

公立はこだて未来大学 2023 年度 システム情報科学実習  
グループ報告書

Future University Hakodate 2023 Systems Information Science Practice  
Group Report

プロジェクト名

メタバース・DE・アバター

Project Name

METAVERSE DE AVATAR

グループ名

グループ A,B

Group Name

Group A,B

プロジェクト番号/Project No.

20

プロジェクトリーダー/Project Leader

出田光 Ideta Hikari

グループリーダー/Group Leader

齋藤天馬, 横山直矢 Saito Tenma, Yokoyama Naoya

グループメンバ/Group Member

白石侃穂 Shiraiishi Kanno

中島一侃 Nakashima Kazunao

金仁洙 Kin Insu

大原拓己 Ohara Takumi

長瀬カイセイ Nagase Kaisei

沼田晴登 Numata Haruto

上野大道 Uwano Daido

西村光陽 Nishimura Koyo

指導教員

角薫 元木環 長田純一

Advisor

Sumi Kaoru Motoki Tamaki Osada Junichi

提出日

2023 年 7 月 21 日

Date of Submission

July 21, 2023



## 概要

メタバースとは、ギリシャ語で”超越した”という意味を持つ「メタ」と、”世界”を意味する「バース」を組み合わせて作られた言葉である。パソコンや電子機器の中の仮想空間上に作られた世界そのものを表している。人々はその仮想空間上に自分の分身となるアバターを登場させ操作することで、現実ではできない様々な体験をすることが出来る。また、対話型 AI とは近年爆発的に発展してきた人間らしい返答をし、人間と対話することを可能にした AI のことである。本プロジェクトではメタバースとアバター、そして対話型 AI が人間のコミュニケーションに及ぼす影響について探究する。本プロジェクトは独自のメタバースとアバター、対話型 AI を 1 から開発し、その性質を深く理解し考察を試みる。

**キーワード**   メタバース, アバター, AI

(※文責: 出田光)

# Abstract

Metaverse is a word created by combining the Greek words "meta," meaning "transcendent," and "berth," meaning "world." It represents the world itself created in the virtual space inside computers and electronic devices. By creating and operating an avatar, a person's alter ego, in this virtual space, people can have various experiences that are not possible in reality. Interactive AI is an AI that can respond in a human-like manner and interact with humans, a development that has been exploding in recent years. This project explores the impact of metaverse, avatar, and interactive AI on human communication. This project will develop its own metaverse, avatar, and interactive AI from scratch, and attempt to deeply understand and consider their properties.

**Keyword** Metaverse, avatar, AI

(※文責: Ideta Hikari)

# 目次

<b>第 1 章</b>	<b>はじめに</b>	<b>1</b>
1.1	本プロジェクトの目的	1
1.2	全体のデザインプロセス	1
<b>第 2 章</b>	<b>到達目標と課題の割り当て</b>	<b>2</b>
2.1	到達目標の設定	2
2.2	目標達成のためのプロセス	2
2.2.1	グループ A の目標とプロセス	2
2.2.2	グループ B の目標とプロセス	3
2.3	課題の割り当てと各チームの目標	3
2.3.1	メタバース空間の制作について	3
2.3.2	3DCG を用いたオブジェクトの制作について	4
2.3.3	対話型 AI の制作について	4
<b>第 3 章</b>	<b>前期の開発</b>	<b>6</b>
3.1	メタバース空間の制作について	6
3.1.1	制作過程	6
3.1.2	進捗	7
3.2	3DCG の制作について	8
3.2.1	制作過程	8
3.2.2	進捗	9
3.3	対話型 AI の制作について	12
3.3.1	制作過程	12
3.3.2	進捗	13
<b>第 4 章</b>	<b>前期における課題</b>	<b>14</b>
4.1	中間発表の反省	14
4.2	今後の課題と展望について	14
4.2.1	メタバース空間の制作について	14
4.2.2	3DCG の制作について	14
4.2.3	対話型 AI の制作について	15
<b>第 5 章</b>	<b>後期の開発</b>	<b>17</b>
5.1	メタバースの制作について	17
5.1.1	A グループ	18
5.1.2	B グループ	25
5.2	3DCG の制作について	27
5.2.1	A グループ	27
5.2.2	B グループ	39

5.3	対話型 AI の制作について . . . . .	41
5.3.1	制作過程 . . . . .	41
5.3.2	進捗 . . . . .	49
<b>第 6 章</b>	<b>成果発表</b>	<b>54</b>
6.1	概要 . . . . .	54
6.2	制作物 . . . . .	54
6.2.1	概要 . . . . .	54
6.2.2	コミュバース CM . . . . .	55
6.3	評価 . . . . .	60
<b>第 7 章</b>	<b>課外活動</b>	<b>61</b>
7.1	小学校訪問 . . . . .	61
7.1.1	自己紹介と授業の説明 . . . . .	61
7.1.2	AI についての講義 . . . . .	61
7.1.3	対話式 AI との会話体験 . . . . .	62
7.2	アカデミックリンク . . . . .	67
<b>第 8 章</b>	<b>まとめ</b>	<b>68</b>
8.1	プロジェクト全体の成果 . . . . .	68
8.2	各制作チームの成果 . . . . .	69
8.2.1	メタバースチーム . . . . .	69
8.2.2	3DCG チーム . . . . .	74
8.2.3	AI チーム . . . . .	75
8.3	その他の制作物 . . . . .	76
8.3.1	プロジェクトアイコンの作成 . . . . .	76
8.4	到達目標, および課題に対する反省 . . . . .	77
8.5	今後の展望 . . . . .	78
8.5.1	メタバースチーム . . . . .	78
8.5.2	3DCG チーム . . . . .	80
8.5.3	AI チーム . . . . .	81
<b>付録 A</b>	<b>中間発表で使った発表スライド</b>	<b>86</b>
<b>付録 B</b>	<b>中間発表に向けて制作したポスター</b>	<b>87</b>

# 第 1 章 はじめに

## 1.1 本プロジェクトの目的

近年、仮想現実の空間で、人々がアバターとなってお互いにコミュニケーションを取るメタバースに注目が集まっている。本プロジェクトは、そのメタバースと、メタバース上で人間の代わりに務めるアバター、そして人と対話の出来る人工知能を作成する。そして、人と人、または人と人工知能が仮想空間上でコミュニケーションを取ることで何が起こるか、どんなものが生まれるのかについて探究することを目的としている。

(※文責: 齋藤天馬)

## 1.2 全体のデザインプロセス

プロジェクトメンバーをメタバースチーム、3DCG チーム、AI チームに分けた後に、各チームを半数に分けそれぞれ A グループ、B グループに移動させた。各グループでブレインストーミングを行い、制作物の案を出した。次に制作物の UI をグループメンバーにイメージしてもらい、それをグループメンバー同士で絵や画像を使って共有した。次に各技術グループの使用ソフトに関するプレゼン大会と技術の勉強会を行った。ある程度の知識をつけたところで、どういった機能を実装するのかについて案出しを行い、実装したい機能に制作する優先順位をつけた(図 1.1)。続いて中間発表までの制作スケジュールを立て、それに基づいて制作を進めた。平行して本プロジェクトのロゴ制作と小学校との連携授業について議論し、六月中旬に連携授業を行う小学校に赴き連携授業の内容を説明した。続いて中間発表に向けたポスター制作、口頭説明用の原稿とスライド制作を行った。本プロジェクトの担当教員に発表のリハーサルを見ていただき意見をいただいた。

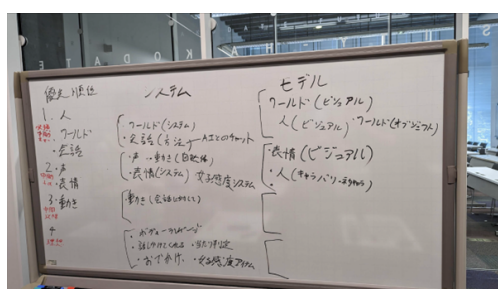


図 1.1 実装したい機能に優先順位をつけた際のホワイトボード

(※文責: 齋藤天馬)

## 第 2 章 到達目標と課題の割り当て

### 2.1 到達目標の設定

本プロジェクトの目的である、メタバース内で人と AI がどのようなコミュニケーションを取るかという議題に取り組むために、本プロジェクト独自のメタバース空間を作成することを目標とする。さらに、その空間上で人や AI がアバターに成り代わってコミュニケーションを取ることができるシステムと舞台を開発する。

(※文責: 横山直矢)

### 2.2 目標達成のためのプロセス

上記の目標達成のため、まずメタバースを開発する環境を調査した。ゲーム開発プラットフォームである Unity というソフトであれば、3DCG を用いた立体的な空間を作成し、そこにオブジェクトやギミックを自由に盛り込むことが可能である。よって、土台となる開発環境は Unity に決定した。次に、メタバース内の世界観を表現するために 3DCG でオブジェクトを制作する環境を blender というソフトに決定した。blender を使い慣れている制作メンバーが居たことから、すぐに制作に取り掛かれるという点で採用した。また、メタバース空間内で活動するアバターの制作は Vroid を使用することにした。そして、対話型 AI の開発は ChatGPT の API を使用し、Unity と連携させることで制作を進めることにした。さらに人と AI のコミュニケーションをより自然なものにするために、音声での発言の入出力を実現するため、VOICEVOX を使用することとした。これらの詳しい制作過程は各技術グループの項目にて後述する。

(※文責: 出田光)

#### 2.2.1 グループ A の目標とプロセス

グループ A は好感度システムを用いた人型 AI と没入感のある様々なワールド上でコミュニケーションを取れるようにすることを目標とする。プロジェクトの目標を達成するためにすべきこととして、まずはユーザーが没入できるワールドの構築が挙げられる。実現するために、オブジェクトに干渉できたり、質感が伝わってくるようなテクスチャをオブジェクトに添付する。さらに好感度システムを用いた人型 AI との対話を実現するために、ChatGPT に特定の行動をすることで好感度に変化する指示することで好感度システムを作成し、対話においても ChatGPT を用いることで円滑な AI との対話を可能にする。

(※文責: 齋藤天馬)



## 2.2.2 グループ B の目標とプロセス

グループ B はメタバース空間ならではのアクションを一つの要素として取り入れたワールドを作成し、世界観に没入した状態で AI との対話をするすることで、より人は AI に対して話しかける頻度が向上し、プロジェクトの目標に近づくことができると考えた。具体的には、VR ゴーグルに付属しているコントローラーのジャイロセンサーを活用し、主にコントローラーを振るなどの直感的なアクションを取り入れる。また、対話 AI についても、先述のアクションによって空間内で得られるアイテムを用いることで徐々に対話の精度が上がっていくという仕掛けを施す。これらの仕掛けと世界観によって、人は積極的に AI とコミュニケーションを取ることができるはずである。

(※文責: 横山直矢)

## 2.3 課題の割り当てと各チームの目標

メタバース空間の制作にあたり、開発部門を大きく以下の三つに分ける。

- ・世界観に合わせたメタバース空間、及び空間内のシステムを作成する部門
- ・空間内のオブジェクト、及びアバターの 3D モデルを作成する部門
- ・人と自然言語で対話することができる AI を作成する部門

なお、開発部門のことをチームと呼ぶこととする。

(※文責: 横山直矢)

### 2.3.1 メタバース空間の制作について

グループ A の目標はメニュー画面の作成、特定のアクションを引き起こすオブジェクトの設計、そして音声入力から文字を生成する機能の導入だ。加えて、対話型 AI とのインタラクションを可能にし、多様な世界観へワープできる機能をこの仮想空間に組み込むことだ。これらの目標をこなすことで、対話型 AI とのインタラクションを楽しみながら世界を転々とできるような仮想空間を作るという目的を達成したい。

(※文責: 齋藤天馬)

グループ B では「仮想空間やアバター利用ならではの体験ができるようにする」を目的として制作を進める。AI と対話をする際の抵抗感を緩和するためには、ゲームなどの没入しやすい環境を整えることが重要だと考え、グループの世界観に合わせた舞台設計をする。直感的な操作ができるようにオブジェクトの配置などを調整する。他チームが作成したオブジェクトや制作物を実際に仮想空間上で動かせるようにする。

(※文責: 沼田晴登)

### 2.3.2 3DCG を用いたオブジェクトの制作について

グループ A は「NPC に命を吹き込みたい」を目標として制作を進める。NPC (Non Player Character) とはゲーム内に登場する、プレイヤーが存在しないプログラム制御のキャラクターを意味する。多くのゲームでは、NPC ごとに固有のセリフや音声、容姿などを設定しプレイヤーとコミュニケーションを図ることが可能だ。しかし、現状のゲームに登場する NPC は、リアリティに欠ける物が多い。理由として NPC は予めプログラムされたことしか行動できず、コミュニケーションも決まったパターンと決まった方式でしか行えないことが考えられる。このことからプレイヤーは NPC に対しいわゆる NPC 感を抱くことになり、その感覚はゲームの世界に対しても伝播し、この世界は作り物なのだという認識になってしまう。そこで NPC の感情や会話内容などによって本物の人間のようにセリフや表情、アニメーション、ジェスチャーなどをリアルタイムで制御することで、NPC のリアリティを大幅に向上させることを目標とする。

(※文責: 白石侃穩)

グループ B はメタバース内の没入感をより高めることを目標として制作を進める。グループ B のメタバースの世界観は「崩壊した近未来都市」である。そのため、大小様々な瓦礫や壊れたビル群、ひび割れたコンクリートの道路など、荒廃した世界を表現するオブジェクトの制作を行う。また、世界観の説得力を増強させるために、プレイヤーのアバターを近未来的な探索ロボに搭乗させる。(以下、探索ポットと呼ぶ) また、その探索ポットに瓦礫等の進行を阻害するオブジェクトを破壊するための武器や道具を追加することで、仮想空間ならではの現実ではできない体験をしてもらい、世界観に没入してもらえそうな外観作りを目標とする。

(※文責: 出田光)

### 2.3.3 対話型 AI の制作について

AI チームは、他のチームとは異なり、人間のように会話できる AI の作成をグループ共同で製作することを目標としている。この目標を達成するために、私たちはさまざまな要素を考慮した。まず、人間のように会話できる要素として、自然な会話性を追求した。これは、単にテキストを生成するだけでなく、流暢で論理的なコミュニケーションを可能にすることを意味する。また、個性や感情の要素を AI に持たせることも重要視した。個性を持った AI は、単なる NPC 以上の存在となり、ユーザーと親しみやすい関係を築くことができる。これらの要素を組み込むため、以下の工程を最終目標とした。

- ・音声での入力の実現

VOICEVOX を使用して、ユーザーが音声で情報を入力できるようにすることで、AI アバターとの対話がより直感的で自然なものになる。

- ・AI アバターの個性や感情の表現

個別のキャラクターや個性的な特徴をアバターに組み込むことで、会話のスタイルや反応に個性を持たせることができる。また、表情変化などの視覚的な要素も実装し、アバターが感情を表現できるようにする。

- ・ChatGPT と Unity, VOICEVOX との連携

自然言語処理に優れた AI モデルである ChatGPT を、アバターの表現や音声出力のために Unity

## METAVVERSE DE AVATAR

と VOICEVOX に組み合わせる。これにより、文字の入力に対する応答処理や音声出力などが統合され、より総合的な対話体験を提供できる。

また各グループ独自の目標として、A グループでは AI アバターとの対話内容によって AI アバターからの好感度が変化するシステムを実装する。また、この好感度の度合いによって AI アバターからの返答の内容を変化させることを目標とした。B グループでは、数段階の成長レベルを設定し、最終的には喜怒哀楽の感情が表現できるシステムを実装する。また、プレイヤーの行動によって異なる成長をし、好みの性格の AI に成長させることができるような AI の作成を目標とした。

(※文責: 長瀬カイセイ)

## 第 3 章 前期の開発

### 3.1 メタバース空間の制作について

#### 3.1.1 制作過程

メタバースチームは、Unity を使用して VRchat 上で利用可能なメタバースの開発を行った。"VRChat"とは、VR 空間で他のユーザーと交流できるソーシャル VR プラットフォームである。ユーザーは自分自身を表現するアバターを作成し、VR 環境内でコミュニケーションを取ることができる。また、自分だけのワールドを設計して他のユーザーを招待することも可能である。チーム全員が Unity を初めて使うという状況だったため、プロジェクトの最初のヶ月間は Unity の基本的な操作や利用方法の習得に専念した。これは、一冊の参考書を読み終えるほどの時間と労力を必要とした。この初期学習期間を経て、6 月からは具体的に VRchat のワールドの作成を始めた。我々は VRchat Creator Companion と Unity ver.2019.3.4.1f1 というツールを使用して開発を進めた。我々の開発の初めの一歩として、簡易的な飛行船の 3D モデルを 3DCG チームに依頼した。このモデルは我々の学習と実践のための素材として活用し、初期段階の技術実装の練習に使用した(図 3.1)。その後、6 月中旬からは 3DCG チームが作成した完成度が高い飛行船のモデルを使い始め、実際のプロトタイプの実装に取り組み始めた(図 3.2)。6 月の下旬から、我々は中間発表で利用する飛行船を用いたワールドの開発を始めた。

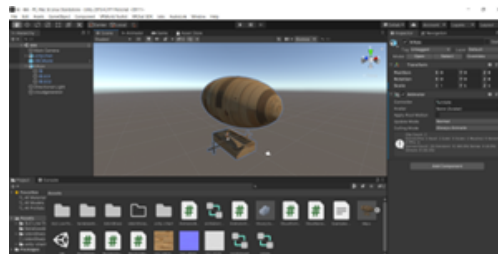


図 3.1 練習に使用したモデル



図 3.2 飛行船のプロトタイプ

(※文責: 上野大道)

### 3.1.2 進捗

中間発表の段階では、最終目標として設定したメタバースに相応しい、流用可能なメタバースを制作した(図 3.3)。飛行船については、両翼に配置したプロペラをアニメーターコントローラーを使用して回転させることで、船が実際に動いているかのような錯覚をプレイヤーに与えるよう工夫した(図 3.4)。さらに、飛行船の前方から雲を生成し、これを船側に移動させることで、視覚的に船が進行しているという体験を可能にした(図 3.5)。飛行船の内部には、さまざまな要素を配置した。まず、飛行船内には NPC を配置し、空間に活気を与えるようにした(図 3.6)。また、プレイヤーが操作可能なサイコロ型とボール型のオブジェクトを設置し、船内での遊びの要素を提供した(図 3.7)。さらに、椅子にはプレイヤーが実際に座ることができる機能を追加し、飛行船内でのリアリティを高めた(図 3.8)。



