). The term 'believability' is often used to determine how realistic an NPC is, but as Togelius et al. (2012) states, it seems not to have a generally agreed upon or precise definition. However, Togelius et al. concluded that the definition of believability is: *"someone believes that some character or bot is real"*.

Warpefelt and Verhagen (2017) add another layer to that definition, stating

	Single Agent	Multiple Agents	Social Structural	Social Goals	Cultural Historical
Act	Goal directed Route following Uses language Uses tools	N/A	N/A	N/A	N/A
React	Adaption Acquires information Crisis response Interruptability Awareness Models of self Rapid emotional response Navigation	Learns from others Models of others Turn taking	Class difference Mob action Social ranking	Disillusionment	Advertising Institutions Roles
Interact	N/A	Face to face Group making Social interaction Memory of previous interactions	Coercion	Clan Wars Cooperation Group conflict Patriotism Power struggles Team player	Etiquette Norm maintenance Sanctions

Figure 2. The Game Agent Matrix of (Warpefelt and Verhagen, 2013; as cited in Warpefelt and Verhagen, 2017)

believability in games is dependent on the context in which the believability is judged. If the player is informed of any special properties or narratives of the alternate reality, improbable feats can be made believable.

De Melo and Gratch (2015) attempt to refute the credibility of using believability to judge NPC realism, stating that because the definition of believability isn't precise, nor agreed upon, and it thus being prone to subjective interpretation, it's hard to measure with any precision or reliability and difficult to study from a scientific standpoint.

To address this problem, Warpefelt and Verhagen (2017) constructed the GAM, seen in Figure 2, which is a model that shows the level of complexity needed for an NPC to be perceived as believable in a social context. This will be relevant later.

What type of non-player character are the passengers in the communication training?

The passengers in the communication training are meant to converse and react to the pilot.

The description from the Manipulator subtype states the following: *"These NPCs exist to socially manipulate the player into performing certain actions. They generally require a higher degree of social capability than their Opponent siblings, and must be able to convincingly portray values from the Cultural Historical column."* This description matches the passenger character best out of the types in the updated NPC typology. (for more, see appendix 1)

Thus, the conclusion can be drawn that the passenger characters fall within the Opponent type and the Manipulator subtype.

What criteria will a passenger character have to adhere to, to be considered realistic?

As Warpefelt and Verhagen (2017) describe, the role the Manipulator represents is to socially manipulate the player into performing certain actions. To be able to do this, the Manipulator requires a high level of social capability, higher than their Opponent parent-type, high enough to be able to portray the values from the Cultural Historical column in the GAM, which means the NPC requires the highest complexity necessary in the context of a social interaction in order to be perceived as believable. To achieve this, they need to be able to adapt to the situation and the player, but also interact with the player in such a way that they are perceived to have their own personalities and that they exist in some kind of complex social environment. They should not only be perceived as a socially complex creature, but should also be convincing enough to be able to deceive the player. This list of criteria will henceforth be referred to as the 'passenger character believability criteria' in the remainder of this report.

Considerations for maximizing realism in a non-player character

As Warpefelt and Verhagen (2017) point out, adding to the description of the Manipulator type, the complexity of social capability required for the Manipulator to be perceived as believable is exceedingly rare and very difficult to construct.

Wages et al. (2004) also point out a precaution to take into account when increasing realism, by stating that when increasing realism, immersion might decrease, as when the absolute difference between reality and the virtual environment decreases, the spectator will often be drawn to the remaining differences. Which leads to the question, what is more important, realism or immersion?

3. How should the passenger character(s) behave to achieve the desired result?

In addition to knowing how a passenger character should behave as an NPC, knowing how it should behave as intended in the training is just as important. The passenger characters not only serve the purpose of creating the right atmosphere and environment for the training, but also influence the direction, flow and difficulty of the training.

Expert interview

To get to know how the passenger character(s) should behave to achieve training goals an expert interview has been conducted. In this interview, the expert, an employee of KLM Flight Operations, answered a list of questions. These questions along with the answers given by the expert can be found in appendix 2.

Using these answers, passenger characters behavioural requirements can be determined. The passenger characters should:

- Behave as an average passenger; express emotion, but not too extreme.
- Be able to use explicit language, but not to the extent that it becomes a personal attack.
- Clearly express their interest in the pilot, either through body language, actions or looking at the pilot.
- Challenge the pilot to deal with uncommon behaviours.

4. How can the unpredictability and autonomy of the passenger characters be enhanced?

Knowing how the passenger characters should behave to be perceived as realistic and uphold the training goals, how can it be made so they actually show this behaviour? To find an answer to this question a literature study has been done on the subject.

How can the passenger characters benefit from large language model capabilities?

As described by Volum et al. (n.d.), NPCs enhance the player experience by providing interaction, often through means of conversation. Currently, conversations between player and NPC are highly scripted, which leads to the typical scenario where players aren't able to have a free form conversation with the NPC, but instead have to converse through a set of preset responses that they can give to the NPC.

Even though the player can't have a free form conversation with the NPC, it can still be perceived as believable, as Warpefelt and Verhagen (2017) conclude that no current NPC types need to exhibit emergent behaviour in order to be perceived as believable. However, emergent behaviour in NPCs would improve believability.

Large language models can provide this emergent behaviour to NPCs as they are naturally suited for holding conversations. When implemented in the NPC's dialogue system, they can generate dynamic and contextually appropriate responses based on the player's input, allowing for a free form conversation and making interactions with NPCs more engaging and realistic, reducing repetitive conversations and providing a more explorative experience in the game. (*What Are Large Language Models? | NVIDIA Glossary*, z.d.)

However, as described in the passenger character believability criteria, the LLM will require a high complexity of social context to be perceived as believable. Choi et al. (2023) conclude that currently, LLMs' social capabilities perform moderately at best, with even large LLMs (>10b parameters) varying widely in their conversational abilities, making it difficult to determine whether an LLM will be able to achieve the high complexity of social context required for it to be perceived as believable.

Even though it might not be able to reach the height of being perceived as believable, NPCs might still benefit from the introduction of emergent behaviour.

5. What is a large language model?

To enhance NPC unpredictability and autonomy by leveraging the capabilities of large language models, a study has been conducted on what large language models are.

Large language models are machine learning (ML), and more specifically deep learning algorithms called transformer networks. A transformer network learns by tracking relationships in sequential data, like words in a sentence, from examples in very

large datasets, after which it will be able to recognize, summarize, translate, predict and generate content. (*What Are Large Language Models?* | *NVIDIA Glossary*, z.d.)

How do large language models work?

Before it can be used, a large language model has to be trained. Foundation model LLMs are trained using unsupervised learning. Through unsupervised learning, the LLM will try to find patterns in an unlabeled dataset; these patterns are the aforementioned relationships in sequential data. Because of this training method, these foundation models don't need to be trained for any specific task and can instead serve multiple use cases. (*What Are Large Language Models?* | *NVIDIA Glossary*, z.d.)

These foundation models can be trained further to achieve better results in specific use cases using zero-shot learning and its variations, one-shot and few-shot learning. Using these learning methods, the foundation model is given one or a few examples on how a task can be accomplished using a labeled dataset, which has been labeled for the specified use case. Later on, these models can be customized even further using several different techniques to achieve higher accuracy for specific use cases. (for more, see *What Are Large Language Models?* | *NVIDIA Glossary*, z.d.)

What are large language models used for?