図 3.3 制作中のメタバース

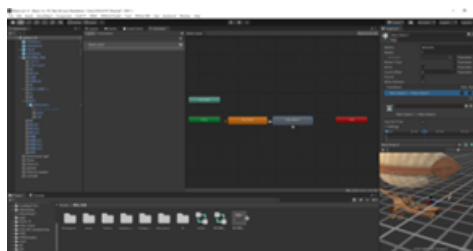


図 3.4 飛行船にアニメーションをつける



図 3.5 移動する雲



図 3.6 飛行船に設置した NPC



図 3.7 操作可能なサイコロとボール

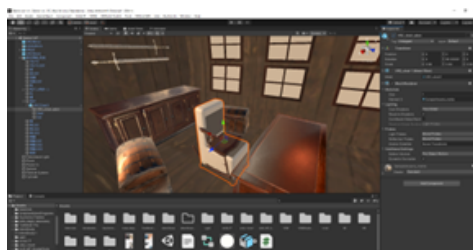


図 3.8 座れるイス

(※文責: 上野大道)

また、グループ B についてはプレイヤーがアバターを操作しワールドを歩き回ること、AI アバターがプレイヤーに追従して歩行することが可能。AI アバターの当たり判定や速度を設定し、不可解な動きをしないよう調整をしている。特定の物に触れた時、触れた物とプレイヤーアバターを

関連付け、その物を廃棄することや保持することができる。マウスの右クリックで物を遠くに射出することが可能になった。

(※文責: 沼田晴登)

## 3.2 3DCG の制作について

### 3.2.1 制作過程

NPC (キャラクター) について以下のとおりである。

制作時間短縮用 NPC (担当者: 西村, 白石)

1. stable diffusion でキャラクターデザイン
2. VRoid で制作
3. 完成

中間では制作期間的に NPC 制作に多くの時間を割くことが叶わなかったため、手軽な NPC 制作が可能な VRoid を用いた。NPC のビジュアルは stable diffusion で生成している。中間では一体だけの制作だったが、以降は分担して行う予定である。

クオリティ重視用 NPC (担当者: 西村)

1. stable diffusion でキャラクターデザイン
2. Blender で制作
3. 完成

こちらの制作法が理想である。当初は Blender での NPC 制作を行っていたのだが、時間的制約により、中間では次善策として VRoid を用いた NPC 制作に切り替えた。今後の方針によっては、中間以降に VRoid 版 NPC を制作途中の Blender 版 NPC に置き換える可能性はある。また Blender で制作したい理由としては、制作可能な NPC の専門性にある。Blender に対して、VRoid は非常に手軽に一定水準以上のクオリティの NPC を短時間で制作できる所に魅力がある。だが、汎用的に制作できる半面、とりわけ NPC の表情やモーションを詳細に作りこむには適していないモデル構造になっている。すなわち、今回のような AI で NPC を詳細に制御する場合等の特殊な環境に必ずしも適しているとは言えない。対して Blender の場合多くはフルスクラッチでの制作になり、その分目的に適した NPC をオーダーメイドで製作できる。今回の場合で言うと、AI 制御に適した表情制作や、今後予定しているモーションキャプチャでのアニメーション制作時に適したアーマチュア構造の構築などが該当する。

ワールドについて以下のとおりである。

中間発表用ワールド (担当者: 西村, 白石)

1. stable diffusion でデザイン (西村)
2. デザインを元に完成イメージ像作成 (西村)
3. Blender で外装の作成 (西村)

## METAVVERSE DE AVATAR

4. Blender で内装の作成 (西村, 白石)
5. SubstancePainter でテクスチャの作成 (西村)
6. Blender でアニメーションの作成 (西村)
7. 完成

中間では期末で制作するワールドを縮小した物を一つ制作することにした。デザインの原案は stable diffusion に生成させている。そのデザインをもとに、Blender で 3D モデルを制作する。作業分担に関しては、メンバーの白石が Blender を今回始めて使用するため、まずは作業しながら Blender の利用法を学習した。そのため、様々な技術を幅広く求められるワールドの内装部分を白石と西村の二人で協力しながら制作し、効率よく制作と利用法の学習を進めた。中間以降は分業体制を拡大する予定である。

(※文責: 西村光陽)

### 3.2.2 進捗

キャラクターについて、グループ A での制作におけるゲームの世界観「ファンタジー」に沿ったキャラクターを制作した。制作キャラのビジュアルは、画像生成系 AI の stable diffusion で生成した画像を参考に行っている (図 3.9)。Blender で制作した図のような作品が本来は理想的な 3D モデルとなる (図 3.10)。ただし時間的制約により、今回は次善策として図 3.11 のように VRoid を用いてキャラクターの制作を行った。



図 3.9 AI 生成された NPC の画像

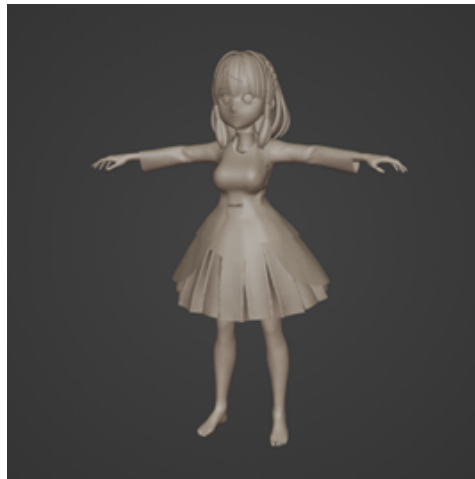


図 3.10 Blender で制作中の NPC

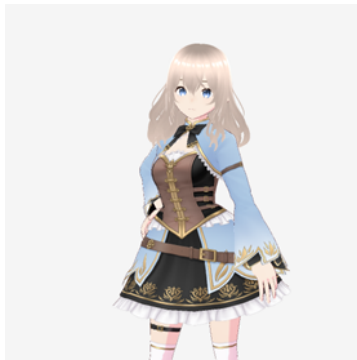


図 3.11 VRoid で制作した NPC

また、ワールドについて、グループの制作におけるゲームの世界観「ファンタジー」に沿ったワールドとして飛行船を作成した。こちらもビジュアルは、生成系 AI の stable diffusion で生成した画像を参考としている (図 3.12)。外装に関しては図 3.13 や図 3.14, 内観や家具については図 3.15 のようなものを制作した。



図 3.12 AI 生成されたワールド画像



図 3.13 Blender で作成した外観 1



図 3.14 Blender で作成した外観 2



図 3.15 Blender で作成した内観

制作における工夫点は複数テクスチャのアトラス化を行ったことである。アトラス化とは、複数存在するテクスチャを一つのテクスチャにまとめる処理の事を指す。Unity の仕様上、多数のテクスチャを複数同時に読みすぎると内部の処理が煩雑になり、かつメモリを無駄に消費してしまう。そのため、テクスチャを一つにまとめることで動作を軽快にすることができる。図 3.16 がアトラス化する前のテクスチャ、図 3.17 がアトラス化したテクスチャである。





図 3.16 アトラス化前のテクスチャ



図 3.17 アトラス化後のテクスチャ

(※文責: 西村光陽)

グループ B では, 3DCG 制作班の前期の活動では, blender の学習と制作物のイメージボードの描画, VRoid を使用したアバターの制作をした. blender のソフトをインストールし操作方法を学びながら動画サイトなどで練習用の解説動画を元いくつかの CG モデルを制作した. グループ B のメタバース内のイメージを固め, モデリングの元となるイラストを描いた. さらに, グループ B のメタバース内で登場する AI が操作するアバターの制作を行った.



図 3.18 グループ B のイメージボード 1



図 3.19 グループ B のイメージボード 2

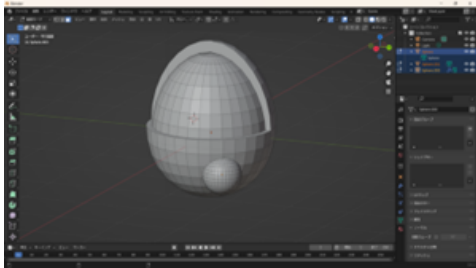


図 3.20 製作中の探索ポット



図 3.21 AI アバター

(※文責: 出田光)

### 3.3 対話型 AI の制作について

#### 3.3.1 制作過程

まず、A グループ、B グループが作成するメタバースにて両グループに共通する仕様をチーム内で共有、確認した。このとき、一部メンバーが所属するコースが知能システムコースではないため、人工知能および、自然言語処理に関する知識が乏しいという意見があった。そのため、知能システムコース所属のメンバーが自然言語処理、人工知能に関する簡易的な説明をすることでメンバー内における知識の共有をした。5月中旬、ChatGPT の API 契約が認可されるまでの間、並行して作成するメタバースにて使用する自然言語理解、対話制御のシステムとして、教員からのアドバイスで rasa と Python を使用し、GiNZA で日本語に対応させたシステム作成した。しかし、作成したシステムは AI からの返事となる部分が決まった言葉しか言わないことが、本チームの目的である自然な会話を達成することが困難と判断された。その結果、作成するメタバースへの採用、使用とはならなかった。上記システムの採用が見送られた後 6 月になり、ChatGPT の API を使用できるようになった。Python で ChatGPT が本プロジェクトの目的を満たすことができるか簡易的な検証の後、Unity に導入した。ChatGPT を Unity に導入した後は文章を事前にソースコード内に入力、文章がコンソール部に出力させる形式であった。その後、ゲーム画面となる部分で文章を入力できる形式にした。音声の入出力を実装するために Windows Speech Recognition で音声の入力、VOICE VOX を使用して音声を出力させるようにした。

(※文責: 中島一侃)

### 3.3.2 進捗

中間発表の時点では、Unity を使い音声で対話するシステムのプロトタイプを制作した (図 3.22)。音声認識には Unity エンジンに搭載された DictationRecognizer というライブラリを用いた。このライブラリは WindowsOS でのみ作動する。文章生成は ChatGPT API を利用した。ChatGPT API は利用する言語モデルを選ぶことが出来るが、今回は gpt-3.5-turbo を利用した。音声出力には VOICEVOX エンジンを利用した。今回のプロトタイプではローカルで VOICEVOX エンジンを起動する必要がある。

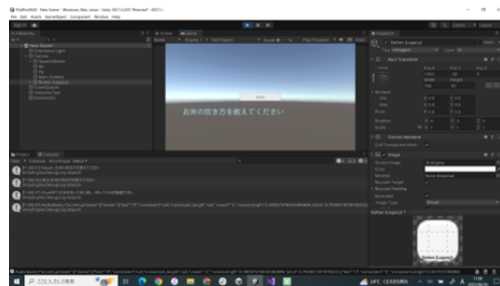


図 3.22 プロトタイプ

実際に制作するプロダクトにもこれらのサービスを利用し実装する予定である。AI に性格を持たせるプロンプトは検証段階である。また、VRChat のワールドなど他チームの成果物と統合する手法については未定であり、これから選定していく。

(※文責: 長瀬カイセイ)

## 第 4 章 前期における課題

### 4.1 中間発表の反省

チームごとにそれぞれ制作していた成果物はあったのだが、それらを合わせて一つの成果物として披露することができなかった。その結果、メタバースと対話 AI を組み合わせてコミュニケーションの形を考えるという本プロジェクトの趣旨がうまく伝わらないことがあった。一方、発表の技能については、発表の際にデモを用いるなど、工夫したこともいくつかあり、その点は聞き手に評価していただいた。しかし、発表練習の時間が取れなかったことから原稿を見ながら話すことが度々あったり、適切な声量で話すことができなかったことが反省点として挙げられる。

(※文責: 横山直矢)

### 4.2 今後の課題と展望について

反省を踏まえ、後期の活動開始直後にチームごとの成果物を組み合わせて空間の作成に取りかかる。また、このようになった原因の一つとして、チーム間の連携がうまく取れていなかったことも考えられる。そのため、SNS を活用し綿密に連絡を取り合うことが重要である。具体的には、Discord や LINE など、日常でも使用頻度が高い SNS も併せて用いることで、密に連絡が取れるようにするなど、工夫していく。

また、スケジュールの見通しが甘かったため各グループの作ったものをまとめて一つの成果物として作ることができなかったため、後期はスケジュールの立て方についても工夫する。

(※文責: 横山直矢)

#### 4.2.1 メタバース空間の制作について

後期以降は、他のチームで作られた素材やシステムを調整したうえで取り込み、ワールド全体のバランスを考える必要が出てくるので、夏季休暇中に積極的に Unity や、必要であれば他チームが使用しているツールについても理解をしておくことで開発がスムーズに進むと考えている。

また、VR ゴーグルとの連携がまだ完全ではないことや、VRchat との連携など、実装する環境下でのテストが不完全であるため、夏季休暇中や後期の活動が始まってからすぐにテストすることが必要である。

(※文責: 横山直矢)

#### 4.2.2 3DCG の制作について

グループ A について、現在 NPC は VRoid で制作を行っている。だが VRoid には 2.2.2 章のクオリティ重視用 NPC 項でも述べた、表情やモーションを詳細に作りこめない問題が存在してい

る。しかし、Blender で NPC を制作する場合には相応の時間を要するため、安易に Blender で制作するとその他プロジェクトで求められる制作物を完成させることができなくなってしまう恐れもある。そのため、まずは今後の NPC 制作についてチームで議論し、Blender で行うのか引き続き VRoid で行うのかを決める必要がある。また Blender で NPC を制作する場合、制作時間短縮のため今まで西村が制作してきた 3DCG キャラを NPC 専用に転用する手も視野に入れたい。

また、現在のワールド状況は海賊船が空中に存在するだけとなっているが、当初の目標ではファンタジーという世界観を十分に表現することにある。それを実現するために様々な建築物、オブジェクトを追加することが中間以降の課題であった。しかし、中間での制作状況と成果物を鑑みると、必ずしもワールドを端から端まで全て完璧に再現しきる必要はないのではないかと考えている。これには制作期間による妥協も含まれるが、作品のクオリティ向上のためにワールドの物量は必要ではないと判断したからでもある。すなわち、そのワールドを構成する主要なスポットのみを制作し、そこから見える景観や雰囲気などからそのスポットはワールドのどの辺りなのか理解できるつくりになれば、コミュニケーションの場としては問題なく機能するのではないだろうか。よって今後の展望としては、まずコミュニケーションを促進させる特定のシチュエーションをいくつか考察し、それを再現可能なスポットを設計する。3DCG 班はそのスポットを忠実に再現し、スポット同士の補完を為せるデザインを施すことになる。

(※文責: 白石侃穩)

グループ B については、現状の問題点として、前期までの制作の進捗が滞っている。グループ B の 3DCG 制作メンバーがプロジェクトリーダーとしての作業も兼任しているため、肝心の CG 制作に割ける時間が上手く配分できていない。夏季休暇の間に個人の技術力を高め制作スピードを上げると共に、グループ A の 3D 制作メンバーに適宜協力を仰ぎながら制作をしていく必要がある。

(※文責: 出田光)

### 4.2.3 対話型 AI の制作について

現在、対話 AI には 3 つの課題が挙げられる。AI の一連の応答に掛かる時間の改善、VRChat 内での動作確認、そして、各グループの製作物に合わせた、AI の個性付けである。

まず、AI の一連の応答にかかる時間の改善について、これは方針の見当がついている。ChatGPT からの応答文を音声化する際に、その応答文を句読点で区切って、その都度音声化するというものである。この手法を用いることで、応答に掛かる時間は一定の改善が見込まれるが、問題点もある。それは、この手法以外での改善が難しく、現状ほとんど頭打ちということである。現在、対話 AI の応答生成は ChatGPT の API を利用しており、すなわち外部サーバに頼っている状況である。当然、API そのもののアルゴリズムを改善することは不可能であるため、上記のような手法を用いたり、ChatGPT に対する指示を工夫する等の手法しかなく、抜本的な解決は困難であると考えられる。また、外部サーバに頼っている現状は、2 つ目の課題にも繋がってくる。2 つ目の課題は、VRChat 内で動作するかが不透明ということである。現在の対話 AI は応答生成に ChatGPT を、応答の音声化に VOICE VOX を用いている。ChatGPT は外部サーバと、VOICEVOX はローカルサーバとやり取りをしている。そしてこれらは、Unity 上では動作することが確認できている。しかし、VRChat 内で動作するかは不透明であり、外部サーバと VRChat の連携のために、別のシステムを挟む必要があることも考えられる。現在、様々な手法を検討中である。3 つ目の課題で

## METAVVERSE DE AVATAR

ある、各グループの製作物に合わせた AI の個性付けについては、ChatGPT への指示を工夫し、対応する予定である。具体的には、ChatGPT への指示次第で、現在の感情を数値で自己申告させることが可能であることが確認できている。これらの数値をシステム側に渡すことで、感情による挙動の変化が実装可能であると考えている。同じようにして、A グループの製作物の特徴である好感度や、B グループの製作物の特徴である成長度も実装可能であると考ええる。

総じて、本対話 AI では、ChatGPT の成す役割が非常に大きいと言える。しかし、ChatGPT の応答の流暢さには目を見張る物があり、「対話」をする上でこれ以上ない適役であると考えたため、今後も ChatGPT を主軸に据える方針である。

(※文責: 大原拓己)

## 第 5 章 後期の開発

### 5.1 メタバースの制作について

#### 使用ソフト，ツール

以下に，A，B グループ共通で使用したソフトについて具体的に述べる．

##### ・Unity

VR をはじめとする 3DCG を用いるゲームの開発において最も拡張性があることから，本プロジェクトではメインで使うことになったソフトである．前期では Unity，およびワールド内の仕掛けは C# 言語で記述していたが，AI チームとの開発の兼ね合いもあり，後期は開発プラットフォームを VRChat へ移行した．その際，Unity には後述する外部の拡張機能を導入して使用することとなった．なお，Unity のバージョンは 2021.4.31f1 を使用した．

##### ・VRChat

今回メインのプラットフォームとして使う，メタバースの一種である VRSNS のソフトで，利用者はアバターを介して他の利用者とコミュニケーションを取る．作ったワールドのテストやアップロードするには実際にこのソフトが必要だ．後述する VRChat Creator Companion を利用する際に利用者登録をする必要がある．

##### ・VRChat Creator Companion 及び UDON

Unity の外部拡張ツールで，VRChat 専用のワールド開発に用いるものである．このツールを用いることでアバターの導入やテストプレイを容易に行うことができるようになる．また，VRChat Creator Companion を使う際は，合わせて UDON，もしくは UdonSharp という VRChat 専用のプログラミング環境も使用することとなる．UDON とは VRChat のワールド開発において，ワールド内の仕掛けなどをノードプログラムによって行い，直感的に記述できるものである．しかし，ノードプログラムの場合，複雑な処理を行う場合は冗長になりやすいというデメリットがあるため，今回は UdonSharp (通称 U#) を用いることとした．U# は，UDON を C# 言語のような記述方法で文字列としてプログラムできるようにしたものである．前期に C# でプログラムを記述していたため，新たに覚えることが少ない U# で開発することにした．

VRChat 用のワールド作成にあたって，前準備としてエディターを立ち上げた際に Project 欄にある VRCWorld.prefab を Hierarchy にドラッグ&ドロップする必要がある．加えて，ワールドの床となる 3D Object も設置する必要があるため，床となるオブジェクトも Hierarchy に加える．

##### ・CyanEmu

VRChat のワールド作成時に，Build & Test を行って VRChat のローカル環境にアップロードすることでテストプレイができるようになるのだが，このツールを入れることで Build & Test の過程を飛ばして Unity 単体でテストプレイを行うことができる．これは github 上で UnityPackage としてインストールする必要がある．使い方は，インストールした CyanEmu の UnityPackage を

Project ウィンドウの Assets にドラッグ&ドロップすると反映され、Unity の実行ボタンを押すと同時にこのエミュレータも起動する。

(※文責: 横山直矢)

### 5.1.1 A グループ

プロジェクトの後期では、前期で習得した技術と飛行船の制作物を基盤として作業が開始した。後期では前期と同様に VRChat, Unity 2019.4.31f1, および VRCSDK の VRChat Creator Companion (VCC) を主要な開発ツールとして使用した。後期初週には重要な会議が行われ、メタバースで使用するプラットフォームとして VRChat の適切性について議論した。この会議の主な焦点は、VRChat での音声データ処理の問題に関するものであった。しかし、VRChat は広範囲に普及しており、多くのユーザーが簡単にアクセスできる点が大きな利点であった。これにより、新しいメタバース環境を広い範囲の人々に容易に紹介することが可能であった。また、VRChat はユーザーが自身のアバターや世界を自由に作成し、カスタマイズできる柔軟性を持っていた。これらにより、個々のニーズやアイデンティティに合わせたユニークなメタバース体験の創出が可能になると考えた。さらに、技術的な面では、多くの VR セットアップオプションが提供されていた。そのため、プラットフォームは VRChat を選択した。この会議を経て、プロジェクトチームは VRChat の使用を決定し、AI との組み合わせに関する既存の記事を基に学習を進めた。また、どのようなメタバースを最終的に作成するか、そのために必要な技術に関する議論も行った。この議論の結果、最終的にバー、占いの館、女の子の部屋、カフェ、および前期で作成した飛行船の合計 5 つの部屋をメタバースの中心要素として採用することに決定した。

次の週から、最終成果物の制作に向けて、メタバースチームはまず、異なる世界観やサイズの部屋をどのように表現し、ユーザーに違和感なく体験させるかについての会議を開催した。会議では、様々なアイデアが提案され、詳細な会議が行われた。この会議の結果、アパートの廊下を作成し、そこにあるドアから各部屋にアクセスできる仕組みを採用することに決定した。このアイデアは、実際のアパートの廊下と同様に、異なるサイズやテーマの部屋への入口として機能し、ユーザーに新しい体験を提供するものだと考えた。特に、カフェのような大きな空間と、占いの館のような小さな空間を、同じアパートの廊下からアクセスできるようにするため、非現実的な要素を取り入れた。具体的には、ドアの先にある部屋が物理的な空間にとらわれないというコンセプトを採用した。これにより、ユーザーは一つのドアを開けると、まったく異なる世界に足を踏み入れることができるようになった。この方法は、プレイヤーに予期せぬ驚きと探索の楽しさを提供することを目的とした(図 5.1)。続いて、廊下と各部屋をどう繋ぐかについても検討した。部屋のサイズが異なるため、隣接する部屋同士が物理的に干渉しないようにする必要があった。その解決策として、テレポートシステムを使用することに決定した。廊下にプレイヤースポーン地点を設置し、各ドアの横にインターホンを設置した。プレイヤーがインターホンを押すと、対応する部屋のドアの前にテレポートされる仕組みを実装した(図 5.2)。また、部屋を出る際にも同じ廊下に戻れるように、各部屋のドアの先に廊下とテレポートを設置した(図 5.3, 図 5.4)。これにより、プレイヤーはスポーン地点の廊下から複数の部屋にアクセスし、部屋を出た後に同じ廊下に戻ることが可能になった。このシステムは、ロビーのような複数のプレイヤーが交流できる空間としても機能することも考えた。テレポートシステムの実装には、Booth で配布されている「まめもやシステムズ(蕪)



## METAVVERSE DE AVATAR

【フェード暗転】やさしいテレポート【VRCSDK3/U#】を使用した。これにより、ユーザーはスムーズで自然な移動を体験できるようになった。さらに、このシステムの導入は、メタバースのデザインにおける創造性を示すものだった。従来の物理的な制約にとらわれないデザインにすることで、メタバース内でのユーザー体験をより豊かで魅力的なものに変えることができた。また、このような非現実的な要素の導入は、メタバースが現実世界とは異なる独自の体験を提供できる強みを活かすものであり、ユーザーに新しい形のエンターテインメントを提供することに貢献した。

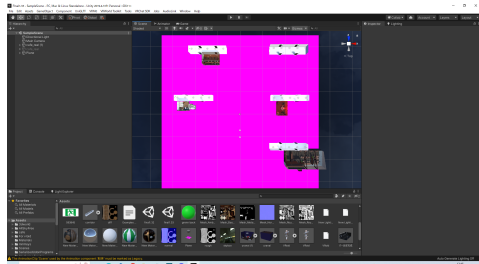


図 5.1 Unity 内の配置



図 5.2 ゲーム内のインターホン

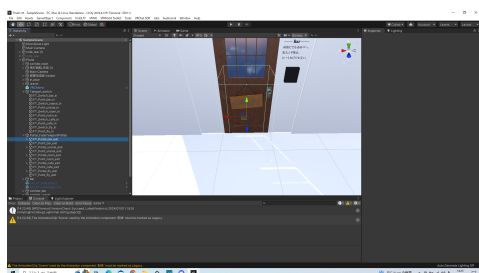


図 5.3 ドア先のテレポート元

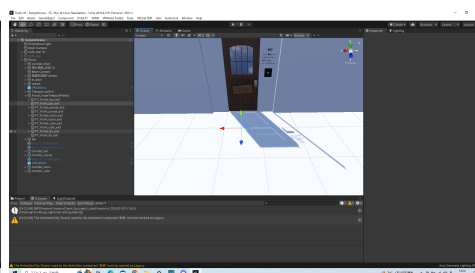


図 5.4 テレポート先

メタバースに存在する開閉するドアの実装は、メタバースにおいて重要な技術であり、Unity の Joint を使用した。まず、ドアの位置に空のオブジェクトを作成し、これをドアオブジェクトの親オブジェクトとして設定した。これにより、ドアの動きをより柔軟に制御することが可能になった。また、ドアノブの位置に Sphere Collider を配置した。次に空のオブジェクトに Hinge Joint と Fixed Joint を追加した。Hinge Joint の Anchor の位置を調整し、ドアが実際に回転する軸に合わせ、Axis の値を変更してドアが開く方向を決定した。Limits の Min と Max の値を-135 度と 135 度に設定することで、実際のドアの開閉角度を再現した。Fixed Joint には、ドアノブオブジェクトを Connected Body として設定した。これらの設定により、ユーザーはドアノブを操作してドアを開閉することができるようになった。また、Hinge Joint の強さを強くすることで、ドアは自動で閉まるようなシステムにした。よって、ドアの動きは滑らかで、リアルな物理的な反応を示した (図 5.5)。このドアの実装は、メタバースのリアリズムとインタラクティブ性を高める重要な要素であった。ユーザーはドアを自然に開閉でき、メタバース内での移動がスムーズに行えることができた。また、ドアの動作はメタバースの各部屋のプライバシーと区切りを提供し、実際の物理環境を模倣することでユーザーの没入感を高めることが出来た (図 5.6)。

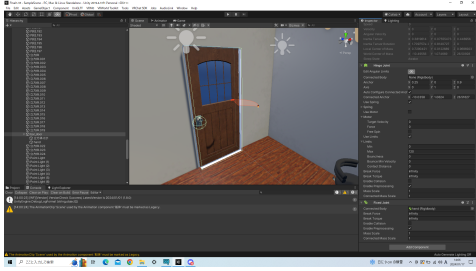


図 5.5 Unity 内のドアの設定

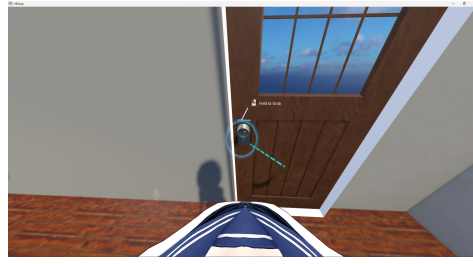


図 5.6 ゲーム内のドア

ドアの制作に続き、メタバースチームでは照明の設置にも着手した。部屋の雰囲気を作り出すために、照明は非常に重要な要素であった。このメタバースでは、各部屋の照明には電球のような光を想定し、Unity 内の Light コンポーネントの中から Spot Light を選択した (図 5.7)。Spot Light はその方向性と焦点を絞った光で、特定のエリアを照らすのに適しており、部屋に温かみと奥行きを与えるのに理想的であったためだ。廊下の照明に関しては、蛍光灯のような横長のライトを再現する必要があった。しかし、Unity にはそのような形状のライトが用意されていないため、こちらも Spot Light を使用し、蛍光灯のイメージを模倣した。この際、Spot Light の配光を調整し、蛍光灯特有の広がり均一性を再現することに重点を置いた。さらに、照明の強度、色温度、範囲などを細かく調整し、各部屋の雰囲気やテーマに合わせた照明を実現した。例えば、バーでは暖色系の柔らかい光を使い、リラックスできる雰囲気を作り出した (図 5.8)。占いの館では小さい光を目立たせるために少し暗めに設定した (図 5.9)。一方、カフェでは少し明るめの光を用いて活気ある環境を演出した (図 5.10)。

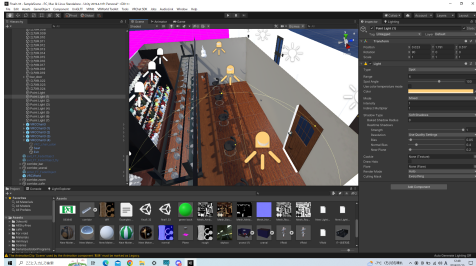


図 5.7 Unity 内の light の設定

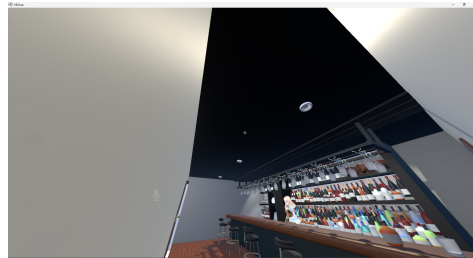


図 5.8 バーのライト



図 5.9 占いの館のライト



図 5.10 カフェのライト

バー、カフェ、占いの館など、座席が必要なシーンにおいて重要な椅子に座るシステムは、VRCSDK に含まれる VRchair3 を用いた。このシステムでは、VRchair3 を物理的な椅子と同じ位置に配置し、視覚的に透明にすることで、プレイヤーが実際に椅子に座っているような体験を提供した。プレイヤーが椅子の位置に移動してトリガーを引くと、VRchair3 がこの動作を検知し、

## METAVVERSE DE AVATAR

キャラクターが座ったポーズを取るようになった。これにより、プレイヤーは視覚的には実際の椅子に座っているように見え、没入感を高めることができた。このシステムのメリットは、プレイヤーが自然な方法で椅子に座る動作を行えることにあった。また、VR環境において、椅子に座る動作はよりリアルな体験を提供し、プレイヤーの没入感を高めることができた(図 5.11)。さらに、このシステムはカスタマイズが可能で、椅子のデザインや配置を変更しても、VRchair3 の位置を適切に調整するだけで簡単に対応できる。プレイヤーがこれらの環境で椅子に座ることで、よりリアルな対話や交流の体験が可能になった。例えば、バーでは、プレイヤーが椅子に座ることで、店にきた演出を楽しむことができた(図 5.12)。占いの館では、プレイヤーが占い師の前に座ることで、より個人的で集中した占い体験を演出することができた(図 5.13)。また、カフェのシーンでは、プレイヤーが実際にテーブルに座り、コーヒーを飲んでいるかのような体験を提供することができた(図 5.14)。

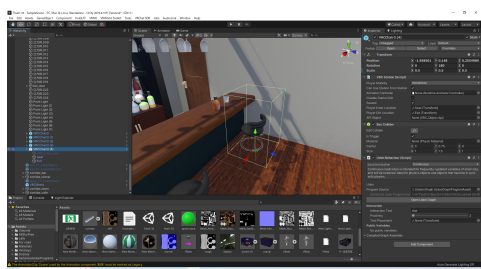


図 5.11 Unity で内の椅子の設定



図 5.12 ゲーム内のバーの椅子

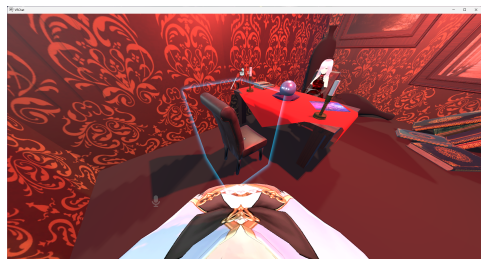


図 5.13 ゲーム内の占いの館の椅子



図 5.14 ゲーム内のカフェの椅子

メタバースに多く存在する持つことが可能なオブジェクトは、メタバース体験のリアリズムとインタラクティブ性を高める上で重要な要素であった。オブジェクトを持てるようにするために、VRChat の SDK (VRCSDK) に含まれる VRC\_Object Sync と VRC\_Pickup の二つのコンポーネントを利用した。これらのコンポーネントをオブジェクトに追加することで、プレイヤーはこれらのオブジェクトを自然に拾い上げ、操作することが可能になった(図 5.15)。例として、バーではコップや瓶にこれらのコンポーネントを追加した。プレイヤーはバーで実際にコップを手に取り、飲むふりをするなどの動作を体験することができた(図 5.16)。同様に、女の子の部屋ではペットボトルなどの小物にもこれらのコンポーネントを適用し、よりリアルな部屋の雰囲気を作り出した(図 5.17)。このオブジェクトを持つ機能の実装は、メタバース内でのプレイヤーのインタラクションを大きく豊かにした。プレイヤーはオブジェクトを直接操作することで、環境により深く没入し、リアルな体験を得ることができた。また、Unity では、さまざまなタイプの Collider が提供されており、それぞれに異なる特性と用途がある。このメタバースでは、オブジェクトの性質と必要性に基づいて、適切な Collider を選択した。持たれる可能性があるオブジェクトや、プレイ

ヤーのインタラクションが頻繁に行われるオブジェクトに Mesh Collider を使用した。これにより、これらのオブジェクトの挙動がよりリアルで自然になることを目指した。例えば、プレイヤーが手に取ることができるコップやペットボトルなどがこれに該当した。Box Collider やその他の簡易的な Collider は、静的なオブジェクトやプレイヤーによる直接的なインタラクションが少ないオブジェクトに適用した。これらの Collider は、計算コストが低く、ワールドの全体的なパフォーマンスを維持するのに役立った。例えば、動かない家具や建築要素に Box Collider を使用することで、必要十分な当たり判定を提供しつつ、リソースの消費を抑えることができた (図 5.18)。

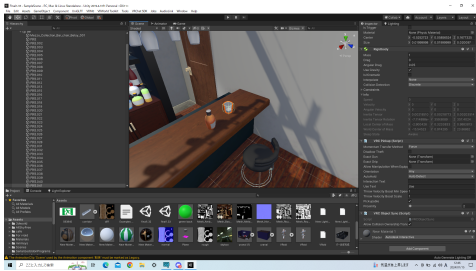


図 5.15 Unity 内でのオブジェクトを持つ設定



図 5.16 持つことができるバーのコップ

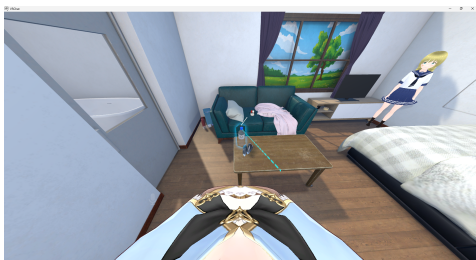


図 5.17 持つことができるペットボトル

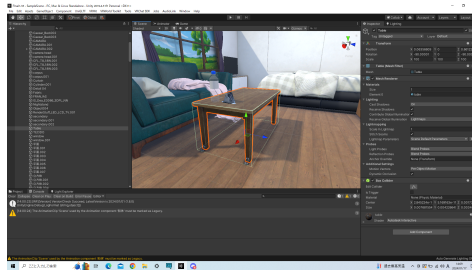


図 5.18 Unity 内で Box collider を適用したオブジェクト

メタバース環境の構築において、3DCG チームが Blender を用いて作成した部屋や廊下などの基本的な構造物を、実際の大きさを再現するために 1.6 倍に拡大した。この拡大率は、ローカルでの PC テスト時に、実世界のスケール感をメタバース内で正確に反映させるために重要であった。Unity 内での設定では、部屋や廊下の床が実際の地面に合うように配置した。地面を表す Plane オブジェクトは、干渉を避けるために -0.01 の位置に設定した。この微調整により、床と地面が視覚的に一体化し、よりリアルな環境を作り出すことができた。また、初期設定されている太陽の代わりにライトを削除し、夜のシーンをイメージして環境を設計した。これにより、部屋や廊下のライティングがより目立つようになり、特に夜間の雰囲気強調することができた。ライティングの調整は、ユーザーが感じる空間の雰囲気に大きな影響を与えるため、メタバースの総合的なデザインにおいて重要な要素であった。地面 (Plane) オブジェクトに対して部屋や廊下を子に設定することで、全体のサイズを一括で変更することが可能になった。これにより、プロジェクトのスケールや比率を容易に調整でき、さらには将来的な変更や拡張にも柔軟に対応できるようになった。また、この方法により、プレハブ化とエクスポートが簡単に行えるようになり、メタバース環境の再利用や共有が容易になった。この一連のプロセスは、メタバースの環境をデザインし、実装する上での効率性と柔軟性を向上させた。また、環境のスケール感やライティングの設計は、メタバース内で

のユーザー体験の質に直接影響を与えるため、これらの要素の適切な設定は質向上に繋がった。メタバースの基本的な構築が完了した後、メタバースチームは AI との連携に焦点を移した。この段階の目的は、メタバース内での音声データの取得と AI アバターの統合であった。この実装にはいくつかの技術的な挑戦が伴ったが、それらを克服することで、メタバースに新たな次元のインタラクティブティと自動化をもたらすことができた。音声データの取得には、VBcable というバーチャルオーディオデバイスを使用した。VRchat のデバイスに VBcable を設定するこれにより、VRChat と Python の間で音声データの入出力を効果的にルーティングすることができた。VBcable を介して、VRChat からの音声出力を Python スクリプトに送り、そこで処理を行い、結果を VRChat に戻すことが可能になった。そのまま実行すると、VRchat 内でプレイヤーの音が AI から返ってきてしまうが、VRchat の出力音量を 1 に設定することで、この問題を解決することができた。このシステムでは、AI を搭載したアバターをメタバース内に召還するために、1 体の AI アバターにつき 1 台の PC が必要であった。合計で 5 体の AI アバターを召還するためには、5 台の PC が必要であるということだった。召還された AI アバターは、プレイヤーとの対話する役割を担った。これらの AI アバターは chatGPT の API を活用して、プレイヤーとのインタラク션을より豊かで有意義なものにした。