As large language models are transformer models which are trained to recognize, summarize, translate, predict, and generate content using very large datasets, it's important to know it isn't just language used in human communication that it can be trained to use. Code is the language of computers. Protein and molecular sequences are the language of biology. LLMs can be applied to such languages or scenarios in which communication of different types is needed. (Lee, 2023)

For example, an LLM which has learned from a dataset containing molecular and protein structures, can provide chemical compounds to help scientists develop new vaccines or treatments. Other examples of how LLMs could be used are, LLM powered search engines, to improve search results, tutoring chatbots to help students with their studies, composition tools for songwriters, poets and writers, and more. (Lee, 2023)

6. Designing an experiment for assessing perceived non-player character believability

To confirm whether the benefits provided by a LLM have their expected effects on the perceived believability of an NPC, an experiment has been conducted.

What is the primary focus for this experiment?

The experiment is focussed on comparing two different ways of interaction between participant and an NPC, and specifically the perceived believability of the interaction.

The results of this experiment will be used to determine whether the implementation of an LLM can increase the perceived believability of an NPC in the eyes of the participant.

Which ways of interaction will be examined?

A comparison will be made between two different ways of interaction, which I've named:

- The 'Bethesda' interaction
- The 'Inworld' interaction

The 'Bethesda' interaction is named after the classic interaction style Bethesda Softworks uses in their role-playing games (see image 2).

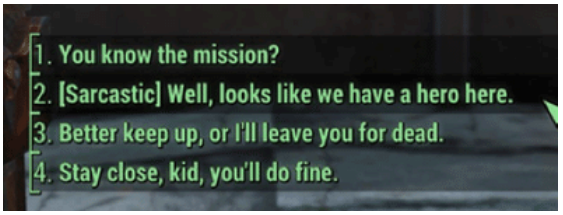

Image 2. Bethesda Softworks typical role-playing game interaction (Fallout 76 example).

In the ‘Bethesda’ interaction, the participant interacts with an NPC through a user interface; the ‘choice menu’. They will be presented with four choices which they can choose from using the user interface. The NPC will respond using a predetermined response.

The ‘Inworld’ interaction is named after the unitypackage used to implement an LLM into an NPC, namely: Inworld. In this interaction, the participant interacts with an NPC through speech. They will be allowed to speak out loud freely (in English). The NPC will recognize the choice and respond with a large language model generated response, based on a predetermined description of the response.

Which steps will the experiment follow?

The process for each time the experiment is conducted will follow the following steps, in order:

1. Preparation
2. Participant briefing
3. ‘Participant information’ questionnaire
4. Experiment execution; interaction 1
5. ‘Besthesda interaction’ questionnaire
6. Experiment execution; interaction 2
7. ‘Inworld interaction’ questionnaire

(for more information on these steps, see appendix 3)

What does the physical test environment and the experiment application look like?

The physical test environment is a quiet and isolated room without any auditory distractions. The only two people present in the physical test environment are the participant and the observer.

The environment in the experiment application is the cabin of an Embraer 190 airplane. It’s filled with 84 passengers. Row 1 up to and including 5 will have animated passenger NPCs, displaying an idle animation. Rows 6 up to and including 10 will be filled with static floating 3D passenger heads. Rows 11 up to and including 25 will be filled with static floating 2D passenger heads.

The experiment application will be made in Unity3D. The Meta XR interaction Unity packages are implemented into the experiment application and a Meta Quest 2 VR headset is used to transport the participant from the physical test environment to the environment in the experiment application. (see chapter 8: Creating the experiment application, for more)

How will data be gathered?

Results will be gathered before, during and after the experiment. Before the experiment is conducted, the participant answers the ‘Participant information’ questionnaire. This questionnaire has been constructed to gather information on the participant by which the results will be categorized.

The first part of the experiment will be executed and the first interaction, which is randomly selected, will be done. After this interaction is completed, the participant fills in its respective questionnaire, after which the same process is repeated for the second interaction.

During the experiment the observer takes note of any noteworthy details, such as anomalies and interesting observations during the interaction. These will be mentioned alongside the results.

The interaction questionnaire (see appendix 5) has main questions, present in both questionnaires, and specific questions, present in only a single questionnaire. These main questions are used for identifying anything that could influence the specific questions and the specific questions are about the interaction.

7. Potential ways of implementing a large language model into the experiment application

To utilize LLMs in Unity, a choice will have to be made about how to implement the LLM. There are multiple potential ways to implement an LLM into Unity. Some of these implementations are plug and play while others require work to implement and finetune.

The possible ways of implementation can also be differentiated between being made by a third- and first-party.

Made by third-party

Third parties offer a variety of LLM implementation into Unity. While using an LLM implementation made by a third party is more time effective, it often comes at the cost of a usage fee. Some examples of these implementations would be Inworld and ChatGPT.

Inworld offers a plug and play solution which enables anyone to implement LLM assisted NPCs into their Unity project. NPCs can be created and adjusted on their website and implemented through their unity package, accessible through the unity asset store. (see Inworld, n.d., for more)

ChatGPT offers an application programming interface (API) which enables anyone to send prompts and receive replies from their LLMs. This API can be used in Unity after implementing it using a wrapper made by a third party, which can be imported into the Unity project using the package manager. After implementation, more work is required to implement it into an NPC, make it work with voice recognition, etc. (see Brockman et al., 2023, for more)

When using third-party implementations, the KLM regulations have to be conformed to. In this case, this primarily means confidentiality is required.

Made by first-party

Making a first-party LLM implementation allows for easy finetuning and specialization for very specific use cases. To the contrary of third-party LLM implementations, a first-party implementation is often cheap or even free, but will require a lot of work to set up. There are some pre-existing code bases, which could speed up the creation of an LLM implementation, some examples being LLMUnity, Android's Neural Network API (NNAPI) and Unity's Sentis.

LLMUnity makes it easy to set up and run LLMs in Unity. This enables anyone to host and use their own LLM within Unity and build their application around them. (see Undreamai, n.d., for more)

Android's NNAPI provides a Native Development Kit (NDK) which enables anyone to create their own machine learning model hosting solution. This does mean the LLM implementation is to be made from scratch when using this API. (see Google, n.d., for more)

Unity's Sentis provides a similar service to Android's NNAPI, as it provides a code base which enables anyone to create their own machine learning model hosting solution. As with Android's NNAPI, the LLM implementation would have to be made from scratch. (see Unity Technologies, n.d., for more)

8. Creating the experiment application

The experiment application will contain two separate experiences, one where the participant interacts with the passengers using the 'Bethesda' interaction and one where they will use the 'Inworld' interaction.

For the 'Inworld' interaction, the choice has been made to use Inworld as LLM implementation in the NPC. This decision was made because the XRCoE was already familiar with Inworld and preferred the choice as further exploration would help them as well.

The environment

The technical environment is the following:

Developed in Unity, version: 2022.3.11f1
Unity packages used:

- *Meta XR Core SDK by Oculus, version: 63.0.0*
- *Meta XR Interaction SDK by Oculus, version: 63.0.0*
- *Meta XR Interaction SDK OVR Integration by Oculus, version: 63.0.0*
- *Meta XR SDK Shared Assets by Oculus, version: 63.0.0*
- *AI NPC Engine v.3.0 - Dialogue & Behavior for Unity - Inworld by AI NPC - Inworld AI, version: 3.3.1*

The virtual environment (the environment in the experiment application) is as described in chapter 6's 'What does the physical test environment and the experiment application look like?' (see Image 3).

Image 3. The virtual environment

In addition to the description from chapter 6, there is an objectives list for the participants to follow (see Image 4). It's located to the left of the participant when they're in the virtual environment (see Image 3).