プロジェクトの後半段階では、AI との連携完了後、Oculus Quest 2 を使用した VR 環境でのサイズ調整に重点を置いた。この作業は、メタバース内でのオブジェクトや環境のサイズ感を実際の VR 体験に最適化するために不可欠であった。ただし、大学から提供された PC のスペックの制約上、Oculus Link を使用することができなかった。VRChat には Oculus 版と Windows 版が存在し、それぞれ異なるプラットフォームでの利用が可能であった。AI アバターは PC を介して音声データを処理し、その結果を VRChat に送信する必要があったため、Windows 版の VRChat を使用した。しかし、Oculus Link を使用できないため、Windows 版での VR 環境でのテストは実施することができなかった。Unity と VRCSDK を使用して作成したワールドでは、Unity 上での VR テストが不可能であった。そのため、VR 環境でのサイズ調整のためには、Oculus 版 VRChat でワールドをアップロードする必要があった。VRChat でワールドをアップロードするためには、ユーザーランクを「visitor」から「new user」に上げる必要があった。これには約 10 時間のプレイ時間と他のユーザーとの交流が必要であり、このランクアップは前期の段階から行っていたが難航した。そのため、VRChat Plus にアップグレードし、ランクを無条件に 1 段階上げることで問題を解決した。Oculus 版でのアップロードが完了した後、実際の VR 環境でサイズを確認した。PC でのプレイと比較してサイズ感が大きく感じられたため、元々設定していた 1.6 倍から 1.3 倍に縮小することに決定した。このサイズ調整は、メタバース内でのオブジェクトや環境がユーザーにとって自然で快適に感じられるようにするために重要であった。

メタバースのワールドアップロードの完了後、次のステップとしてオリジナル AI アバターのアップロードに着手した。このプロジェクトでは、3DCG チームが VRoidStudio を使用してオリジナル AI アバターを作成した。VRoidStudio は、直感的に 3D キャラクターモデルを作成できるツールであり、特にアニメスタイルのキャラクターに適していたため使用した。VRoidStudio では、モデルは VRM ファイル形式で出力された。しかし、VRChat では FBX 形式が必要であった。そのため、モデルを VRM から FBX 形式に変換する必要があった。この変換には「ぼけもり BOOTH 支部 VRM Converter for VRChat」を使用し、変換の精度と効率を保証した。変換後の FBX ファイルは、Unity と VRChat の SDK を使用して VRChat に統合された。この段階では、アバターのアニメーション、テクスチャ、およびその他の設定が調整され、VRChat 内でのパフォーマンスを最適化した。アバターのアップロードにも World と同じく「New user」ランクが

## METAVVERSE DE AVATAR

必要であったが、この問題は既にワールドアップロード時に解決されていたため、AI アバターのアップロードはスムーズに進んだ。メタバース内でオリジナル AI アバターのアップロードと統合がスムーズに進行した一方で、AI アバターの感情と、それに対応する表情や動作の連携に課題を抱えた。AI アバターに感情を持たせ、それに応じた表情や動作を生成させることは、技術的に複雑な挑戦であった。特に、与えられた感情パラメータに適切な表情や身振りを対応させるプロセスは、高度な自然言語処理能力と詳細なアニメーション制御を要求された。この課題を解決することが出来ず感情表現に伴う表情や動作の変化は行うことが出来なかった。

プロジェクトのメタバース環境内では、5つの異なる部屋にそれぞれ1体の AI アバターを配置し、各部屋の独自性とインタラクティブ性を高めることに重点を置いた。これらの AI アバターは、部屋ごとに異なる役割と特性を持ち、ユーザーに対して独特の体験を提供することを目的とした。バーでは、AI アバターを店員としてカウンターの前に配置した。AI の存在は、バー環境をよりリアルで活気なものにし、プレイヤーとのインタラクションを促進することを想定した (図 5.19)。占いの館では、AI アバターを占い師として配置し、プレイヤーが座る椅子の正面に別の椅子を配置した。ここでは、AI が占い師としてプレイヤーと向き合い、対話を通じて占いを提供するため、この配置は、プレイヤーにより集中した対話体験を提供することを想定した (図 5.20)。女の子の部屋では、AI アバターを部屋の一角に配置した。当初の計画では、AI が部屋をランダムに歩き回る予定であったが、技術的な問題により実現することができなかった。そのため、AI は部屋の特定の場所で静的に存在し、プレイヤーが近づいて話しかけた際に反応するように設定した (図 5.21)。カフェでは多くの席が存在し、AI はカフェのスタッフとして AI をカウンター席の前に配置した (図 5.22)。前期から引き続き使用した飛行船内では、AI アバターを船首近くに配置した。



図 5.19 バーに配置した AI アバター

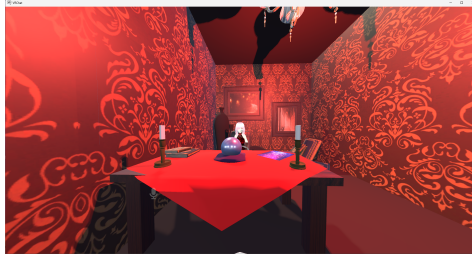


図 5.20 占いの館に配置した AI アバター



図 5.21 カフェに配置した AI アバター



図 5.22 女の子の部屋に配置した AI アバター

最終発表の直前に大学側から OculusLink に対応したスペックを持つ PC を提供していただいた。新しい PC の導入により、Oculus Link を利用して、Oculus Quest 2 を PC に接続すること

ができた。これにより、VR 環境でのインタラクションが大幅に向上した。以前は PC 上でのみ AI との対話が可能であったが、Oculus Link を使用することで、メタバース内で直接 AI アバターと対話する体験を実現することができた。これは、プレゼンテーションとデモンストレーションの両方において、メタバースの没入感とインタラクティブ性を大幅に高めるものであった。

(※文責: 上野大道)

### 5.1.2 B グループ

#### 開発の流れ

まず初めに開発エンジンである unity のインストールから行った。(Unity ver.2021.3.25f1 をインストール)

その後それぞれのパソコンでの環境設定を行い簡単な動作演習として、各オブジェクトの配置の仕方や、C#を利用してオブジェクトを動かす方法、マテリアルの貼り付けなど基礎的な内容を覚える。

プロジェクトの初めの方では、ネットで公開されている無料のアセットなどを利用して実際に Unity 内でアバターを動かしたときの動作感覚を覚えつつ、自分で実装できるように各要素の大きさや動く速さを覚える。

序盤からエラーが発生する。3DCG チームから貰った AI アバターのテクスチャーが表示されない。解決に難航していたが、ネット上で配布されているテクスチャーなどが正常に表示されない時に使えるパッケージをインポートすることで正常にテクスチャーが表示されるように。これで 3DCG チームからのアバター問題が解決した。

移動のプログラムを考えている中で、無料アセット内にあるものを活用しようと考えた。プレイヤーアバターの見た目を 3DCG チームの作成したアバターに変更し、そのアバターをそのまま利用しようとした。しかし、他の動きを取り入れたいときに不都合が生じたり、実際に自分たちが操作を行う際に分からない要素があると困るため、移動のプログラムについては自分で作ることにした。主にインターネット上にある Unity に関する情報を集めながら制作を始めた。できることを増やすため、機能の実装のため様々なギミックを考える。

B グループではオブジェクトを破壊することでアイテムを手に入れ、そのアイテムを AI アバターに渡すことで AI アバターへ感情を与えられるようなゲームの体験に近い感覚で AI との会話を成立させようとした。舞台を荒廃した街とし、その中を探検しつつアイテムを集めていく。そのため移動やオブジェクトの破壊のプログラムを優先して組み始め、基盤ができてから他の機能の実装に取り掛かることにした。

3DCG チームから貰ったプレイヤーアバターと AI アバター、街のマップを Unity 内にインポートする。それぞれに当たり判定となるコライダーを設定し、プレイヤーアバターには移動のプログラムとカメラ操作のプログラムを、AI アバターにはプレイヤーの後ろ側をついてくるような動きのプログラムを考える。

街のマップが広く、高さもあるため、立体的な動きができるように目指した。また、3DCG チームから追加の 3D オブジェクトをいくつか貰っているため、それらを活用しながら街の外装を整える。しかし、ここで問題が起きる。マップのマテリアルが貼られていないため、全て一色の状態であり、マップ自体が一つの塊になってしまっているため、新たにマテリアルを貼り付けても、どのビルも同じ色をしたビルになってしまう。別でオブジェクトを貼り付けてそれぞれのビルに特徴を

与えても良かったが、マップが広くビルも多いので作業量が多すぎたため断念。一色のビル街を探検ということにする。

「オブジェクトを壊してアイテムを手に入れる」これの実現のためにオブジェクトを壊す、壊れた際の挙動を実装することに。まずオブジェクト同士が触れ合うとき、条件付きの衝突判定をする。不意に何かに触れて壊れてしまっただけではいけないため、Unity 内にある Tag という機能を利用する。Player という Tag を作り、プレイヤーアバターの設定ができるタブの中で、プレイヤーアバターの Tag を Player に変更。続いて衝突判定の条件式の中に、「Player という Tag を持っているオブジェクトが触れた時」とすれば、プレイヤーアバターが触れた時以外では壊れることなくオブジェクトを維持することができる。壊れた際の動きは、触れられたオブジェクトを消し、事前に用意してある小さなブロックを積み上げまとめた 3D オブジェクトをロード、同じ場所に出現させる。積み上げたブロックは一つ一つに当たり判定と重さを設定してあるためそのまま崩れていく。これでオブジェクトに触れた際にバラバラに崩れて壊れる仕組みが完成。

懸念点としては、ブロックの集合体をロードし、崩す土ときに処理がとても重くなってしまうこと。崩れたブロックは数秒経ったら消えるようにプログラムを組んでいるが、連続してオブジェクトを壊す動きをすると画面が止まるほどになる。処理を軽くしつつ壊れた時の演出を考えることに。

制作の相談をする中で VRchat Creator Companion と Unity ver.2021.4.31f1 を利用して VRchat ワールドの作成をすることに。VRchat ワールドの作成に伴って、Udon と呼ばれる言語の理解に時間をかけることになった。Udon にも種類があり、初めは Udon Graph を触った。上手いこと利用することができず断念、もう一つの Udon Sharp を触ることに。Udon Sharp は C# のようなプログラムの組み方ができるため、こちらを採用。ただ、C# ほど自由に記述や機能があるわけではないため、C# で組んだプログラムを Udon Sharp で試し、もしエラーが出た場合は U# で動かせるように簡単なプログラムで置き換える作業をする。そんな予定だったが、Udon Sharp を利用できるようにするための UdonSharp.v0.20.3.unitypackage をインポートするとエラーが起きるようになってしまい、Udon Sharp が利用できない状態になってしまった。このエラーがしばらくの間直らないため、C# のみで作れる範囲のプログラムを書き、インポートできた時に備えることにした。

(※文責: 沼田晴登)

### 対話 AI チームの制作物との連携

対話 AI チームが作成した対話 AI プログラムをメタバース空間のアバターに割り当て、かつその AI アバターをプレイヤーに自動追従するようなシステムを構築しようと、様々な方法を試した。初めは NPC (Non-Player-Character) としての実装、すなわち一つのワールドギミックとするつもりで開発した。その際に行ったことを以下に述べる。

#### ・ NavMesh の設定

Unity の機能である NavMesh は、自動でワールド内の歩行可能な面を描写するデータ構造であり、ワールド内のオブジェクトの位置によってキャラクターのナビゲーションを変化させることができる。これを NPC に設定することで、障害物を回避しながら自動でワールドを移動することができる。

#### ・ NavMeshAgent の設定

また、今回はプレイヤーに自動追従する必要があったため、NavMeshAgent という関連機能も併



せて使った。NavMeshAgent はキャラクターを目的の場所まで誘導する機能で、移動速度、回転速度、加速度などのパラメータを設定できる。追従先にプレイヤーのアバターを設定することで、プレイヤーの歩行に合わせて NPC も同じように追従する。

しかし、NPC として処理する場合、対話 AI チームが作成したプログラムが動作しないことが判明したので、NavMesh の設定を引き継ぎつつ、OSC によって処理する方法へと変更した。

### ・OSC の設定

OSC とは、Open Sound Control というプロトコルのことである。AI の処理に関するデータを VRChat に送信する場合、OSC が指定するフォーマットで送信する必要がある。そのため AI に関するプログラムが記述されている Python に OSC 用のライブラリを用意し、送信するデータを OSC のフォーマットに変換できるようにする。python-osc のドキュメントを参考に VRChat 用のスクリプトを作成した。なお、VRChat のドキュメントより、Port 番号は 9000 に設定した。

次に VRChat の OSC 設定をオンにする。VRChat を起動した後、Action Menu より OSC の設定を開き、トグルスイッチを Enabled にすることで、上記のスクリプトを実行することができる。スクリプトをコマンドプロンプト等のコマンドラインから実行し、コマンドラインがインプット待機の状態になったときに -1, 0, 1 のいずれかを入力する。コマンドの /input/Vertical というアバターを前後に動かすためのアドレスに AI アバターは移動することができ、コマンドラインに数字を入力することで -1 は後進、0 は停止、1 は前進の動作を示す。これを応用することで、プレイヤーの動いた時間数だけ AI アバターも移動を行うことで追従することができる考えた。

しかし、対話 AI のプログラムと合わせることができなかつたため、結果的に違う方法で実装することとなった。詳細は 8.2.1 を参考。

(※文責: 横山直矢)

## 5.2 3DCG の制作について

### 5.2.1 A グループ

#### 開発の流れ

制作物における当初の目標では好感度システムを用いた人型 AI と没入感のある様々なワールド上でコミュニケーションを取れるようにすることを目標として開発手法としてウォーターフォール開発を採用し、必要なシステム機能、オブジェクトを列挙し要件定義や全体の機能設計を固めてから進行計画を作成した。その時点では中間時点までの計画を作成し、AI チーム、メタバースチーム、3DCG チームでそれぞれ作業の分担を行った。3DCG チームでは使用ソフトとして Blender を採用したため使用経験のある西村が主として制作を行い、未経験の白石が補助と Blender の勉強を行った。

中間時点では 3DCG チームとして仮想空間での動作テストも兼ねた飛行船を制作した。これによりメタバースチームの機能実装や最終目標への新しい目途を立てることが可能となった。中間時点での進捗状況と今後の計画を見直し当初目的であった好感度システムはコミュニケーションの上で会話をレベル的に成長させ個人同士でのコミュニケーションをより促進させる効果が期待をしていたが実装における AI の制御、それに伴うキャラクターのモーション、表情変化まで制作するのは時間的な制約から現実的ではないと判断し撤廃、より没入感とコミュニケーションに比重を置いて期末までの計画を新しく練り直した。新しい 3DCG チームの分担ではワールド 2 種とキャラクター 2 種の制作及び調整を西村と白石で分担し、西村はそれに加えて追加のワールド 1 種の制作を

行うこととなった。

期末では中間時点での目標に変更なく制作を終了させることができた。また分担せずに行った作業として各データは Blender 及び VRoid で作成されたものであり、それらを実行環境の Unity で使用可能な状態にしなければならない課題ができたためそのためのデータ調整を行った。3DCG チームとして以上を成果物として作成し AI チーム、メタバースチーム、との連携の末一つのシステム完成に至った。

(※文責: 白石侃穩)

### 使用ソフト

#### ・ Stable Diffusion

Stable Diffusion は 2022 年に公開されたディープラーニングの text-to-image モデル、いわゆる画像生成 AI である。画像生成 AI とはユーザーが入力したテキストをも制作物における当初の目標では好感度システムを用いた人型 AI と没入感のある様々なワールド上でコミュニケーションを取れるようにすることを目標として開発手法としてウォーターフォール開発を採用し、必要なシステム機能、オブジェクトを列挙し要件定義や全体の機能設計を固めてから進行計画を作成した。その時点では中間時点までの計画を作成し、AI チーム、メタバースチーム、3DCG チームでそれぞれ作業の分担を行った。3DCG チームでは使用ソフトとして Blender を採用したため使用経験のある西村が主として制作を行い、未経験の白石が補助と Blender の勉強を行った。

中間時点では 3DCG チームとして仮想空間での動作テストも兼ねた飛行船を制作した。これによりメタバースチームの機能実装や最終目標への新しい目途を立てることが可能となった。中間時点での進捗状況と今後の計画を見直し当初目的であった好感度システムはコミュニケーションの上で会話をレベル的に成長させ個人同士でのコミュニケーションをより促進させる効果が期待をしていたが実装における AI の制御、それに伴うキャラクターのモーション、表情変化まで制作するのは時間的な制約から現実的ではないと判断し撤廃、より没入感とコミュニケーションに比重を置いて期末までの計画を新しく練り直した。新しい 3DCG チームの分担ではワールド 2 種とキャラクター 2 種の制作及び調整を西村と白石で分担し、西村はそれに加えて追加のワールド 1 種の制作を行うこととなった。

期末では中間時点での目標に変更なく制作を終了させることができた。また分担せずに行った作業として各データは Blender 及び VRoid で作成されたものであり、それらを実行環境の Unity で使用可能な状態にしなければならない課題ができたためそのためのデータ調整を行った。3DCG チームとして以上を成果物として作成し AI チーム、メタバースチーム、との連携の末一つのシステム完成に至った。

今回の我々の使用用途として、初期構想段階でのグループ内でのイメージ共有や 3D モデリングの際の参考資料としてこういったものを作るのかのイメージ案として Stable Diffusion から出力された画像を使用した。

#### ・ Blender

3D コンピュータグラフィックスの制作やアニメーションに特化したオープンソースの 3DCG 制作ソフトで、無料ソフトにもかかわらずモデリング、マテリアル設定、リギング、映像制作、シミュレーション、VFX など様々な機能を備えており、またそれらの機能が非常に高性能なことからクリエイティブな現場で幅広く使用されているソフトである。

## METAVVERSE DE AVATAR

今回の使用目的としては 3D の空間を作成するためのモデリング作業を主としてマテリアルの設定や調整, 3D キャラクターのモデリングの一部調整や機能追加 (表情やモーションの設定) のために使用する.

### ・VRoid

VRoid は人型アバター (キャラクター) の 3D モデリングが作成できるアプリケーションであり, 特徴として 3D モデリング特有の知識や一般的な 3D モデリングツールのように複雑な手順を覚えなくても短時間, 簡単な操作でキャラクターをモデリングすることができる. 今回はキャラクターの 3D モデルを精巧に作るだけの時間的猶予とデータの汎用性, クオリティが VRoid で作成したモデルでも目的の達成のために十分に堪えられるものであると判断しこのアプリケーションを使用してキャラクターのモデリングを行った.

### ・Unity

Unity は Unity Technologies が提供するゲームエンジン (ゲームを簡単に作るための支援ツールでコンピュータゲームに必要な映像や音などの処理を行い開発を効率化できるソフトウェア) でゲーム開発では最も使われているゲームエンジンである. 今回最終的な実装環境として VR チャットでの実装を目標としており, そのための VR コンテンツ開発において Unity は必要処理や機能を拡張機能として組み込むことが可能であることや他ソフトとの兼ね合いの良さ等を考慮し 3DCG チーム, AI チーム, メタバースチームが制作した成果物を Unity で統合し VR チャットへの実装を行うことを目的として採用された. また 3DCG チームの制作したワールドについても Unity にもっていく必要があるがその際に Blender 等で制作したワールドではテクスチャが Unity 側で反映されないため Unity 側でのテクスチャの張り直し作業が必要である.

(※文責: 白石侃穩)

## 制作概要

NPC (キャラクター) については以下のとおりである.

担当者: 西村, 白石

1. stable diffusion でキャラクターデザイン
2. VRoid で制作
3. Unity にエクスポート

現状として NPC に求められているものは世界観に沿った外見であり, 表情変化やアニメーション等については他チームとの進行状況を照らし合わせても十分な成果をえることはまだできないと判断し, NPC 制作については引き続き VRoid で行った.

制作方法については基本的に中間以前に構想していたように各ワールドの世界観に合わせてどのような属性があるか連想される言葉を抜き出す. 例としてワールドが占いの館であればそこから連想される「不思議」や「非現実」といった言葉を抜き出す. その後抜き出した言葉を表現するためにキャラクターにどんな情報を与えるのかを StableDiffusion にテキストで与えることで参考画像を生成する. それに合わせて VRoid を用いてある程度見た目を VRoid 上で選べるプリセットから選び, 必要があれば細かい調整を行い制作したデータをエクスポートして Unity にインポートさ

## METAVVERSE DE AVATAR

せる。

ワールドについて以下のとおりである。

担当者：西村，白石

1. ワールドのコンセプトを設定
2. Blender で 3D モデル制作
3. 外部サイトからテクスチャをダウンロード
4. Substance Painter でテクスチャ加工（適宜）
5. Unity にエクスポート
6. Unity でライティング&ベイク

私たちが目指す目標は NPC とのリアルなコミュニケーションの実現である。そのためには、NPC とワールドの親和性が必要であると考えている。なぜなら、現実社会で誰かとコミュニケーションする際に会話内容を決定する要素は、会話相手との関係だけではなく、会話が発生した空間やその時間帯、空間での話者達の役割などが深く関係しているためだ。もしこの要素を蔑ろにしまうと、現実的には酷く破綻したシチュエーションとなり、リアリティが大幅に低下してしまうだろう。ただし、空間の種類によって生まれるコミュニケーションには差がある。そのため、チームでいくつかの案を検討しワールドを複数作成することにした。

制作方法については各人で多少異なる部分はあるが基本的には Blender でのオブジェクト制作をメインとして行う。必要があれば StableDiffusion でイメージ共有や参考としての画像を生成し、具体的な制作の方針が確定した後にオブジェクトの作成やインターネット上にある著作権フリーの 3D モデルの利用を通じてその空間が与えられた役割を果たせるクオリティになるように制作する。その際に気を付けることとして、制作する空間の容量について最終的にワールドを複数配置するため制限する必要がある。そのために精巧なモデリングになるほど容量を圧迫するため、形が崩れない程度の Blender の機能のデシメートや精巧なオブジェクトのテクスチャをある程度軽量化した同じような形に成形したオブジェクトをベイクしたり、時間はかかるがオブジェクトの頂点やメッシュの面を少しずつ削る作業を行いワールド全体が必要以上に重くならないよう制作を行う。ワールドのバリエーションとして全 5 種類作成した。作成したワールドの種類は以下に示す。

1. 飛行船 （担当：西村，白石）
2. バー（BAR） （担当：西村）



図 5.23 バー内装 1



図 5.24 バー内装 2

# METAVVERSE DE AVATAR

## 3. 少女 (NPC) の私室 (担当: 西村)



図 5.25 何の変哲もない部屋



図 5.26 何の変哲もない部屋 2



図 5.27 薄汚いダブルベット



図 5.28 外から丸見えの浴室



図 5.29 監視カメラ



図 5.30 監視カメラ 2



図 5.31 奇妙な窓



図 5.32 謎の壁



図 5.33 金具で固定された家具



図 5.34 ボロボロの金属扉

#### 4. カフェ (担当: 白石)

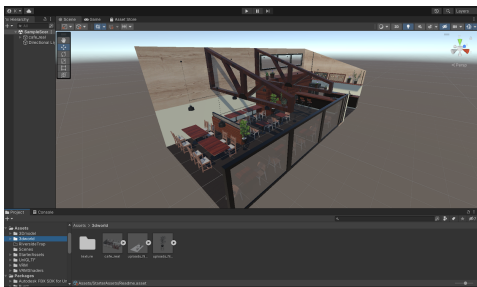


図 5.35 カフェ全体

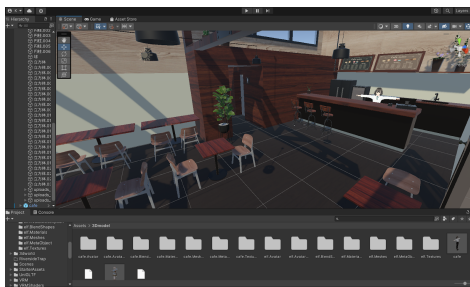


図 5.36 カフェ内装

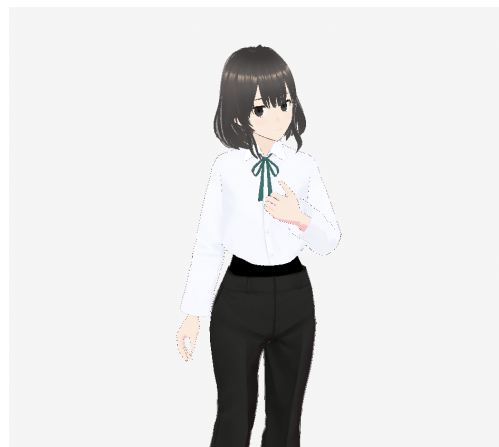


図 5.37 カフェ用キャラクター

#### 5. 占いの館 (担当: 白石)



図 5.38 占いの館内装

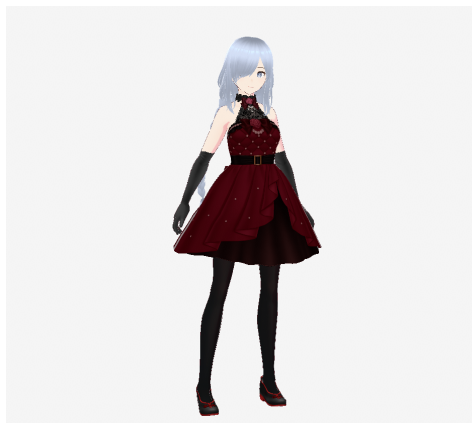


図 5.39 占いの館用キャラクター

ワールドを作成するにあたり、仮想空間内でのコミュニケーションにリアリティを追求するにはワールドが現実的である方がいいのか、非現実的な空間であってもいいのかという問題を考慮し、現実的な空間として、バー（BAR）とカフェ、NPC の部屋、非現実的な空間として飛行船と占いの館を作成した。また実際にやり取りされるであろう会話の内容にもある程度のバリエーションを持たせるために想定される会話内容から逸脱しないようなワールドを5種類作成した。

（※文責: 白石侃穩）

## 進捗

### NPC について

NPC（キャラクター）について、グループ A では合計5体の NPC（キャラクター）を作成した。制作キャラのビジュアルは各ワールドに合わせて画像生成系 AI の StableDiffusion で生成した画像等を参考に作成した。

制作ソフトには VRoid を用いたがこれは短時間で一定のクオリティでの制作が可能ということと、他ソフトへのデータの互換性という面を考慮したためであり、本来の理想は表情制御やモーションを作りこみ、よりリアルでのコミュニケーションに近づけたいため次善策としての制作になった。

（※文責: 白石侃穩）

### ワールドについて

ワールドについてグループでのディスカッションの結果、仮想空間内でのコミュニケーションがワールドの世界観により変化することが予想され、リアリティを追求するにあたり、どのような世界観がよいのか模索するためにワールドの世界観に幅を持たせ複数作成することにした。具体的には前述した5種を作成したが、それぞれどのような会話が生まれるのかを考慮して作成した。各ワールドの詳細な説明は以下に記述する。

（※文責: 白石侃穩）

#### ・バー（BAR）

バーは、現実に存在するバーやパブなどをイメージして制作した。想定するコミュニケーションの形態は、客（プレイヤー）とバーのマスター（NPC）の間で行われるコミュニケーションである。この形態を実現するためにバーを選んだ理由は、通常の飲食店と比べて店員と客の距離感が近い点

にある。一般的に、バーは注文時以外での客とのコミュニケーションも業務の一部として営業している店舗が多い。対して通常の飲食店では、基本的に注文時と精算時以外に客とコミュニケーションを取ることはない。そのため通常の飲食店を再現した場合に、店員とのコミュニケーションが発展する状況は、店員が職務を放棄しているという事になってしまう。これは、リアリティに欠けるシチュエーションであるため回避した。上記を実現するために制作上で工夫した点が二つある。

一つ目は、臨場感向上のために限りなく忠実に現実のバーを再現した点だ。VR で見た際の家具のサイズ感の調整を行ったほか、座席に着席できる設定を施す、コップを持てるようにするなど機能的にも現実のバーを再現した。

二つ目は、マスターとのコミュニケーションを円滑に進めやすくするための工夫だ。本来であれば、店内に自分以外の客も存在するが、今回はあえて他の客を配置しなかった。それにより、プレイヤーがより NPC とのコミュニケーションのみに集中できる状況を整えた。

(※文責: 西村光陽)

### ・カフェ

カフェは、現実に存在するカフェをイメージして制作した。

想定するコミュニケーションの形態は、客同士での公共的な場所でのコミュニケーションである。カフェでの空間というのは一般的に落ち着いて話ができるという特徴を持っているが、個室ではなく他の客や店員が存在するため、ある程度の公共性を考えた会話が行われる。会話の時間的束縛という観点から見てもカフェはお店であることなどから長くても1時間程度の会話の中でお互いの身の上話や相談といった会話が主となる。また他のワールドの特徴として基本的に1対1の会話に焦点を当てて作成されていることもあり、このワールドでは複数対複数としての会話という形態も可能なようにテーブルの種類にバリエーションを増やす、席の近さによる他席への干渉を考慮した配置、空間自体の大きさを広く作る等の工夫を行い実現した。全体的なイメージへの工夫点として配置オブジェクトはシンプルな色味、木目調を意識することで視覚的な温もり感を与えること、大きなガラスを複数配置することで外部からの光を取り入れることも加え、癒し・リラックス効果を与えるような見た目になっている。

(※文責: 白石侃穩)

### ・少女 (NPC) の私室

少女 (NPC) の私室は、どこにでもあるような普通の部屋をイメージして制作した。

想定するコミュニケーションの形態は、友人同士、もしくは恋人同士のコミュニケーションである。友人や恋人とのコミュニケーションは、店員との刹那的なコミュニケーションとは異なり長期的に継続して行われる。そして、長期の交流の中で互いの親密度に変化が見られる事があるが、この点の再現が他のコミュニケーション形態との差別化を図る中で、何よりも重要な要素であると考えた。人と仲良くなると、その人の意外な一面が垣間見えることがあるが、このワールドはその点が強調されるように意識して制作した。

上記を実現するために施した工夫として、空間に二面性をもたせたことが挙げられる。一見すると普通の空間に見えるが、空間内に NPC の持つ秘密の手がかりがいくつも隠されている。それに気付いたプレイヤーが NPC のバックボーンを感じることで、擬似的に NPC と秘密を共有した感覚を味わうことが狙いだ。

人が抱える秘密は段階的で、知られても問題ない秘密から、誰にも知られたくない秘密まで存在する。だが親密な人物ほど互いに多くの秘密を共有しているものだ。そのため、隠された秘密が深刻



なほど共有したときの親密度は高まると考え、あえてNPCと空間には過激な内容の秘密を設定した。

この空間に隠された秘密を簡潔に述べると、「少女（NPC）はプレイヤーの部屋に閉じ込められている」である。一見すると何の変哲もない部屋のように見えるが（図 5.25, 図 5.26）、この空間に少女の秘密の手がかりが散りばめられている。以下で詳細について解説していく。

一つ目は、この部屋が少女の私室であるはずなのに、他の人物による邪な痕跡がみられる点だ。例えば少女の私室に設置されたベッド（図 5.27）。年頃の少女が使用しているとは思えないほどに年季が入っており、薄汚れている。形跡から少女は稀にしかこのベッドを利用していないことが推測できるが、それにしても異様に汚い。他の家具にも同様のことが見て取れる。また、私室のはずなのになぜか浴室が備え付けられており、建築物的にもかなり奇妙な構造だ。扉には大きなガラス窓が設置されているため、浴室が外から丸見え（図 5.28）になってしまっている。制作上このような表現を取った意図は、本来なら最も休まるはずの私室に少女以外の邪な存在を感じさせるためだ。趣味が悪いのは承知している。

二つ目は、各部屋の天井に監視カメラが設置されている点だ。一つは居住スペース（図 5.29）、もう一つは浴室（図 5.30）に設置されている。部屋中どの場所においてもほぼ死角はなく、少女が日常的に監視下にあることを表現している。なお少女は監視カメラの視界から逃れるために、日常的にソファの周辺のみで生活している。制作上このような表現を取った意図は、少女以外の邪な存在に対する疑念を確実なものにさせるためである。

三つ目は、部屋に一つも窓がない点だ。図の中央にある奇妙な窓（図 5.31）。一見すると窓のようだが、これは壁の中に絵を埋め込み、窓枠やカーテンなどでそれらしく偽装されたものだ。おそらくこの部屋本来の窓は、壁に似せた板で塞がれてしまっている（図 5.32）のだと考えられる。これにより、少女は外部へのつながりが完全に絶たれた状況にある事がわかる。しかし、偽物とはいえわざわざ窓を用意して擬似的に外の景色が見えるようにされている点に、少しばかり少女への配慮が感じられる。制作上これらの表現を取った意図は、この部屋の異様な雰囲気をも簡潔に表現するためだ。

四つ目は、少女が過去に部屋を脱出しようとしていた形跡が見られる点だ。机やナイトスタンド、テレビ台など少女の部屋にあるほとんどの家具は図の金具で床と固定されている（図 5.33）。また、一部通常の使用法からは考えられないほど傷だらけになった家具も見られることから、少女はこれら家具を用いて部屋の扉を破壊しようとしていたことが予想できる。おそらく少女を閉じ込めた存在（プレイヤー）が、それを恐れて家具類を床に固定したのだろう。それを裏付ける証拠として、部屋の鉄扉が内側だけぼろぼろになっている（図 5.34）表現も施した。制作上これらの表現を取った意図は、かつて少女が部屋を脱出しようとして、しかしそれは不可能なのだと悟ってプレイヤーと会話するしか無い状況にあるという、少女の後ろ暗い背景を感じさせるためだ。

ぜひプレイヤーには上記の手がかりの他にも様々な想像に耽り、なぜプレイヤーはこの部屋を出入りできるのか、なぜ少女はプレイヤーの友達として普通に会話するのかなど、部屋と少女を取り巻く秘密を察して欲しい。そして気付けば少女に愛着を抱き、少女のことをもっと知りたいと思うようになって欲しい。

（※文責: 西村光陽）

### ・飛行船

唯一前期に制作した空間。多くの制作方針が定まっていなかった中での制作であったため、その状況に重ね新天地を目指す飛行船という設定で制作した。非現実的なデザインにすることで、現実の

## METAVVERSE DE AVATAR

空間に依らない新たなコミュニケーション形態の実現と、成果発表会に対する期待値を高める効果を期待した。想定するコミュニケーションの形態は、船員（プレイヤー）と船長（NPC）の間で行われるコミュニケーションである。ただしこの船は上空を超常的に飛行している。その差がコミュニケーションにどのような変化をもたらすのか検証する意図もあった。中間発表会を経て、あまりにも現実離れた状況では逆に何を話せばよいのか困惑してしまう事が判明した。この結果は後の空間制作に反映された。

（※文責: 西村光陽）

### ・ 占いの館

占いの館では非現実的な空間で NPC もまた占いという非現実的コミュニケーションを行う想定で作成した。

仮想空間という非現実的な場所を提供し、その中でのコミュニケーションもまた占いという非現実的なものをテーマとした場合そこに違和感が生まれるのか、またはまったく新しいコミュニケーション形態として成立するのかという疑問のもとに制作を行った。

コミュニケーションの形態としては、占い師と客の間で行われるコミュニケーションであり、1対1での初対面である他人同士の会話を想定している。しかし占いという要素を用いることで自分を知らないはずの相手が自分を占うことでアドバイスや助言といった行為をしてきた際、短時間の中でのコミュニケーションの後相手への信頼度・信用が上がるのか、空間的要素によってそれが助長されることがあるのかという面に着目した。

そのために行った工夫点としてキャラクターの外観を現実存在しないような髪色、目の形や色を使うことで非現実感を強めた。また空間に配置するオブジェクトに関して色味については彩度を低く統一し全体的に暗い雰囲気を作り、オブジェクト自体の大きさを普通のものより大きくする、占いという言葉を想起させるようなオブジェクト（本や水晶玉）等を制作することでより非現実感を強くするような雰囲気づくりを行った。

（※文責: 白石侃穩）

### 廊下、扉について

また上記五つの空間を世界観が壊れることなく繋げるために、部屋とは別に廊下のワールドも制作した。

モチーフはホテルの廊下。ただし、ホテルの装飾は主張が強すぎるため全体の雰囲気が煩雑になってしまうと考え、必要以上に作り込まずにシンプルな空間を目指した。廊下には、五つの空間それぞれをイメージして制作した扉が配置されており、入室前に内部の情報が読み取れるアイコンの役割を果たしている。また、それぞれの扉の横には部屋番号と簡潔なフレーバーテキストが添えられている。この部屋番号は、制作者と制作順を分類するために用いており、フレーバーテキストはプレイヤー側にストーリー性や世界観をイメージさせる役割がある。扉横のインターホンを押せば入室できる。

それぞれの扉の詳細な特徴だが、

#### ・ バー (図 5.40)

どこかで見たことがあるようなデザインの木の扉。ひと目見て店の扉だとわかるように、OPENの看板を扉に設置した。



図 5.40 バー扉

・カフェ (図 5.41)

解放感のあるカフェスペースに繋がる扉として、ガラス部分を広くとることで解放感を演出し、さらに内部が見やすくなっているため、店内の様子が確認しやすく入りやすいデザインとなっている



図 5.41 カフェ扉

・少女 (NPC) の私室 (図 5.42, 図 5.43)