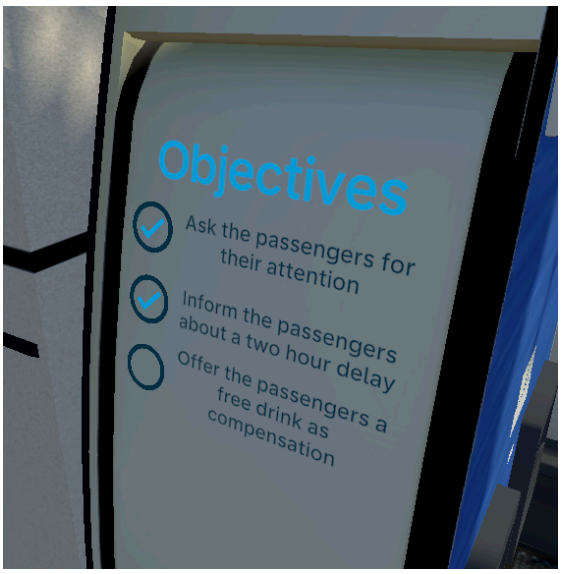

Image 4. The objectives list

For the 'Bethesda' interaction, another menu is present in the virtual environment, the choice menu. As described in chapter 6's 'Which ways of interaction will be examined?'; the choice menu (see Image 5) is used by the participant to select their response to the passengers. It is located next to the objectives list (see Image 6).

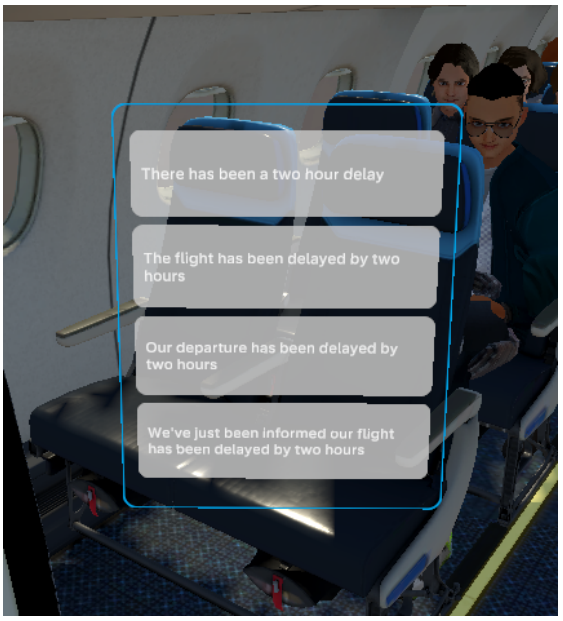

Image 5. The choice menu

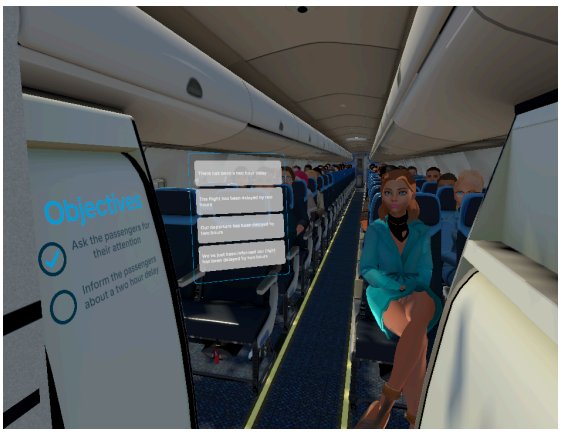

Image 6. Choice menu location

The passengers

The passengers can be divided into three sections. As described in chapter 6: ‘What does the physical test environment and the experiment application look like?’, rows 1 up to and including 5 are filled with animated passengers, rows 6 up to and including 10 are filled with static 3D floating heads and rows 11 up to and including 25 are filled with static 2D floating heads.

The animated passengers have a perpetual idle animation and can look at the participant or the window. The 3D and 2D floating heads are static.

The response from the passengers doesn’t come from any specific passenger, but instead an ethereal female voice.

The passengers with Inworld integration came with their own animations, while the ‘Bethesda’ passengers didn’t. Eliminating the difference between them didn’t fit within this paper’s leftover timeframe so the decision to make the passengers' behaviour as minimal as possible and the voice to be ethereal was made. This decision might influence the experiment’s results, but more on that can be found in appendix 3.

The scenario

The interaction scenario is the following; the flight has been delayed by two hours. The participant has to go out into the cabin to inform the passengers of the delay. They’re allowed to compensate the passengers with a free drink and snack for the discomfort.

The steps the participant must follow in this scenario are:

- Ask the passengers for their attention.
- Inform the passengers about a two-hour delay.
- Offer the passengers a free drink as compensation.
- Offer the passengers a free snack as compensation.

9. Conducting the experiment

The experiment was conducted as per the test plan (see appendix 3). This chapter will go into any deviations from the test plan and any additional information not previously discussed.

Participant information

The experiment was conducted with a total of 20 volunteers with different backgrounds and occupations. Out of these participants, 60 percent identified as male, 35 percent as female and 5 percent as other. Participant ages ranged from 15 to 62 years old, with an average of around 34 years old.

Highest education between participants ranged from VMBO-GT to university graduates and graduates with the title of master of sciences.

Out of the participants, 75 percent had already experienced VR at least once, 25 percent had no prior VR experience. In addition, all participants had flown on an airplane before.

Participants all had at least elementary English proficiency, with 95 percent having limited working proficiency or above.

Lastly, three participants claimed to have (a type of) social anxiety or stage fright. Out of these participants, all three find it difficult to speak to large groups of people.

Deviations

The main deviation from the test plan is that around 50 percent of the participants stood during the experiment, while the other 50 percent sat down. This was due to some technical difficulties. While setting up the test a

technical difficulty arose with setting up the VR headset's airlink. As there was no cable present that was long enough to connect the laptop with the VR headset while the participants were standing, the experiment had to be conducted while sitting.

Another deviation was with the test environment. To minimize the risk of outside interference impacting test results (for more, see appendix 3: Risks and Mitigations), the environment in which the experiment was conducted had minimal outside interference. With most, the experiment was conducted one on one in a closed environment, however, this wasn't possible in some cases. In these cases, an effort was made to minimize interference by removing most other sources of distraction, simulating the closed environment in which the experiment was conducted with most participants. When an outside source or a technical issue did cause the participant to be distracted, the step of the experiment where the issue occurred was redone.

Lastly, due to some technical limitations in the experiment application, the voice that speaks to the participant differs between interactions. However, they are as similar as possible.

10. Experiment results

'Bethesda interaction' results

The results from the 'Interaction' questionnaire (see appendix 5) show the realism score for the 'Bethesda' interaction averaged at 5,65 on a scale of 1 to 10. This number was calculated using the data gathered from question 1 of the 'Interaction' questionnaire (see figure 3 and appendix 5). It indicates average, if not lacking realism, and the conclusion can be made that participants were not very immersed.

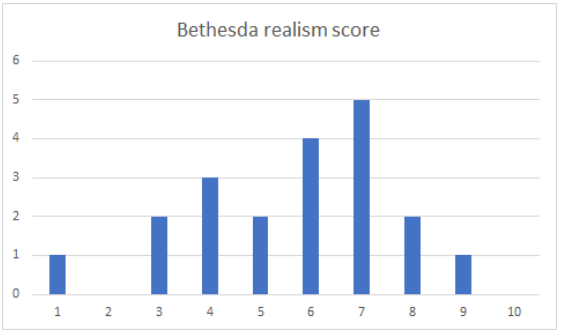

Figure 3. 'Bethesda' interaction realism score

The questionnaire results also show the NPC response time score for the 'Bethesda' interaction averaged at 7,65 on a scale of 1 to 10. This number was calculated using the data gathered from question 3 of the 'Interaction' questionnaire (see figure 4 and appendix 5). Also, the NPC response time opinion averaged at 3,95 on a scale of 1 to 5, which was calculated using the data gathered from question 4 of the 'Interaction' questionnaire (see figure 5 and appendix 5).

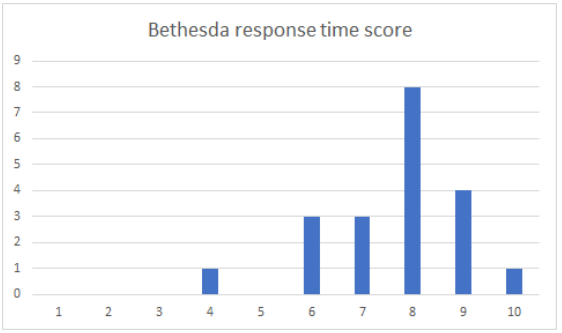

Figure 4. 'Bethesda' interaction NPC response time score

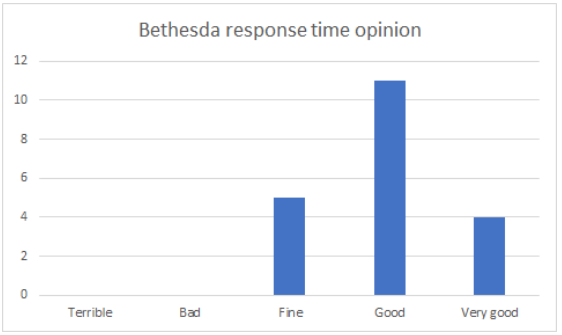

Figure 5. 'Bethesda' interaction NPC response time opinion

As for the more specific questions about the interaction itself, according to the first question in the 'Bethesda interaction'

questionnaire (see appendix 5), sixty-five percent of the participants stated that having to interact using the user interface negatively influenced how realistic the interaction felt to them (see figure 6). When counting negatives as -1, positives as +1 and none as 0, the resulting score would be -11, which is quite bad and clearly shows the negative impact of interacting using a user interface.

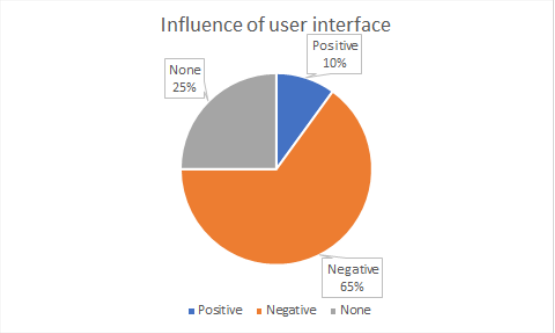

Figure 6. Influence of interacting using a user interface

Additionally, according to the second question in the 'Bethesda interaction' questionnaire (see appendix 5), having to choose between four answers also negatively impacted the experienced realism, as sixty percent of the participants stated as such (see figure 7). Using the same scoring system as before, the resulting score would be -7, showing having to choose between four answers lowers experienced realism.

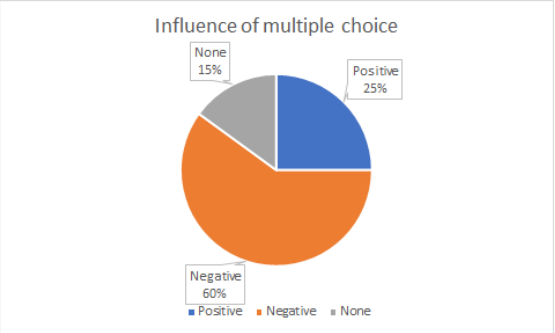

Figure 7. Influence of choosing between four predetermined answers during an interaction

Lastly, regarding choosing between four predetermined answers, according to the third question in the 'Bethesda interaction' questionnaire (see appendix 5), fifty percent of participants stated wanting to have chosen a different answer than shown.

This result could be due to two things, one of which being wanting to rephrase one of the options and the other being; not agreeing with the action that should be taken as directed by the scenario.

Eight participants found one or more steps in the scenario unusual, out of which 5 participants stated they found the answers or the flow of the conversation to be unusual. Out of these, 2 participants stated they wanted to rephrase one of the options, and 3 participants stated they did not agree with the action that should be taken as directed by the scenario.

To summarize, out of the eight participants who wanted to react differently, only two wanted to do so due to wanting to rephrase one of the options and three didn't agree with the action that should be taken as directed by the scenario, meaning five could be attributed to bad scenario writing, and not to the concept of multiple choice.

'Inworld interaction' results

The realism score for the 'Inworld' interaction turned out higher than the 'Bethesda' interaction's realism score, with an average of 7,05 on a scale of 1 to 10. This number was calculated using the data gathered from question 1 of the 'Interaction' questionnaire (see figure 8 and appendix 5).

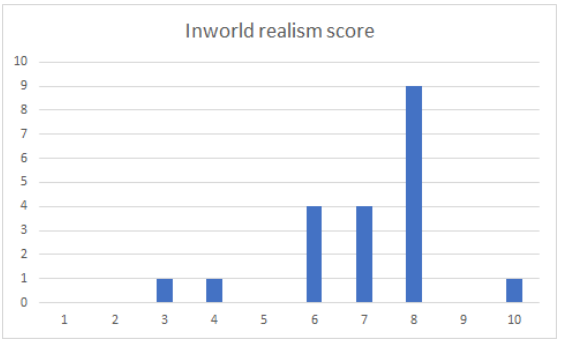

Figure 8. 'Inworld' interaction realism score

The NPC response time score for the 'Inworld' interaction on the other hand, is lower than that of the 'Bethesda' interaction, averaging at a 7 on a scale of 1 to 10. This number was calculated using the data gathered from question 3 of the 'Interaction' questionnaire (see figure 9 and appendix 5). Additionally, the NPC response time opinion also averaged lower, at 3,7 on a scale of 1 to

5, which was calculated using the data gathered from question 4 of the 'Interaction' questionnaire (see figure 10 and appendix 5).

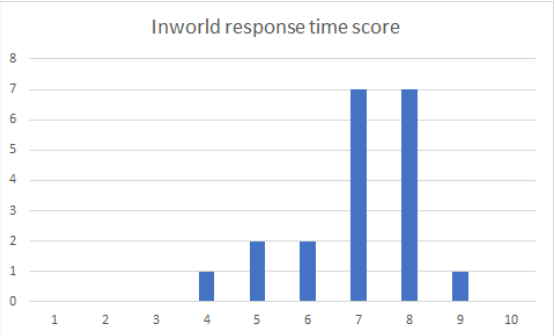

Figure 9. 'Inworld' interaction NPC response time score

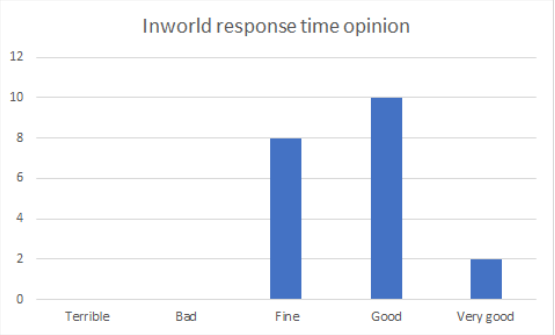

Figure 10. 'Inworld' interaction NPC response time opinion

Looking at the more specific questions about the interaction itself, according to the first question in the 'Inworld interaction' questionnaire (see appendix 5), ninety percent of the participants stated that speaking out loud to interact with the passengers positively influenced how realistic the interaction felt to them (see figure 11). When using the same method of counting negatives as -1, positives as +1 and none as 0, the resulting score would be 18, showing the influence of speaking out loud was overwhelmingly positive.