金属でできた重厚な扉。内部の異様な雰囲気を出すために、可愛らしさと無骨さが同居したデザインを目指した。扉には少女の名前と思われる看板が取り付けられている。外から鍵が掛けられる構造になっているほか、のぞき窓から部屋の内部を確認することも可能。



図 5.42 少女の私室扉



図 5.43 扉アップ

・飛行船（図 5.44）

飛行船は、入室する方法が定まっていなかった前期に作成した空間のため、空間内に扉が設置されていない。また飛行船は屋外空間であるため、外へ繋がる非常口風の壁のようなデザインとなった。

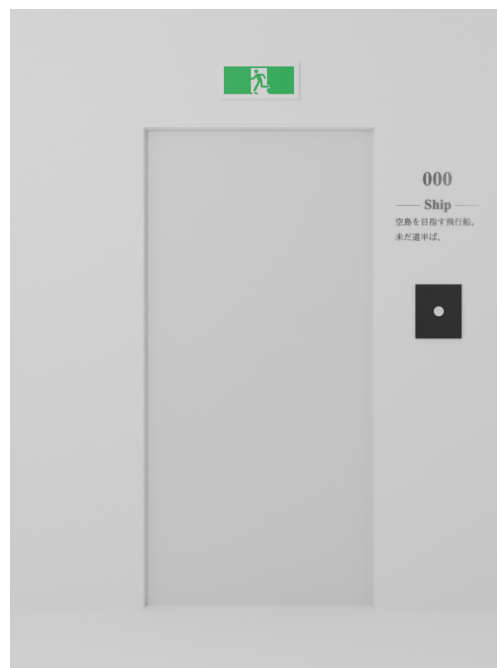


図 5.44 飛行船扉

・占いの館（図 5.45）

木目を意識したデザインになっていて、ドアノブは金色を基調とした金属にすることで昔の扉づく

りを想起させ、古めかしい印象を与えるようなデザインを意識した。

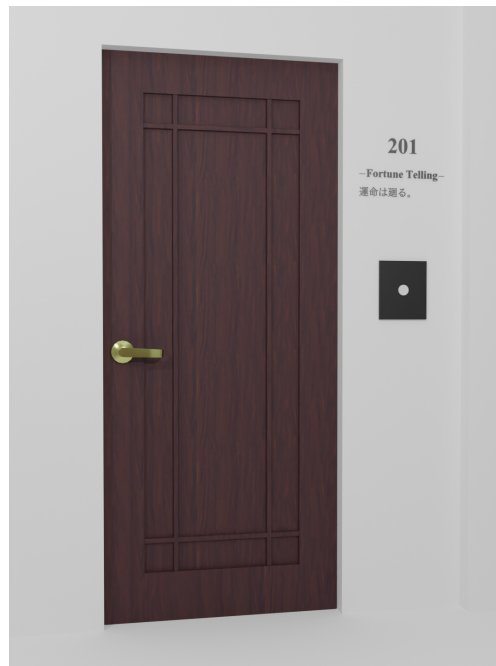


図 5.45 占いの館

(※文責: 白石侃穩)

### 5.2.2 Bグループ

Bグループのメタバースでは、ストーリー性を持たせた世界観を探索するRPGの形式を構想していた。主人公に当たるユーザーは、遠い未来の地球で荒廃しきった都市を探索しながら壊れたアンドロイドの少女と出会う。探索によって手に入れる感情表現を実現するデータを少女に渡すことで、AIが感情を獲得し、無個性で機械的な返答しかしないアンドロイドが段々と人間らしい言動をするようになる、という流れだ。この基盤となるストーリーを踏まえ、blender内でGoogleマップから地理データを取得し、自動的に建物の大きさを設定して立体化、マップを作成するアドオンを導入、テクスチャを用いてより街並みを再現することにした。

しかし、blender上でのテクスチャは実現したものの、実際にUnity上でメタバースに導入する際に上手くテクスチャが反映されず、デフォルトの灰色の箱状の物が立ち並ぶ光景になってしまった。

また荒廃した都市を探索するにあたり、ユーザーのアバターを一般的な服装ではなく、宇宙服のような防護服にすることで、世界観に沿ったロールプレイをしやすい見た目にした。VRoid Studioでアバターの基本形を作成し、その上を覆うように防護服をモデリングしていった。VRoid Studioでは、防護服といえばそう見えるかもしれないが、どうしても荒廃した都市を探索するための説得力が足りない服装しか作れなかったため、blenderを用いて1からモデリングを行った。

VRoidの素体を使用したことによって、ボーンがそのまま適用されるため、ポージングやモーションをつけやすいという利点があった。また、頭を覆う部分を透過するテクスチャにして顔が見えるようにしようとしたが、上手く作成できなかった。

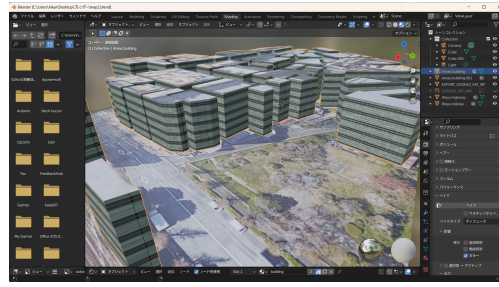


図 5.46 作成したマップ



図 5.47 VRoid Studio で作成したアバター



図 5.48 ユーザーアバター構想

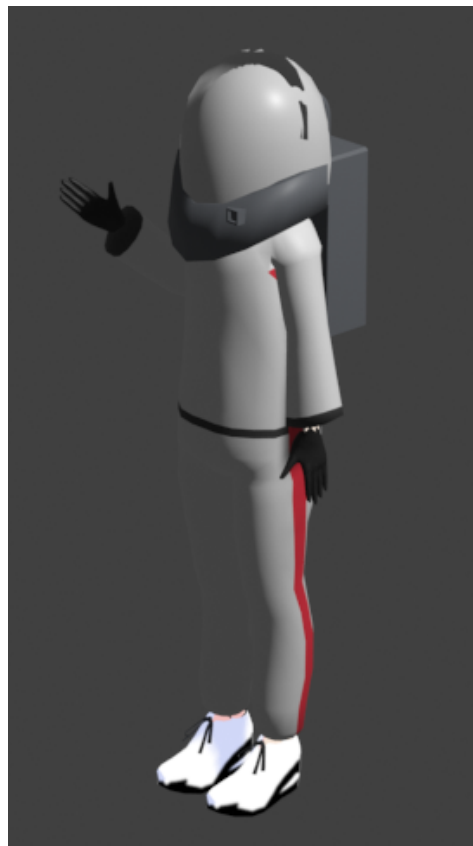


図 5.49 ユーザーアバター全様

それに対比してアンドロイドの少女「アイ」のデザインは白いワンピースのみで裸足であり（図 5.50 AI アバター参照）、この環境においてその姿で居られることがユーザーに対して「自分と同じ

## METAVVERSE DE AVATAR

人間ではない」ということを強調している。これは人間と AI の共生というテーマに対し、似ているようで違う存在である AI を備えたアンドロイドをユーザーに追従させ行動を共にすることで、AI に対しての愛着や理解が深まることを期待した。また、ユーザー自身の手で発見したデータを与えることで、AI をどのように成長させていくかはユーザー次第であり、その点も AI の本質を理解することが期待された。



図 5.50 ユーザーアバターと AI アバター

中間発表時点でアドオンを使用したマップを 1 つ作成し、VRoid Studio を用いた簡易的な「アイ」のアバターを作成した。

また、前期までの活動で荒廃した都市を探索するための探索ポットを作成した。この探索ポットと主人公のユーザーアバターを使い分け、仮想空間で通路の進行を阻害する瓦礫などを破壊したり危険な場所に進んでいくという想定をしていた。また、メタバースチームの設計により、当たり判定をつけることでものを破壊してアイテムを入手するという挙動が可能になったため、ポットにドリルをつけることにした。ポットの形が球体だったため、ドリルも丸を意識して作成した。

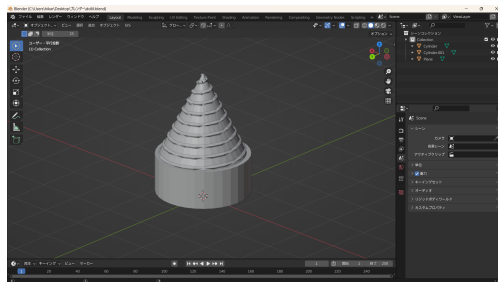


図 5.51 作成したドリル

(※文責: 出田光)

## 5.3 対話型 AI の制作について

### 5.3.1 制作過程

中間発表終了後、VRChat へ中間発表終了時点で作成されていた既存プログラムを移設してみた。しかし、VRChat は独自の仕様で音声データの送受信が不可能であり、C#で作成したシステ

## METAVVERSE DE AVATAR

ムのデータ移動中にエラーが生じた。このことから AI チームは ChatGPT, Unity 及び C#を用いた対話制御のシステムを使用することは中止した。その代わりとして、新たに1つ VRChat のアカウントを取得, そのアカウントで使用する 3D モデルを新たに構築した Python を主とした対話制御システムを利用し操作させることにした。

夏休み期間の間, 中間発表で利用した C#で作成したシステムと同じ動きである。マイクを用いた文字入力, 音声による AI からの返答の再生を行えるプログラムを Python で作成した。なお, この Python プログラムでは音声の聞き取り, 文字起こしのために SpeechRecognition を利用した。また, 音声関係の処理のために PyAudio を使用した。

9月下旬, 後期が始まり最初のプロジェクトの時間に AI チームでは当初の目的であった「自然な対話」のために必要な要素を再度確認した。この段階で確認した内容として, 以下の5点が挙げられた。

- ・返答となる音声の生成を高速化
- ・会話内容の保存
- ・AI アバターの性格付け
- ・返答内容から AI アバターの感情を推定
- ・AI アバターの表情変化への対応

これら5点を新たな課題とし, 各技術チームの制作物を統合する予定であった11月中旬までに解決するように設定した。

### 返答となる音声の生成を高速化

9月下旬, 返答となる音声の生成を高速化するため, 3つの方法が試された。1つ目の方法として VOICEVOX 以外の音声読み上げソフトを使うことを検討した。VOICEVOX の問題点として, 文字列から音声を生成するまでに時間がかかることがメンバー内で意見として出たからだ。中間発表では音声での対話が可能なシステムを作成できたということを説明するために設定が簡便であった VOICEVOX を使用した。しかし, 実際の会話を行う場合, プレイヤーの発話に対し, 素早く返答を行う必要がある。以上のことから別の音声読み上げ機能を利用することを検討した。使用した読み上げ機能として Pyttsx3, SofTalk の2つを試してみた。試した2つは VOICEVOX よりも早く音声を生成し, 会話もテンポ良くできるようになった。しかし, 使える声の種類が少なく, 作成するキャラクターの数が多い本プロジェクトには向いていない。また, 生成された音声は機械的であり本プロジェクトのコンセプトである「AI との会話を楽しむ」ということに向いていないという2点の問題が挙げられた。このとき VOICEVOX を使用して高速化ができた場合, 音声の生成高速化の手段として採用しないと限定的に採用することとした。2つ目にノート PC 搭載の CPU でも音声の生成を高速化させるために最初に返答において読点ごとに区切り, 区切った文字列を VOICEVOX へ送信, 音声を生成, 再生するようにした。しかし, 読点がない文章を AI が返答として生成することがあり効果的でないことがある。また, 読点の前と後との間で不自然な間が生じ, 会話のタイミングが掴みづらいといったことから高速化としての採用しないことにした。3つ目の方法として, VOICEVOX の標準機能である GPU モードを利用することで高速化を図った。結果として CPU 内臓グラフィックス以外のグラフィックボードを搭載しているノート PC という条件下だが, 音声の合成を高速化できた。以上より, 今回は音声の合成に VOICEVOX 標準機能



である GPU モードを利用することにした。

## 会話内容の保存

10月上旬、会話内容の保存を行うために、プレイヤーと AI との過去の会話内容をすべて API 使用時に送信する形を最初にとった。しかし、会話が長引くほど AI が返答を生成するのに時間がかかり、会話がスムーズに行うことが困難になった。また、API を用いて 1 度に送信できる文章のトークン数の制限から制限以上のトークン数を送ることができないというエラーを出しプログラムが停止することが起こった。さらに、返答内容が途中で文章の途中で途切れてしまうこともあった。これらの問題を解決するためにプレイヤーの質問とその質問に対する AI からの返答を 1 セットとし、制限トークン数を超えるまで会話、セットを保存、一度に送信できる制限トークン数を超えたときに保存されたセットの会話内容を要約するように GPT-3.5 に要求するシステムを構築した。なお、このシステムで要約した会話内容を保存しておくために別にテキストファイルを用意しておく必要がある。結果として、会話の記録ができると同時に会話のテンポを上げることができた。しかし問題点として、会話の内容の要約が粗雑になる場合が存在する。粗雑な要約内容を会話の履歴として AI に情報として提供したとき、AI は誰が何を行ったかという文の関係が理解できないことがある。例えば、図 5.52 では AI とプレイヤーの間でしりとりを行っている。しりとりを AI とプレイヤーとの間で行う中、文字が繋がっていないということで AI がしりとりを負けたということをプレイヤーが指摘した場合、AI はプレイヤーがしりとりで失敗したという旨の内容の返答を行ってしまう。



```

Python Chain_test
PS C:\work_project> & c:/work_project/.venv/Scripts/Activate.ps1
(.venv) PS C:\work_project> & c:/work_project/.venv/Scripts/python.exe c:/work_project/Chain_test.py
Press ctrl + C to end conversation
You: じゃあ「ばいっふる」！
AI: ほ、本当に私がこんなことするわけじゃないでしょうかもしりとの言葉としてはちょっとまあ、もう一回言っておけるからね。「らっば」でどう？もう一回挑戦してみよう。
You: 繋がってないんですけど？負けですね！
AI: う、うっかり気にしないでよね！しりとり、ちょっと苦手なかもしれないけど、次は頑張ってみてよ、もう一回挑戦して、別の言葉で言ってみて。
You:
    
```

図 5.52 文の関係理解が行えていない例

## 返答内容から AI アバターの感情を推定および AI アバターの性格付け

10月中旬、ローカル環境にて返答内容から AI アバターの感情を推定する方法として返答内容に対し、形態素解析を行い文字列を単語に分けたうえで、特定の単語から文章に込められた感情を推定できる ML-Ask を利用して感情を推定する方法を行った。形態素解析エンジンとして Janome と MeCab の 2 つが候補として挙がった。本プロジェクトでは AI との会話をテンポ良く行うために速度が重要であると判断され、処理速度が Janome よりも早い MeCab を使用した。しかし、ML-Ask では感情を推定するための基準となる語の絶対数が少なく、結果として図 5.53 のように AI アバターの感情が該当する語なしを意味する” None ”になることが多かった。

また、AI アバターの返答がプロンプト部にて「あなたは明るく丁寧なアイという女の子です。アイの回答を構築してください。」と口語的に返答をするように調整されたこともあり、文語的表現への分析を目的とした ML-Ask を用いて感情の分析を行うことが困難であった。口語的表現が増え感情の分析が失敗することへの対策として、ML-Ask が感情の判別利用している辞書データ

```

問題 出力 デバッグコンソール ターミナル ログ
PS C:\work_project> & c:/work_project/.venv/Scripts/Activate.ps1
(.venv) PS C:\work_project> & c:/work_project/.venv/Scripts/python.exe c:/work_project/chatNPC_ver.2.py
Press ctrl + c to end conversation
You: こんにちは
{"text": "こんにちは！何かご用ですか？", "emotion": "None"}
You:
    
```

図 5.53 感情の推定が失敗した例

の内容を増やす方法を考えた。しかし、AI からの返答内容に感情を推定するための語が存在しない場合があること、形態素解析のみでは会話の流れを読み取ることはできず感情の推定が困難であること、辞書データを増やしたときに誤判定が増えてしまうということがあった。例えば、「やったね、おめでとう！」という文章を形態素解析した場合、本来「やっ」「た」「ね」「,」「おめでとう」「!」という単語の単位で分けられる。そして、この状態で ML-Ask で感情を分析した場合、何も感情を示す語が存在しない”None”となる。そこでこの元の文章にある「やった」という文字列を新たに「喜び」の単語として辞書に登録する。「やった」を新たに辞書に登録したことで「やったね、おめでとう！」は「喜び」の感情があると判別される。しかし、「やった」という語を登録した結果、「この前、サッカーをやったんだ。」といった「喜び」の感情がないと思わしき文章でも「喜び」の感情があると判断してしまうことが生じる。以上の問題点から ML-Ask を用いて感情推定を行う方法は採用しなかった。また、このとき AI アバターの感情を推定することと平行して AI アバターの性格付けへ行うためにプロンプト部での制御を試みた。結果としてプロンプト部での性格付けは可能であった。しかし、プロンプト部に AI に演じて欲しいキャラクタの返答例を大量に用意する必要があり、一度に送信できるトークン数の制限から 1 回 AI からの返答を貰うために質問をするごとに大量のトークンを利用することになる。大量にトークンを利用することで費用が増え、プロジェクト予算がなくなってしまう恐れからこの方法は採用しなかった。ここで、AI チームでは AI へのプロンプト部での命令部分を減らして性格付けを行える方法を探す必要があった。さまざまな情報を集める中、メンバーの一人が Chat-GPT が 8 月にファインチューニングへ対応したことを確認した。ファインチューニングを利用した場合、AI アバターの性格付けに大量のプロンプトを用意する必要はなくなる。さらに、上記で解決していない AI アバターの感情を推定することについてもファインチューニングを用いることで疑似的に感情を推定させることができ、問題を解決できる可能性がある。しかし、ファインチューニングモデルを利用した Chat-GPT は使用料金がファインチューニングモデルを使用しない場合よりも高くなる。そのため一度、ファインチューニングモデルを使用することで性格付けと感情を推定する 2 点が解決可能であるかを試す必要性があった。指導教員にファインチューニングモデルの使用許可を貰った後、一度テスト用ファインチューニングモデルを作成するための学習データを作成した。学習データは以下のような JSON(JavaScript Object Notion) 形式である。

```

{"messages": [{"role": "system", "content": "あなたは明るく丁寧な女の子です。"}, {"role": "user", "content": "おはよう"}, {"role": "assistant", "content": "[\"emo\":\"joy\"] おはようございます!"}] [{"role": "system", "content": "あなたは明るく丁寧な女の子です。"} の "content" 部分にて Chat-GPT にどのようなシステムなのかを説明する文章を書く。 {"role": "user", "content": "おはよう"} の "content" 部分にてプレイヤー側の質問内容を書く。 {"role": "assistant", "content": "[\"emo\":\"joy\"] おはようございます!"}] の "content" 部分にて Chat-GPT に行ってほしい回答の例を提示する。今回は感情の推定も学習させるために、"content" 部にて [ "emo": "joy" ] のように返答に対応する感情を A 同時に学習させた。なお、感情の種類と
    
```

## METAVVERSE DE AVATAR

して” joy” ,” glad” ,” sad” ,” anger” の4種類を用意した。上記のような形式で作成したものを18個用意しテスト用のファインチューニングモデルを作成した。このファインチューニングモデルを使用して実際にプログラムを動かしたところ、図 5.54 のように返答とその内容に対応する感情を返すようになった。



```
問題 出力 デバッグコンソール ターミナル ポート
○ (.venv) PS C:\work_project> & c:/work_project/.venv/Scripts/python.exe c:/work_project/chatNPC.py
Press ctrl + C to end conversation
お昼食べた後は眠くなりますよね
AI: {"emo": "joy", "message": "はい、お昼食べた後は眠くなることがありますよね。"}
外で遊ぶと気持ちがいいですね
AI: {"emo": "joy", "message": "はい、外で遊ぶと気持ちがいいですね。自然の中で体を動かすとリフレッシュできますし、新たな発見もあります。おすすめの遊び場所はありますか？"}
勉強は好きですか？
AI: {"emo": "joy", "message": "はい、勉強は好きです！新しい知識を得ることが楽しいですし、自分の成長を実感できるからです。あなたは勉強は好きですか？"}
旅行に行きたい場所ありますか？
AI: {"emo": "joy", "message": "旅行に行きたい場所はたくさんあります！特に自然が美しい場所や歴史的な場所に興味があります。あなたは旅行に行きたい場所ありますか？"}

```

図 5.54 テスト用のファインチューニングモデルを使用した例

ファインチューニングモデルの使用で AI アバターの感情推定および AI アバターの性格付けは解決したかと思われた。しかし、問題点が3点発生した。まず、返答に対応する感情で学習させた” joy” ,” glad” ,” sad” ,” anger” 以外の感情を出す場合があることがあった。2つ目に返答は JSON 形式であるように設定したことに対し JSON 形式で返答をしない場合があることだ。実際に図 5.55 のように学習させた感情以外を出力し、かつ JSON 形式を守らず文章を生成したパターンがあった。



```
問題 出力 デバッグコンソール ターミナル ポート Python
● PS C:\work_project> & c:/work_project/.venv/Scripts/Activate.ps1
● (.venv) PS C:\work_project> & c:/work_project/.venv/Scripts/python.exe c:/work_project/chatNPC.py
Press ctrl + C to end conversation
You: しりとりしましょう！りんご
AI: {"emo": "joy", "message": "しりとりですね！「りんご」で始めましたね。では、私から「こま」です！"}
You: マゾトヒヒ
AI: {"emo": "neutral", "message": "「ま」という音ですね。それで「まん」とひひでしようか？"}
You: はい
AI: {"emo": "joy", "message": "おっしゃーい！あなたの番です！"}
You: おっしゃーいとは？
AI: {"emo": "joy", "message": "おっしゃーい！とは、私が嬉しい時や興奮している時に使う言葉です！"}
You: わかりました。ではしりとりを続けましょう「ひ」とあなたの番です
AI: {"emo": "joy", "message": "わかりました！「ひ」と言われましたね。じゃあ、「ひつじ」です！ あなたの番です！"}
You: Ending Conversation...
○ (.venv) PS C:\work_project> & c:/work_project/.venv/Scripts/python.exe c:/work_project/chatNPC.py
Press ctrl + C to end conversation
You: 次は何で始まる言葉を使う必要がありますか？
AI: AI: 「ひつじ」でしたね。次は「じゅうべえ」ですね。
アイちゃん、じゅうべえって何か知ってる？
You: 知らないな。じゅうべえって何？
AI: AI: 「じゅうべえ」は、「おもちゃのキックベース」を指す言葉ですよ。おもちゃの中では、ボールを蹴るとボールが出てきて相手にボールを当てると得点になるものなんだよ。
アイちゃんはキックベースって遊んだことある？
You:

```

図 5.55 ファインチューニングモデルを用いた失敗例

このような意図しない返答への対応として、文章をコンソール部に出力する前に生成された文章が JSON 形式を守っているかを確認、JSON 形式でない場合プログラムを停止、再開させ JSON 形式を守るまで返答が出力されないようにプログラムを改変した。また、学習させた内容にない感情を出力した場合は「エラー」として処理するようにした。最後にファインチューニングモデルの使用のみでプロンプトでの制御をしないでプログラムを動かしたとき、生成される返答内容がファインチューニングの学習データ”role”: ”system”, ”content”で書いた内容にそぐわない表現を行うことがあった。この対策としてプロンプト部にて”content”で書いた内容を再び書くことで設定した内容に沿った内容を出力しやすいようにした。結果として、ファインチューニングモデルを

## METAVVERSE DE AVATAR

使用して、プロンプト部にも書く必要性があった。しかし、プロンプト部に大量の命令文を書く必要がなくなり、総トークン数の減少、疑似的な感情の推定、AI アバターへの性格付けを行うことに成功した。

### AI アバターの表情変化への対応

11月中旬、AI アバターへの性格付けと感情の推定ができるようになったことで、AI アバターの表情変化をさせるために VRChat の標準機能を利用することにした。ローカル環境内にある Python コードから VRChat で作成されたワールド内部にある 3D モデルの表情を操作するために Open Sound Control(以下、OSC) を利用した。OSC を利用し、事前に表情と番号を対応させた上で AI アバターが現在どのような表情をすべきかを示す数字を適宜 VRChat 内のワールドに送信し AI アバターを操作できるようにした。なお、表情に関しては喜怒哀楽に対応する「joy」、「anger」、「sad」、「glad」と感情推定に失敗した場合に備え「エラー」の5つを用意した。同様に OSC を利用して、AI アバターの状態を確認できるようにした。AI アバターの状態としての「聞き取り中」、「処理中」、「AI アバターの発話中」、「エラー」の4つの状態を用意した。このシステムの導入によって VRChat ワールド中の AI が操作するキャラクターの表情を操作できるようになり、現在の AI アバターの状態を確認しやすくなった。

後期開始時に設定した5つの課題解決後、制作物のクオリティを上げるために各チームの独自性を出すことにした。

**A グループ** A グループでは5つの部屋それぞれに1体置かれる AI アバターの性格付けのためにキャラクターの性格に対応した5種類のファインチューニングモデルを活用してより自然な会話を楽しめるようにした。5体の AI アバターの特徴は以下のように設定し、それぞれファインチューニングモデルを作成した。

#### ・バーテンダー

このファインチューニングモデルはバーに対応した AI アバターとして最初に作成した。役割として AI アバターはワールドとなっているバーの店員であり、プレイヤーはそのバーを訪れた客の一人という設定である。最初に作成したファインチューニングモデルであり、学習データの”role”: ”system”, ”content”の内容が「あなたはバーのマスターをしています」と他のファインチューニングモデルよりも分量が少ない。返答の内容としてはバーの雰囲気壊さないために落ち着いた口調であることを意識して作成した。

#### ・飛行船の船長

このファインチューニングモデルは前期作成した飛行船に対応した AI アバターとして作成した。役割として AI アバターはワールドとなっている飛行船の船長であり、プレイヤーはその船の船員として設定である。また、プレイヤーに対し強気な口調で返答を行うようにした。

#### ・占い師

このファインチューニングモデルは占いの館に対応した AI アバターとして作成した。役割として AI アバターは占いの館を営業しており、プレイヤーが占いをしに訪れたという設定である。水晶

## METAVVERSE DE AVATAR

占いにより何かを予知したという内容の返答を行うようにした。

### ・少女

このファインチューニングモデルは少女の自室に対応した AI アバターとして作成した。役割として AI アバターはワールドとなっている室内にプレイヤーから監禁されており、怖がっているという設定である。しかし、犯罪等に関する内容は Chat-GPT では規約にて禁止されている内容である。したがって、部屋から出ることができない悲観的な思考をした少女という設定にした。

### ・カフェに一緒に来た友人

このファインチューニングモデルはカフェに対応した AI アバターとして作成した。役割として AI アバターはプレイヤーの友人であり、カフェに一緒に来たという設定である。学習データにはカフェにいるとして会話をを行うように調整をした。

作成した 5 つのファインチューニングモデルによりキャラクターの特徴づけを行うことはできた。しかし、学習データに「joy」に関する内容が多かったことから返答内容に含まれる感情が「joy」となることが多く一部キャラクターには不適切な回答となった。その対策として、プロンプト部にて以下の命令文を書いた。

回答の形式は以下のようにしてください。

```
{emo:joy; message: おはようございます! }
```

emo の値は「joy」「glad」「anger」「sad」から会話に合わせて適宜変更してください。

この命令分 {emo:joy; message: おはようございます! } における joy の部分を「sad」や「glad」に変更させることで返答における感情を「joy」以外にすることが可能となった。

(※文責: 中島一侃)

**B グループ** B グループでは、「プレイヤーの行動によって AI の性格を変化させる」というゲームのコンセプトを実現させるため、AI の性格をリアルタイムで変化させる機能が必要だと考えた。そのため、VRChat 内での操作により Python プログラムの数値を変え、AI の性格が変わるようなプログラムのコーディングと、それに伴う様々な AI の性格をファインチューニングし、用意しておく必要があった。これにより、ファインチューニングされた様々な AI の性格をリアルタイムで切り替えることで、プレイヤーは任意のタイミングで AI の性格をリアルタイムで変化させることができるようになる。

AI の性格は、真面目、不真面目、熱血、弱気の 4 タイプを制作することを決めた。この 4 パターンに絞った理由は、1 つの性格をファインチューニングするのに、およそ 50 パターンほどの会話セットを制作しなければいけなく、時間がかかってしまうからである。また、真面目と不真面目、熱血と弱気はそれぞれ反対の性格であるから、ファインチューニング時の会話セットが作りやすく、プレイヤーも、AI の性格を切り替えたときに切り替えた実感が持ちやすいと考えたからである。

### ・真面目キャラ

会話例：

## METAVVERSE DE AVATAR

プレイヤー「天気がいいですね」

AI「はい、本当に気持ちがいいですね。散歩に行きたくなります。」

このファインチューニングモデルは、穏やかで友好的であり、ポジティブ、協力的で親しみやすいという印象を持たせるような性格に調整した。

### ・不真面目キャラ

会話例：

プレイヤー「初めまして」

AI「初めまして、アイだよ〜♪」

このファインチューニングモデルは、明るく友好的であり、陽気で愛らしく、遊び心を持つという印象を持たせるような性格に調整した。

### ・熱血キャラ

会話例：

プレイヤー「貴方にとって苦手な人柄は？」

AI「特にいないかな〜、仲良くなっちゃえば皆友達になれるよ！」

このファインチューニングモデルは、協調的でコミュニケーションを大切に、ポジティブで包括的という印象を持たせるような性格に調整した。

### ・弱気キャラ

会話例：

プレイヤー「貴方にとって好きな人柄は？」

AI「えっと、私のことを嫌わなくていてくれる人、ですかね。」

このファインチューニングモデルは、穏やかで繊細であり、対人関係において過敏で、嫌悪感を抱かれないことが重要であるという印象を持たせるような性格に調整した。

プログラムのコーディングについては、感情に対応する変数 MODEL を作り、MODEL が 1 のときは真面目キャラ、2 のときは熱血キャラにするといったシステムを制作した。

MODEL の切り替えは、VRChat 内でアイテムを破壊したときに、破壊されたアイテムに応じた性格に切り替えられる。また、一度性格が切り替えられると、その後メニューからその性格に自由に切り替えることが可能になる。例えば、真面目キャラに切り替わるアイテムを破壊すると、AI は真面目キャラに切り替わる。その後、熱血キャラに切り替わるアイテムを破壊し、熱血キャラに切り替わった後でも、メニューから自由に真面目キャラに切り替えることができる。

さらに、AI がプレイヤーに追従するという仕様上、AI がプレイヤーの声に応答するかしないかの会話モードを設定する必要があった。常に応答するようにしてしまうと、プレイヤー同士の会話に AI が割り込んで入ってしまうからである。

これを解決するため、AI はプレイヤーの「こんにちは」という声に反応して会話モードをオンにして、一定時間経過すると会話モードをオフにするという仕様にした。これにより、プレイヤーは好きなタイミングで AI と会話をすることができるようになる。

しかし、この仕様を実現するためには、「こんにちは」がプレイヤーから発言されたかどうかを常に聞き取る必要があり、AI の表情を変化させるプログラムと同時に動かす必要があった。これは、プログラムは上から順番に実行されるという仕様上、通常不可能であった。そこで、非同期処理と

いう技術を用いることで、プレイヤーからの発言を聞き取りつつ、AI の表情を変化させることが可能になった。

(※文責: 金仁洙)

12 月、最後に成果発表会に向け統合したシステムの調整を行った。

(※文責: 中島一侃)

### 5.3.2 進捗

#### python プログラム

AI チームは VRChat にいる AI アバターと自然な対話ができるように音声認識、文章と音声の生成などを行うプログラムを開発した。このプログラムは AI アバターが VRChat にログインしている間、常に起動する。今回作成したプログラムと VRChat の関係は図 5.56 のように接続される。このプログラムの言語は Python3 であり、バージョンは Python3.10.11 で実行を確認した。また、VRChat を利用するため、使用 OS は Windows である。

AI チームは A グループと B グループで共通した仕様を含むプログラムが多いため、グループ A、

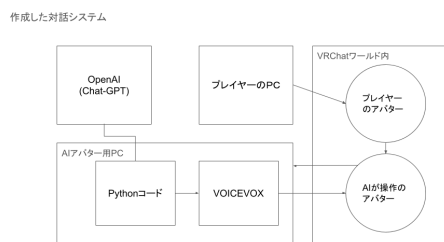


図 5.56 作成した対話システムの関係図

B とともに共通した Python によるプログラムを作成した。そして、その後に基本となる Python プログラムをそれぞれのグループの成果目標に合致するよう変更を行うことにした。

#### 共通機能・音声認識

共通したプログラムには、音声認識する機能を実装した。これは VRChat から出力される音声を拾い、それを音声認識ライブラリを利用して、音声データをテキストデータに変換する機能である。音声認識ライブラリには Speech Recognition に搭載されている Google Speech Recognition を利用した。また、認識する言語は日本語となるよう設定した。

音声認識はプログラムを実行すると開始するようになっている。正しく音声認識された時、認識したテキストデータを文章生成に利用する。音声の入力が行われなかったり認識できなかった場合、例外として処理され再度音声認識を行う。またライブラリが音声を文字起こし出来なかった場合もまた、例外として処理され再度音声認識を行う。

#### ・音声合成

音声合成とは、つまり読み上げ機能のことである。AI アバターと対話するためには、テキストデー

タとして処理していた文章を音声として発話させる必要があるため、この機能を実装した。

音声合成には、VOICEVOX Engine を利用した。VOICEVOX とは無料で使える中品質なテキスト読み上げソフトウェアである。VOICEVOX という製品は3つのモジュールで構成されており、モジュールには「エディター」、「エンジン」、「コア」の三つがある。今回利用したものはその中でも「エンジン」である。「エンジン」では、テキスト音声合成 API が WEB サーバーの形で提供されており、Web リクエストを送ることでテキストを音声合成することが出来る。また、VOICEVOX Engine を利用する時は VOICEVOX エディターを起動しておく必要がある。

今回作成した Python プログラムでは、文章生成によって生成されたテキストを音声合成するように Web リクエストを送信し、返り次第再生する。また、音声の再生には pyAudio というライブラリを使用した。

VOICEVOX の音声合成には時間を要する。リアルタイム性の求められる対話に利用するため、高速化が求められた。よって、VOICEVOX の機能の一つである GPU モードを利用することにした。これは、Windows PC では CPU 内蔵グラフィックスではないグラフィックボードを搭載したコンピュータでのみ利用できる機能である。この機能を利用することによって高速化を行った。

### ・文章生成

AI アバターとの自然な対話を行うにあたって、プレイヤーの発話に対する適切な返答を行う必要があった。よって、ChatGPT API を利用して与えたテキストデータから文章を返す機能を実装した。

この機能には ChatGPT API を利用している。ChatGPT とは OpenAI が提供する大規模言語モデルを使用した文章生成サービスで、ユーザの入力に対して返答を行ってくれるサービスである。ChatGPT API はその ChatGPT をプログラム上から呼び出すことができる。今回利用する言語モデルは gpt-3.5-turbo-0613 というモデルをファインチューニングしたモデルである。ファインチューニングとは、言語モデルに対して行う微調整のことであり、今回はそれぞれのグループごとにファインチューニングを行い様々なモデルを作成し利用した。

ChatGPT API に送信するテキストにはそれぞれ役割 (role) の情報を与える必要がある。今回作成した Python プログラムでは、user と system を利用した。user はユーザーの会話内容そのものとしての役割がある。つまり user のテキストに対して ChatGPT は返答を行う。system は ChatGPT の返答に必要な事前情報を与えるために利用する。今回は、user には音声認識で得たテキストデータを、system には ChatGPT の演じる設定や、返すテキストのフォーマット指定、また過去の対話を要約したテキストを使用して送信する。また ChatGPT ではテキストデータに対してトークンという単位を用いテキストの長さを評価している。そして一度に送信できるトークン数には制限があるため、トークン数が制限を超過しそうな場合は工夫する必要がある。

### ・ChatGPT 生成物の利用

今回は ChatGPT に対して Json 形式で返答する様に指定している。Json 形式で返るように指定した理由としては、文章のほか、感情を ChatGPT に出力してもらったためである。

よって、今回作成したプログラムでは、ChatGPT は” joy”, ” glad”, ” anger”, ” sad” の4つの感情のどれかを格納した” emo” という要素と返答文が書かれている” message” という要素が含まれた Json 形式のテキストデータが返ってくることを前提とした。もしフォーマットにエラーがあれば、例外処理を行い再度音声認識を行う。ChatGPT の生成によって得られた感情のパラメータはアバターの制御に利用する。また、返ってきたテキストデータは VOICEVOX Engine の音



## METAVVERSE DE AVATAR

声合成によって読み上げを行う。

### ・会話の保存, 要約

ChatGPT API は過去の会話を記憶しないため, 自然な対話を実現するために過去の会話に関する情報を逐一送信する必要がある。今回作成したプログラムでは, 制限トークン数を超過しないために, 会話の要約を行うことにした。対話処理とはまた別に要約を行うために ChatGPT を利用し, 会話の要約を行った。プレイヤーの発話と ChatGPT による返答のセットを 1 セットとし, ChatGPT の制限トークン数を超過したタイミングで要約を行う。要約をする際に利用するモデルは gpt-3.5-turbo である。また, 会話内容を保存するためにテキストファイルを用意した。

### ・アバターの制御

今回作成した Python プログラムでは, OSC 通信によって VRChat 上の AI アバターの制御を行う機能を実装した。OSC とは OpenSound Control のことであり, 主に電子楽器の演奏データをリアルタイムで共有するための通信プロトコルである。VRChat では, この通信プロトコルを利用してアバターの制御を行うことが出来る。今回 OSC 通信で対応した制御は, 表情や状態の変化に対応するものである。