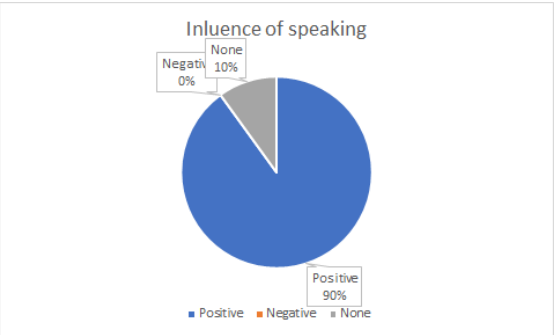

Figure 11. Influence of interaction by speaking out loud

Additionally, according to the second question in the 'Inworld interaction' questionnaire (see appendix 5), having to think up and formulate your own responses had equal positive influence, with ninety percent of participants stating it had a positive impact on how realistic the interaction felt to them (see figure 12).

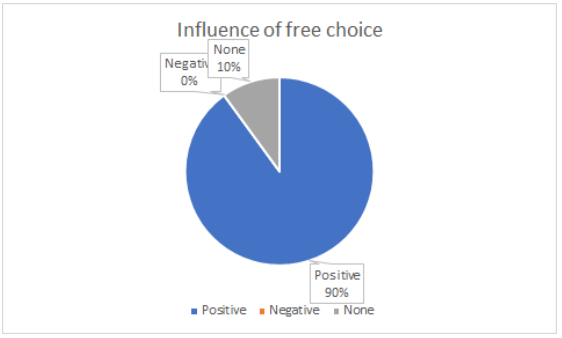

Figure 12. Influence of free choice

Now the question could be asked; "*was realism increased by the addition of free choice and formulation of responses or by the addition of an LLM?".* More on this question later.

Result comparison

Comparing the average 'Bethesda' and 'Inworld' interaction realism scores on a scale of 1 to 10 shows the 'Inworld' interaction to have been experienced as more realistic (see figure 13).

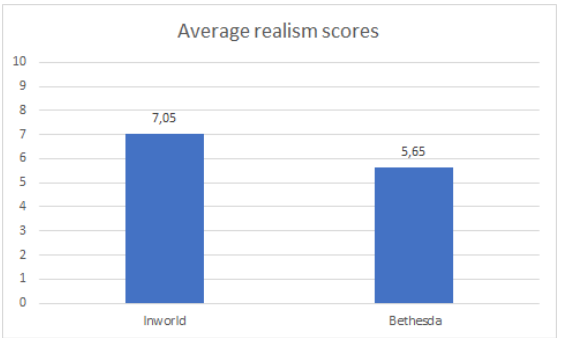

Figure 13. Average realism scores

When taking a closer look, eleven participants scored the experienced realism of the 'Inworld' interaction higher than that of the 'Bethesda' interaction (see figure 14).

These eleven participants scored the experienced realism of the ‘Inworld’ interaction, on average, 3,09 higher than the ‘Bethesda’ interaction. Meanwhile, the five participants who scored the experienced realism of the ‘Bethesda’ interaction higher scored it, on average, 1,02 higher.

Using these numbers, the conclusion can be drawn that on average, the ‘Inworld’ interaction will be perceived as more realistic and when it does, the experienced realism is higher than it would be if the ‘Bethesda’ interaction would be perceived as more realistic.

In simple terms, if a random person would be asked which interaction is more realistic, they are more likely to pick the ‘Inworld’ interaction and they would, on average, score it’s realism higher than the ‘Bethesda’ interaction.

Figure 14. ‘Inworld’ realism score compared to ‘Bethesda’ score

Seven participants stated they experienced unrealistic behaviour during one or both of the interactions (see figure 15).

The four participants who experienced unrealistic behaviour during the ‘Inworld’ interaction scored the experienced realism 6,75 on average. Those who did not experience any unrealistic behaviour during the ‘Inworld’ interaction scored it 7,13 on average.

The four participants who experienced unrealistic behaviour during the ‘Bethesda’ interaction scored the experienced realism a 5 on average. Those who did not experience any unrealistic behaviour during the ‘Bethesda’ interaction scored it 5,81 on average.

The single participant who experienced unrealistic behaviour during both interactions gave the experienced realism of both interactions an equal score.

Using these numbers, the conclusion can be made that when someone recognizes behaviour to be unrealistic, they find the entire interaction to be less realistic.

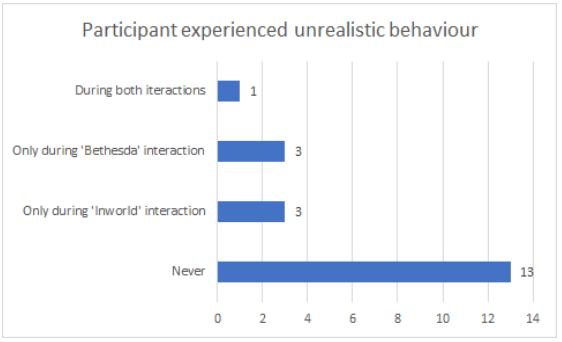

Figure 15. Amount of participants who experienced unrealistic behaviour

Looking at NPC response time scores next, the ‘Inworld’ interaction averages at 7 and the ‘Bethesda’ interaction averages at 7,65 (see figure 16).

The ‘Bethesda’ interaction scores higher on average, but when comparing individual response time- and realism scores, eight participants score the ‘Inworld’ interaction as more realistic, despite giving the ‘Bethesda’ interaction a higher response time score (see figure 17).

The two participants who score the ‘Bethesda’ interaction as more realistic both scored the response time equal for both interactions.

Using this, the conclusion can be drawn that at this level of difference in NPC response time, it does not affect the experienced realism of the interaction, as despite the ‘Bethesda’ interaction’s higher average response time score in many cases the ‘Inworld’ interaction still received the highest realism score.

This conclusion is also reinforced by the previously calculated average response time opinions. The average for both interactions were rounded up to “Good” from 3,95 for the ‘Bethesda’ interaction and 3,7 for the ‘Inworld’ interaction.

The participants found the NPC response times for both interactions to be good, however to support this conclusion, the influence of larger differences in response

times would have to be researched further in future works.

In a future work researching how response times impact experienced realism it would be important to see what the effects of both ends of the spectrum are. For example, when an NPC reacts immediately after you stop talking, would that be considered realistic? Or maybe it takes a minute to react, would that be considered realistic? Where is the sweet spot of response time where it is experienced to be most realistic.

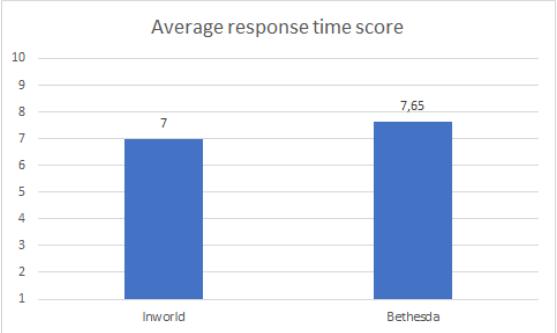

Figure 16. Average NPC response time scores

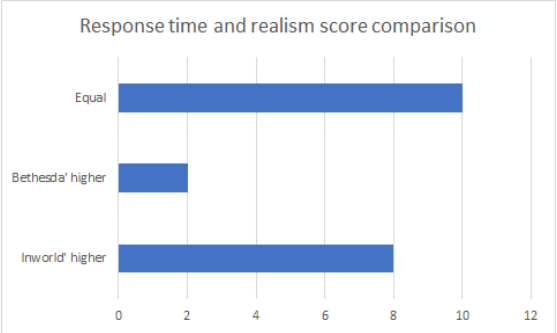

Figure 17. Realism score compared to NPC response time score

Observations and feedback

After the experiment, participants were allowed to give feedback on the experiment. The most prominent subject of feedback were missing elements that participants expect to improve realism.

Passengers speaking to each other, both before and during the training is a frequent topic. First of all, when in an environment with a lot of people, the expectation is that those people would speak amongst each other. This would create an atmosphere that's more realistic in comparison to the test where no passengers speak except for the responses.

Another participant added upon this topic that when responding to the trainee, passengers should also react to each other. For example, passengers who share the same opinion should back each other up and voice their opinion together, or vice versa. This would also place more emphasis on the feeling that the trainee is speaking to a group, instead of an individual.

In addition, improving physical character behaviour, such as improving animations and randomizing animations would also improve the experienced realism, according to multiple participants.

Lastly, one participant stated the responses from the passenger during the 'Inworld' interaction were too perfect. Many people stumble over their words or use uhs and ahs, during their sentences, while the passenger did not. This made the interaction feel unusual for them.

Another topic of feedback was the choice menu present in the 'Bethesda' interaction. Multiple participants found this menu to be immersion breaking and made the experience feel unrealistic.

To the contrary, having to formulate your own sentences and having to think of what to say during the 'Inworld' interaction improved realism a lot, according to multiple participants. It added an additional degree of difficulty, often only found in real social context.

Lastly, some feedback was given in relation to the questionnaires. Additionally, some observations about these subjects were also made.

The questionnaire included some questions which participants interpreted differently, for example, the question about whether or not the participant enjoyed the interaction with the passengers. This question was ambiguous, as some passengers found the passenger annoying, while others found the scenario of a bad news speech to be unenjoyable.

The question about whether the experience could help the participant to get used to speaking to large groups was often interpreted differently. Some would answer

whether or not it could help ‘them’ specifically, while others would imagine whether or not it could help ‘someone’. This contaminated the answers which is also why they were excluded from the results. This was also the case for the previously discussed question.

Conclusion

Non-player characters play multiple roles and are divided into multiple types based on the roles they play.

Each type of NPC has different criteria for them to be perceived as believable. For the passenger NPCs in the KLM’s VR communication training, these criteria include being able to deceive the player, convey values from the historical column of the game agent matrix and that the NPC should be perceived as able to exist within their own social environment.

Large language models can simulate human language and bring emergent behaviour to an NPC, but they do not meet all criteria to be perceived as believable. However, NPCs might still benefit from the introduction of this emergent behaviour.

An experiment where an NPC whose responses are generated using an LLM is compared to an NPC whose responses are predetermined has revealed a couple of positive and negative influences on the experienced realism of an interaction with an NPC.

The biggest positive influence on experienced realism is being able to speak to NPCs. Ninety percent of participants stated it had a positive influence on how realistic the experience of interacting with an NPC felt to them.

In addition, being able to choose freely what to say, instead of having to choose between multiple options also had a positive influence on how realistic the experience felt for ninety percent of the participants.

The biggest negative influence on experienced realism was the exact opposite of the positive influences, namely having to choose one of four options using a user interface, with sixty percent of the participants stating as such.

Additionally, sixty-five percent of the participants stated using a user interface to interact with an NPC had a negative influence on the experienced realism.

Unrealistic NPC behaviour also affected experienced realism, as when unrealistic NPC behaviour is recognized, the entire interaction is felt to be less realistic.

The additional time it takes for the LLM to respond does not affect the experienced realism in a significant way. However, as this experiment did not include the subject of how differing NPC response times affect experienced realism, further research will be necessary to support this claim.

Concluding, the results show that if a random person would be asked which interaction is more realistic, they are more likely to pick the ‘Inworld’ interaction and they would, on average, score it’s realism higher than the ‘Bethesda’ interaction and thus integrating LLMs (using the inworld unitypackage) into NPCs can improve how realistic an interaction with an NPC feels, however, whether this is due to the LLM of the addition of free speech and choice of what to say is to be researched further.

Recommendation

Looking at the performance of the LLM when integrated into an NPC, it can improve the experienced realism of the training. However, looking at the bigger picture, the usage fee for third party LLMs and confidentiality requirements could make the utilizing LLMs in products made by the KLM XRCoE difficult.

The Inworld package used for this experiment provides good results, but confidentiality is not guaranteed, on the other hand, a first party solution would potentially take a long time to produce.

Another point to be discussed is the question asked in chapter 10: ‘Inworld interaction’ results. Was the resulting increase in realism score brought on by the introduction of the LLM or by allowing the trainee to speak? My (author) hypothesis would be that the increase in realism was caused by the introduction of free speech instead of the LLM.

Implementing speech recognition without the use of an LLM could possibly bring upon similar results, however this would have to be researched further.

In conclusion, the recommendation that I would make to KLM is to sort out whether using third party LLMs lies within KLM regulation and budget. If so, I would recommend they start experimenting with the Inworld package and see if it suits their needs, as implementing LLMs into their NPCs will be able to bring improvement to those NPCs.

References

- Bartle, A. (2004). Designing virtual worlds [Research Gate]. In *Google Books*. New Riders. https://www.researchgate.net/publication/200025892_Designing_Virtual_Worlds (Original work published 2003)

- Brockman, G., Eleti, A., Georges, E., Jang, J., Kilpatrick, L., Lim, R., Miller, L., & Pokrass, M. (2023, March 1). *Introducing ChatGPT and Whisper APIs*. OpenAI. https://openai.com/blog/introducing-chatgpt-and-whisper-apis

- Choi, M., Pei, J., Kumar, S., & Shu, C. (2023). Do LLMs understand social knowledge? Evaluating the sociability of large language models with the SocKET benchmark. In *arXiv* [Journal-article]. https://arxiv.org/pdf/2305.14938.pdf

- De Melo, C. M., & Gratch, J. (2015). Beyond believability: quantifying the differences between real and virtual humans [Research Gate]. In *Intelligent Virtual Agents* (pp. 109–118). Springer. https://doi.org/10.1007/978-3-319-21996-7_11

- Google. (n.d.). *Neural Networks API*. Android Developers. https://developer.android.com/ndk/guides/neuralnetworks

- Inworld. (n.d.). *Inworld: AI-powered gameplay*. https://inworld.ai/

- Loyall, A. (1997). *Believable Agents: building interactive personalities* [Thesis]. Carnegie Mellon University. https://www.cs.cmu.edu/Groups/oz/papers/CMU-CS-97-123.pdf

- NPC. (n.d.). In *Cambridge Dictionary*. https://dictionary.cambridge.org/dictionary/english/npc

- Togelius, J., Yannakakis, G. N., Karakovskiy, S., & Shaker, N. (2012). Assessing believability [Research Gate]. In *Believable Bots* (pp. 215–230). Springer. https://doi.org/10.1007/978-3-642-32323-2_9

- Undreamai. (n.d.). *GitHub - undreamai/LLMUnity: Create characters in Unity with LLMs!* GitHub. https://github.com/undreamai/LLMUnity

- Unity Technologies. (n.d.). *Unity sentis*. Unity. https://unity.com/products/sentis

- Volum, R., Rao, S., Xu, M., DesGarennes, G., Brockett, C., Van Durme, B., Deng, O., Malhotra, A., & Dolan, B. (2022). Craft an Iron Sword: dynamically generating interactive game characters by prompting large language models tuned on code. *Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022)*, 25–43. https://doi.org/10.18653/v1/2022.wordplay-1.3

- Wages, R., Grünvogel, S. M., & Grützmacher, B. (2004). How realistic is realism? Considerations on the aesthetics of computer games [Research Gate]. In *Entertainment Computing - ICEC 2004* (pp. 216–225). Springer. https://doi.org/10.1007/978-3-540-28643-1_28

- Warpefelt, H., & Verhagen, H. (2017). A model of non-player character believability. *Journal of Gaming & Virtual Worlds*, *9*(1), 39–53. https://doi.org/10.1386/jgvw.9.1.39_1

- *What are Large Language Models?* | *NVIDIA Glossary*. (n.d.). NVIDIA. https://www.nvidia.com/en-us/glossary/large-language-models/

Appendices

1. A model of non-player character believability - Results (Warpefelt and Verhagen, 2017)

The results of this study will be described in the following sections, which contain short descriptions of the types in the typology. These descriptions are based on the type descriptions provided by Warpefelt and Verhagen (2016). Furthermore, they contain the GAM based descriptions of the behavior of NPCs of each type, and how this behavior affects believability.

Vendor
Vendors are NPC that that buy and sell items. This type NPC will generally be static within an area. Must exhibit behaviors from the Single Agent/Act cell in the GAM, but more advanced cases can also exhibit Memory of previous interactions and Face to face from the Multiple Agents/Interact cell. In some cases they will also exhibit the Advertising behavior from the Cultural Historical/React cell.

Services
Services are NPCs that provide some sort of service to the player, for example item repairs or transportations. NPCs of this type are often also Vendors. Their behavior is identical to that of Vendors.

Questgiver
These NPCs provide the player with missions and rewards within the game. Their behavior is identical to that of Vendor and Services NPCs, with the exception that they may roam over larger areas.

Enemy
As indicated by the name, these are NPCs that fight the player. They are overall capable of portraying behaviors from the Single Agent column in the GAM, and can convincingly portray at least some of the behaviors such as Mob action and Cooperation from the Social Structural and Social Goals columns. They are generally capable of portraying behaviors from the Multiple Agents/React cell, but often fail to convincingly portray these behaviors when the social context calls for more adaptability, due to limitations in their capability to portray Adaption. This often causes cascading failures across the GAM. Furthermore, they also often fail to actively portray Acquires information and Awareness from the Single Agent/React cell, which can lead to NPC seeming omniscient or oblivious. NPCs of the Boss subtype generally exhibit the same strengths and weaknesses as regular Enemy NPCs, but can often enact more complicated strategies or have more powerful in-game abilities.

Opponent
Opponents are the non-violent enemies found in games. They hinder the player in various ways, for example by blocking moves or chasing them. Overall, the behavior of this type is very similar to the Enemy type in terms of both strengths and weaknesses.

A special case for this type of NPC is the Manipulator subtype. These NPCs exist to socially manipulate the player into performing certain actions. They generally require a higher degree of social capability than their Opponent siblings, and must be able to convincingly portray values from the Cultural Historical column.

Sidekick
Sidekicks are NPCs who accompany the player and provide them with help that is not directly linked to fighting enemies, for example advice, directions, or resources. They will react to their environment and other entities in the game, and thus need to portray behaviors associated with the React/Multiple Agents cell. They must also portray the value Team player from the Interact/Social Goals cell.

Ally
Ally NPCs are the friendly version of Enemy NPCs. They fight alongside the player, but the player cannot control them. They largely have the same requirements as the Enemy NPCs. Their social capability can also be strengthened by making them capable of portraying values such as Clan wars, Cooperation and Team Player from the Interact/Social Goals cell. This is, however, optional.

Companion
Companions are NPC that accompany the player as they play the game. They are functionally similar to the Ally type, but they can be controlled by the player. They are generally as capable as Ally NPCs, but are often persistently present in the game and therefor require that the aforementioned values from the Interact/Social Goals cell actually be portrayed.

Pet
Pets are decorative NPCs that follow the player around. They do not interact with the world, and will only exhibit behaviors related to the Single agent/Act cell.

Minion
NPCs of the Minion type are usually found in strategy games, where they take on the role semi-autonomous disposable avatars of the player. Their behavior is similar to that of Enemy and Ally NPCs, and generally has the same requirements for believability.

Storyteller
This is a type of NPC that provides narrative exposition. They only need to be able to perform behaviors from the Single Agent/Act cell in order to be perceived as believable.

2. How should the passenger character(s) behave to achieve the desired result? - Expert interview transcription

Question:
What is the purpose of the training, for which situation is the trainee being prepared?
Answer:
The purpose of the training is to prepare pilots/candidates for effectively providing information to a group of passengers.

Question:
What aspects are being focused on with the trainee, which points are important?
Answer:
The focus is on where improvements can be made, for example:

- Is the message getting across?
- Does the message contain relevant information?
- Is the goal of the message being achieved (calming, informing, welcoming)?
- Is the verbal congruent with the non-verbal?

Question:
How is the training conducted?
Answer:
The training takes place in a safe training environment such as a classroom.

Question:
What role do the passengers play in the training, and how does their behavior impact the training's objective?
Answer:
Behavior triggers behavior, so how passengers behave will have an effect on the candidate/pilot. Can the pilot distill the right information from it and manage a situation in such a way that their message remains clear? It's an inoculation before the pilot stands in front of passengers in real life.

Question:
How should passengers behave during the training, as in any possible real-life situation, or somewhat nuanced?
Answer:
In essence, passengers react in roughly the same way; after all, situations like these (where a pilot has to address a group) usually involve delivering bad news. Passengers may behave angrily, relieved, saddened, interested, etc. Of course, the training's value increases as the passengers' reactions become more nuanced.

Question:
To what extent should passengers behave inappropriately to achieve the desired effect on the trainee? Does it need to go as far as personal attacks, or is the goal already achieved with small signs of irritation/aggression?
Answer:
This concerns creating a realistic representation of how passengers might behave during a PA. There are certain boundaries to consider. Personal attacks are unlikely within this context.

Question:
What type of behavior from a passenger has the most impact on the training objective, vocal or physical, and in what way is this represented, a vocal response, no vocal response, is the passenger preoccupied with something else or distracted?
Answer:
The objective must be achieved within certain boundaries within which the pilot can operate. So the situation must be managed in such a way that the message gets across. In addition, it may be necessary to call a passenger to order after a disruptive verbal response, but that doesn't happen often. A physical reaction, other than body language, basically falls outside the framework in which we operate.

Question:
What factors must the situation and environment meet in order to achieve the training objective? Should the training be conducted in different situations, for

example, a large or small aircraft, many or few passengers?
Answer:
The training should approximate reality as closely as possible. Although every first step you take can already assist a pilot with preparation (such as the product currently does), the product can be improved to increase training value. This improvement should be viewed within the framework in which KLM operates. So, PA's (Public Announcements) in the cabin of KLM aircraft or in a gate area are the two most common places (besides the cockpit) where pilots make PA's. The number of passengers, whether many or few, can also help create a realistic scenario. For example, an Embraer has fewer passengers than a 777.

Question:
How should the trainee feel during the initial training and how during the final one, what feeling should the trainee have after completing their final training?
Answer:
A trainee should feel the added value of the tool, otherwise it won't be used anymore. A small degree of discomfort is acceptable as long as it's within a safe environment. At the end of the training session(s), a trainee should feel competent enough to perform the exercise in real life.

Question:
Are changes made to the training for each trainee based on their weaknesses, and how are these identified and taken into account, what changes are made to the training?
Answer:
This is definitely a possibility with this tool. For example, introducing background noise or passengers trying to distract the trainee by being uninterested or angry.

Question:
Is the trainee exposed to one type of passenger, the average passenger, or to different ones? In the case of different passengers, is this done within one training session or multiple?
Answer:
The average one. However, the difficulty can be increased to enhance the training value.

Question:
Are ‘standard’ scenarios prepared for the training, or are they always improvised on the spot? In that case, on what basis are they formulated?
Answer:
The best approach is to find a situation that the trainee has experienced or wants to practice because they have seen it with someone else. That's where they derive the most training value from.

3. Experiment for assessing a non-player character's perceived believability after integrating a large language model - Test plan

Test Plan: Assessing the perceived believability of two interactions between participant and non-player character in a virtual reality environment.

Objective:

The objective of this test plan is to assess the level of perceived believability of two interactions between a participant and non-player character (NPC) in a virtual reality (VR) environment, resulting in data which will be used in a recommendation on whether or not large language model (LLM) implementation can improved NPC unpredictability and autonomy, aiming to increase realism.

Testcriteria:

1. **Perceived Believability:** How realistic and believable the interaction with NPCs feels to the user.
2. **Immersion:** The extent to which the user feels immersed in the virtual environment during the interaction.
3. **Naturalness of responses:** The naturalness of the NPC's responses and behaviours during the interaction.
4. **User Engagement:** The level of engagement and interest elicited from the user during the interaction.
5. **Ease of Use:** The ease with which users can initiate and maintain interaction with NPCs.
6. **Technical Performance:** Any technical issues encountered during the interaction, such as lag, glitches, or visual artifacts.

Test Environment:

- VR Headset: Meta Quest 2
- VR Controllers: None (Hand Tracking)
- Hardware Specifications:
    - PC specifications:
        - Device name: Acer Predator Triton 500 PT515-51
        - CPU: i7-9750H
        - GPU: RTX 2060 (laptop); DirectX-version: 12 (FL 12.1)
        - Memory: 16GB
        - Charger: Active

Test Scenarios:

1. **Scenario 1: 'Bethesda' Interaction**
    - Description: Participant interacts with the passengers using a user interface showing four response options. The passengers will respond with a predetermined auditory response.
    - Tasks:
        - Ask the passengers for their attention.
        - Inform the passengers about a two hour delay.
        - Offer the passengers a free drink as compensation
        - Offer the passengers a free snack as compensation
    - Metrics: Perceived believability, immersion, naturalness of responses, user engagement, ease of use, technical performance.
2. **Scenario 2: 'Inworld' Interaction**
    - Description: Participant interacts with the passengers through speech. The passengers will respond with an auditory response generated by a large language model.
    - Tasks:
        - Ask the passengers for their attention.

        - Inform the passengers about a two hour delay.
        - Offer the passengers a free drink as compensation.
        - Offer the passengers a free snack as compensation.
    - Metrics: Perceived believability, immersion, naturalness of responses, user engagement, ease of use, technical performance.

Test Procedure:

1. **Preparation:**
    - Set up the laptop, plug in the charger and open the Unity project with the experiment application.
    - Set up the VR headset and connect it to the laptop through a link cable.
2. **Participant briefing:**
    - Introduce the experiment and explain their goal.
    - Introduce the participant to the virtual environment.
    - Instruct the participant on how to do the ‘Bethesda’ and ‘Inworld’ interactions.
3. **‘Participant information’ questionnaire:**
    - The participant fills in the ‘Participant information’ questionnaire.
4. **Experiment execution; interaction 1:**
    - Conduct the experiment with a randomly chosen interaction, ensuring participants complete all tasks.
    - Monitor participant interactions and take notes on observations.
5. **First interaction questionnaire:**
    - The participant fills in the first interaction’s respective questionnaire.
6. **Experiment execution; interaction 2:**
    - Conduct the experiment with the other interaction, ensuring participants complete all tasks.
    - Monitor participant interactions and take notes on observations.
7. **Second interaction questionnaire:**
    - The participant fills in the second interaction’s respective questionnaire
    - Record user feedback.
8. **Data compilation:**
    - Compile recorded data into easily readable format.

Test Team:

- Test Lead: Chris Huider
- Testers:
    - Theo Puijk
    - Nyla Slot
    - Tim van Vliet
    - Joep Klein Teeselink
    - Maarten Laken
    - Daisy Navarette
    - Ian -
    - Leon Bierling
    - Shane de Hundt
    - Dave Schokker
    - Freek Schokker
    - Stef Keuken
    - Anna Keuken
    - Elsa -
    - Jeannette Janssen
    - Eva Blufpand
    - Roberto Blufpand
    - Daniel de Luca
    - Annabel Boriglione
    - Karel Kiers

Timeline:

- Test Preparation: 05-02-2024 - 30-04-2024
- Test Execution: 30-04-2024 - 13-05-2024
- Analysis and Reporting: 13-05-2024 - 28-05-2024

Risks and Mitigations:

- Risk: Technical issues may impact the reliability of test results.
    - Mitigation: Conduct thorough pre-testing to identify and

address any technical issues before formal testing begins.

- Risk: Additional differences besides the implementation of an LLM in the NPCs may impact the reliability of test results.
    - Mitigation: Minimize the differences between scenarios outside the LLM implementation and address any differences which haven't been removed during data compilation.
- Risk: Outside interference may impact the reliability of test results.
    - Mitigation: Minimize outside interferences in order to not influence participant immersion and reduce the risk of the LLM picking up sentences not originating from the participant.

4. Experiment for assessing a non-player character's perceived believability after integrating a large language model - 'Participant information' questionnaire

1. Name [open] (optional)
2. Gender [open]
3. Age [open]
4. Highest education [open]
5. Have you ever experienced virtual reality before? [yes/no]
6. Have you ever flown/been on an airplane before? [yes/no]
7. How fluent is your English? [no proficiency/elementary proficiency/limited working proficiency/full professional proficiency/primary fluency – bilingual]
8. Do you have (a form of) social anxiety/stage fright? [yes/no/maybe]
    a. Do you find it difficult to speak to big groups? [yes/no]

5. Experiment for assessing a non-player character's perceived believability after integrating a large language model - 'Interaction' questionnaire

1. How realistic was the interaction with the passengers? [0/10]
2. Did the passengers ever respond in a way you found unrealistic? [yes/no]
3. How realistic was the passengers' response time? [0/10]
4. What is your opinion on the passengers' response time? Was it: [terrible/bad/fine/good/very good]
5. Did you enjoy the interaction with the passengers? [yes/no]
6. Was interacting with the passengers intimidating? [yes/no]
7. Do you think this experience would help you to get used to speaking to large groups? [yes/no/maybe]
8. How comfortable was the virtual reality headset? [0/10]
9. Did the virtual reality headset distract you in any way? [yes/no]
10. How comfortable was having to stand during the test? [0/10]
11. Was having to stand during the test distracting you in any way? [yes/no]
12. How realistic was the scenario? [0/10]
13. Did you think of any of the steps in the scenario as unusual? [yes/no]
    a. What about the scenario did you find unusual? [open]
14. How realistic was the passengers' appearance? [0/10]
15. Did the appearance of the passengers influence how immersive the experience felt to you? [yes positively/yes negatively/no]
16. Did you notice the passengers all behaving identically? [yes/no]
17. Did the passengers behaving identically influence how immersive the experience felt to you? [yes positively/yes negatively/no]
18. Did you notice the environment outside of the airplane is empty? [yes/no]
19. Did you notice the fact that the passenger responses came from an ethereal voice? [yes/no]
20. Did the passenger responses coming from an ethereal voice influence how immersive the experience felt to you? [yes positively/yes negatively/no]

Depending on the interaction:

'Bethesda interaction':

1. Did having to interact using a user interface influence how realistic the interaction felt to you? [yes positively/yes negatively/no]
2. Did having to choose between multiple responses influence how realistic the interaction felt to you? [yes positively/yes negatively/no]
3. Did you ever want to respond differently to the provided choices? [yes positively/yes negatively/no]

'Inworld interaction':

1. Did having to interact by speaking out loud influence how realistic the interaction felt to you? [yes positively/yes negatively/no]
2. Did the free choice of how you responded influence how realistic the interaction felt to you? [yes positively/yes negatively/no]