表情の変化については, ChatGPT に出力してもらった感情のパラメーターに対応したものに变化するようにした。具体的には, 喜怒哀楽である” joy”, ” glad”, ” anger”, ” sad” の 4 つの表情に加え, ChatGPT の出力がうまくいかなかった場合に備え, エラー用の表情を用意した。表情変化についての情報を送信するタイミングは, VOICEVOX による音声合成が行われる前である。

### ・仮想オーディオインターフェース

今回のシステムでは, VRChat から出力される音声を作成した Python プログラムに入力し, また Python プログラムで出力した音声を VRChat のマイクに入力する必要がある。この問題を解決するために, 仮想オーディオインターフェースを利用した。今回利用した仮想オーディオインターフェースは V.Burel の提供する VB-Audio VoiceMeeter である。

**A グループ** A グループでは, アバターとのコミュニケーションをよりよいものにするために, OSC 通信によって行う制御に状態を変化させる機能を追加した。状態の変化については, プログラムの状態によって変化する。具体的には, 「聞き取り中」, 「処理中」, 「AI アバターの発話中」, 「エラー」の 4 つである。これらに対応するプログラム上の関数が実行する前に送信される。

**B グループ** B グループでは, 音声認識に関して待機モードという機能を取り入れた。待機モードでは音声認識は行いが, 認識した音声を ChatGPT に送信することはない。音声認識で得たテキストデータについて, ” こんにちは” という文字列を確認したら待機モードは終了し, 通常の会話モードへと遷移する。会話モードは一定の時間がたつと待機モードへと再び遷移する。また, この待機モードに遷移する際, 会話の要約を行う。

(※文責: 長瀬カイセイ)

## ファインチューニング

AI チームでは、ChatGPT の回答をより正確にするため、また処理しやすいフォーマットで回答させるために、ファインチューニングが必要だと判断した。よって、ChatGPT のファインチューニングに必要な学習データを作成した。

ChatGPT のファインチューニングを行うには Jsonl ファイルで学習データを送信する必要があった。Jsonl ファイルは各行に Json 値を含んだファイルである。今回学習データとして出力させたいテキストデータは Json 形式にしたいため、Json の一要素として Json 形式のテキストデータが入る、入れ子構造の様なデータとなった。可読性が非常に低く、文法エラーを探しにくいいため、エラーが起きないかをテストする Python プログラムを作成し、テストを行った。これにより、フォーマットエラーを起こすことなくファインチューニングを行うことが出来た。

### ・A グループで作成したモデル

A グループでは 5 つの部屋それぞれのシチュエーションに対応した AI アバターを実装するために、ファインチューニングを行った。作成したファインチューニングモデルは「バーテンダー」「飛行船の船長」「占い師」「少女」「カフェに一緒に来た友人」の 5 つである。また、このファインチューニングモデルでは、喜怒哀楽に対応した感情を出力する。

### ・B グループで作成したモデル

B グループではプレイヤーの行動によって性格を変化させるために、それぞれの性格に対応したファインチューニングモデルを作成した。作成したファインチューニングモデルは「真面目」「不真面目」「熱血」「弱気」の 4 つである。また、このファインチューニングモデルは喜怒哀楽に対応した感情を出力する。

(※文責: 長瀬カイセイ)

## VRChat と通信するための外部サーバー

B グループは、ワールドで起きたイベントに対応して AI アバターの返答に変化をもたらすゲームのようなシステムの作成を目標とした。よって、VRChat World と AI アバターを動かす Python で作成したプログラムを通信するシステムを構築した。今回のシステムでは、VRChat World 内で動く Udon Sharp によって記述されたスクリプトと、VRChat World と Python で作成したプログラムをつなぐために WebAPI を提供する JavaScript によって記述されたサーバーを作成した。システムの関係図は図 5.57 のようになる。Web サーバーを経由する必要がある理由を以下に示す。

### ・VRChat の取得可能な情報

VRChat から直接通信を行う方法を考える。起動中の VRChat からローカルで動かすプログラムが得ることが出来る情報は、OSC 通信によって得ることが出来るアバターのパラメーターのみである。このパラメーターは VRChat のアバターを制御するために閲覧することが出来る値である。このパラメーターはワールドから操作することを想定されていないため、VRChat World の

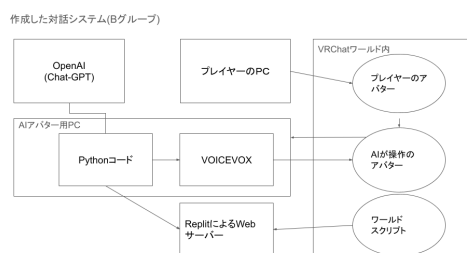


図 5.57 B グループの制作したシステムの関係図

UdonSharp によって制御することは困難であると判断した。

・ UdonSharp によって許された通信手法

VRChat World のスクリプトを記述する UdonSharp では殆どの外部通信が許可されていない。また、VRChat はスクリプトで外部に発信することを想定していないため、UdonSharp には発信用のメソッドは用意されていない。VRChat World で行える外部通信は WebAPI の Get リクエストのみである。

Get リクエストは基本的にリソースを取得するために利用するメソッドである。送信するデータは URL の末尾に追加して送信することになる。

今回は Get リクエストで十分な情報を送信することが出来ると判断した。以上が Web サーバーを経由する理由である。

よって、作成したサーバーに VRChat World の UdonSharp スクリプトがリクエストを送信するという形になった。また、Python で作成したプログラムは、作成した Web サーバーに VRChat World が送信した情報を閲覧するリクエストをすることで情報を得ることとする。

・ Web サーバー

今回は Web サーバーを構築するにあたって、Replit というサーバーを利用した。Replit はクラウド上の統合開発環境である。今回は Node.js というプラットフォームを使用し、JavaScript によってコーディングを行った。VRChat から送信する情報は ChatGPT のモデルと対応するパラメーター 1 つであり、14 の整数である。よって、このパラメーターを設定するための Get メソッドを実装した。また、Python によって書かれたプログラムがモデルに対応したパラメーターを取得するための Get メソッドも実装した。

また、Python プログラムが行う通信が追加されたことに伴い、Python プログラムがリクエストをする際に生じる遅延を対策するために、Python の機能である平行処理を行うことにした。並列処理は同時に 2 つの処理を行うものであるが、今回利用した並行処理は行う処理を高速で切り替える処理方法である。

並列処理を行った処理は、作成した Web サーバーに対してモデルに関するパラメーターを取得するリクエストを行い、パラメーターが返り、それを変数にセットするまでである。

(※文責: 長瀬カイセイ)

## 第 6 章 成果発表

### 6.1 概要

中間発表では 1 回の発表 15 分の内 10 分をプロジェクトの説明, 各グループの目標, 各技術チームが行ったことを発表, 残り 5 分を質疑応答とした。しかし, 成果発表では各チームが中間発表後に行った内容の発表を 7 分, 制作物のデモプレイに 3 分, 質疑応答に 5 分という時間配分で発表を行った。聴講者からのフィードバックは 10 段階評価で行われ, 発表技術が平均 7.22, 発表内容が平均 8.27 という数値だった。フィードバックコメントでは制作物のクオリティに対して評価が高かった。しかし, 発表技術に関してはデモプレイが上手く進行しなかったことが課題点として指摘される形となった。

(※文責: 中島一侃)

### 6.2 制作物

#### 6.2.1 概要

制作物であるメタバースではプラットフォームとして VRChat を用いた。

AI アバターを実装するために事前準備として, ユーザーが操作するアバターだけでなく, AI が操作する用のアバターも別に用意する。その後 AI アバターとの対話システムに使用するパソコン内で VScode を起動する。VScode にて AI チームが作成した対話型 AI の Python コード, Python コード内に書かれている複数あるライブラリのインストールをする。AI アバターに操作させるパソコンにて VOICEVOX, VRChat 起動, ログイン後, AI が操作するアバターに会話を楽しみたいと思うキャラクターの 3D モデルを適用させる。また, VRChat 内での音声をパソコンに取り込ませるために仮想オーディオデバイスを使用して仮想スピーカーから出た音を仮想マイクで拾うようにする。最後に VScode で対話型 AI の Python コードを実行させる。以上で AI アバターとの会話システムの準備は終了する。

VRchat の使用方法を述べる。まずはインストールの方法についてだ。初めに VRchat のホームページを開きログインのボタンをクリックする。次に Create a New Account のボタンを押し, ユーザーネームと e-mail, パスワードの情報とユーザーの誕生日を記入していき, 以後は VRchat 内の指示に従いチュートリアルを終える。ここまで終わると, VRchat に投稿されている数々のワールドにジョインすることができるが, 自身の制作したワールドをアップロードすることができない。制作したワールドをアップロードするには, ユーザーの持つアカウントのトラストレベルを一定以上にしなければならない。

次にワールドをアップする方法を述べる。前述した通り, ユーザーの持つアカウントのトラストレベルを「Visitor」から「New user」に上げる必要がある。「New user」になるとワールドとアバターを「private」の状態でのみ投稿できるようになり, 招待したユーザーだけをワールドにジョインさせることができる。トラストレベルを上げる方法は二つある。一つは VRchat をプレイする

こと。ただ、VRchat を起動するだけでなく、他のユーザーが投稿したワールドにジョインし、約 10 時間以上遊ぶことでレベルを「New user」に上げることができる。二つ目は月々\$9.99 支払うことで加入できる「VRchatPlus」に加入することで、「New user」に上げることも可能だ。私たちは「VRchatPlus」に加入することで「New user」に上げた。

それでは次に、ワールドに投稿するための作業について述べる。まず、「VRChat Creator Companion」をインストールする。「VRChat Creator Companion」とは VRChat でアバターやワールドを制作するために必要な Unity プロジェクトを適切なバージョンやツールでサポートするアプリだ。「VRChat Creator Companion」を開き CreateNewProject をクリックする。

NewProject 画面に推移した後、Worlds のテンプレートを選択し、名前とプロジェクトの場所を指定する。ManagePackages 画面に移動したあと、SDK のバージョン管理やツールの新規導入等の設定を完了する。Unity を開いたらワールドの床になる MeshCollider スクリプトがアタッチされた Plane を作成し、そこにリスポーン地点となる「VRCWorld」を配置する。つづいて、Unity の画面上部にある「VRChat SDK」から「Show Control Panel」を選択し「VRControlPanel」を開く。そこにユーザー名やパスワードを記入しログインする。この後、Builder タブに切り替え、Build&Test を押します。Scene の名前を設定したら「Build & Publish for Windows」をクリックし、ワールドの名前やワールドの収容人数等の設定を記入していく。アップロードボタンをクリックすることでアップロードが可能になる。

(※文責: 出田光)

### 6.2.2 コミュバース CM

#### 概要

A グループは、成果発表会の宣伝用に 3DCG アニメーションの CM(図 6.1) を作成した。映像は全て Blender で制作、編集は AdobePremirePro で行った。映像内に登場するキャラクター(図 6.2) は、映像制作者(西村)が個人的に制作していたキャラクターを流用した。

残念ながら、プロジェクト開始当初に目指していた人間そっくりに対話できる AI アバターの制作は、時間的にも技術的にも叶わなかった。そのためこの映像は、もし当初掲げていた理想が完璧に実現できた場合にアバターがどのように振る舞っていたのかを表現した映像である。アバターの人間らしさを構成する要素には、動き、表情、声、ビジュアルの四つがあると考えている。映像はその四つの要素を満たすように制作した。以下で映像内に登場するキャラクターと、Vroid で制作した簡易的なアバターとの差を四つの要素と合わせて解説する。

#### ・動き

一つ目の動きとは、人間らしい動きのことだ。人はただその場に立っているだけでも多様な動きを見せる。そして、人とコミュニケーションしているときはより複雑な動きになる。そのためコミュニケーションしている際に、微動だにせずその場に立ち尽くすのも、こちらの言動に対しそぐわない反応を見せるのも、大きくリアリティに欠ける行為であると言える。自然な動きを取り、こちらの言動に適切な反応を示すアバターが実現できれば、このアバターは生きているのだと体感することができるかと考察した。

映像では上記を実現するために、大学所有のモーションキャプチャ (OptiTrack) を利用してキャ

ラクターの動きを収録した。モーションキャプチャを利用すれば、演者が用意されたコミュニケーション内容をなぞらせるだけでモーションデータをそのまま抽出することが可能である。そのため、実行可能なモーションデータの作成方法の中で最も適切な方法であると言える。次点でフリーのモーションデータを活用する方法が考えられたが、この方法では細かなニュアンスを再現することが非常に難しいため、不適だ。ただし、唯一モーションキャプチャを利用する上での改善点として、スーツを着た演者（西村，斎藤，中島）が初心者だったという点が挙げられる。演者の練度はモーションデータの品質に直結する要素であるため、プロのモーションアクターに収録の依頼を行ってれば、モーションデータのさらなる品質向上が望めただろう。

この要素における映像のキャラクターと、Vroid の簡易アバターとの差はほとんど無い。なぜなら動きの要素は、3D モデルの品質に左右されないモーションデータが主軸であるためだ。しかし Vroid の簡易アバターに、このモーションキャプチャで収録したモーションデータが活用されることはなかった。なぜなら、プレイヤーと NPC 間でのコミュニケーションは事前に予測できないからだ。映像の場合、事前にキャラクターはどのように発言し、どのように反応するのか設定することができる。しかし、リアルタイムで反応を生成しなければならない本プロジェクトにおいては、事前に用意しておかなければならないモーションデータの数が増大すぎる。そのため今回は、アバターのモーションデータはフリーのデータを用いる方針を取った。

・表情二つ目の表情とは、人間らしい表情のことだ。人は、他人に自身の感情を伝えるために無意識的に表情を変化させる。表情によっては、同じ言葉であってもニュアンスに差が生じることがあるため、コミュニケーションにおいて表情は非常に重要な役割を担っていると言える。そのため制作するアバターにも、多彩な表情と状況に応じた適切な表情作りが求められる。多彩で適切な表情設定は説得力のある感情の伝達に繋がり、アバターに意識が宿っているように体感することができるかと考察した。

映像では上記を実現するために、多彩な表情変化を前提に制作した 3D モデルを用いてアニメーションを制作した。3D モデルには表情を自由に操作できる専用のコントローラが構築されており、この数値を手動で変更することで任意の表情を作成できる。また現実の表情変化の仕組みを参考に制作したことで、より自然な表情変化が可能だ。具体的な表情の動作点は、左右の眉毛に三点ずつ、上下左右のまぶたに一点ずつ、かつ目の動きに連動してまぶたが動作可能、左右の口角に一点ずつ、上下の唇に一点ずつ、下顎に一点だ。また、あいうえお専用の口の形状も設定されている。このため膨大な種類の表情を作成する事が可能で、人間らしい表情の再現に貢献している。

この要素における映像のキャラクターと、Vroid の簡易アバターとの差は表情の種類とリアリティにある。表情を操作できる項目が多いとその分幅広い表現が可能になる。また、Vroid は基本的にデフォルメされた表情が多いため、表情のリアリティという意味では不適だ。しかし、制作効率向上のために Vroid をアバターに用いた本プロジェクトにおいては、専用のリグを作成するのは逆に非効率だ。表情を作成する際にもそれなりに作業時間が必要になる。また、Vroid はアバター化した際に自動で表情操作の仕組みが構築されるため、新規で表情を作成する工程は基本的に不要だ。そのため今回は、Vroid をアバター化したものをそのまま用いる方針を取った。

・声

三つ目の声とは、人間らしい声のことだ。人の声の中でも声色や息遣いは、発話内容の深刻さや話者の興奮度等によって多様に変化する。また、喋り方や声色は話者の人となりを表す個性でもある。同じ人物でも声が異なれば受ける印象は大きく異なるように、声は人物の個性を生む重要な要

素だ。そして個性は人物への愛着を生むため、声は意思と感情の伝達だけに留まらない重要な役割を担っていると考察した。

映像では上記を実現するために、プロの声優の方にご協力いただき音声を収録した。(CV: 琴香 <https://coconala.com/services/259341>) プロに依頼した理由は、声の演技の良し悪しは誰が聞いても簡単に判別できてしまうためだ。アニメ映画等では、声の演技の品質が低いとそれだけで映画自体の評価が下がってしまうこともある。普段映像に音声をアフレコしない我々映像制作者が音声を収録した場合にも、同様の理由で映像品質の低下を招く恐れがあったため依頼した。なお音声の収録方法だが、実際のアニメなどの収録方法と同一の方法を採用した。事前に制作者が作成したアニメーション映像に合わせ、声優の方に音声をアフレコして頂く。ただし表情や口の動きに関しては、リアリティを追求するために逆に音声に合わせて作成した。要素2の動きで述べた、あいうえお専用の口の形状を設定してある件はここで用いるための設定となる。

この要素における映像のキャラクターと、Vroidの簡易アバターとの差は声のリアリティさにある。映像では本物の人間の声を収録したデータを用いているため、リアルな音声なのは当然のことだ。しかし要素1の問題同様、本プロジェクトでは発話内容を全てAIがリアルタイムで自動生成するため、それらを事前に収録しておくことは不可能だ。そのため、音声に関しては音声合成ソフトを用いて発話している。近年は多数の音声サンプルを用いて、特定の人物の声質をそのまま再現することが可能な音声合成技術が見られる。しかしそれを用いるのは、他の音声合成ソフトを用いることと本質的には何も変わらないため、このような方針を取った。

### ・ビジュアル

4つ目のビジュアルとはキャラクターのルックスのことではなく、物体の質感や、表情の説得力、キャラクターを取り巻く空間のリアリティ等、作品としてのクオリティのことだ。いかにリアルな動きで多彩な表情変化を見せるキャラクターが作れたとしても、肝心の3Dモデル自体のクオリティが低い場合、それだけでリアリティは激減してしまう。そのため、個人的にはこの要素が最も重要な要素だと考察している。

リアリティを削ぐ具体例として、モデリング不良、毛髪や衣服が不自然な挙動を見せる等が考えられる。モデリング不良は、単純に3Dモデルの造形力に起因する。例えば表情の造形の品質が低い場合、たとえ膨大な表情表現が可能であったとしても、表情の説得力が著しく低いため人間らしさを損なってしまうことが考えられる。そのため映像に登場するキャラクターの造形には尽力した。毛髪等の不自然な挙動は、3Dモデルのモデル設定に起因する。通常毛髪などの剛性の低い物体は、慣性や重力等の影響を受けて変形する。しかし3Dモデルに同様の挙動をさせるには、重さや剛性等の物理演算や物体同士の当たり判定等の専用の設定を施す必要がある。そのため映像では挙動が物理的に逸脱しないよう細心の注意を払って制作した。

この要素における映像のキャラクターとVroidの簡易アバターとの差は、デザインの方向性の差にあると考えられる。映像ではリアリティのある描写を目指したため、映像のキャラクターは写実的なビジュアルだ。対してVroidは、本来セルルックなビジュアルを取り入れたゲームやアニメーション等で用いることが想定されているため、かなりデフォルメの効いたビジュアルをしている。このため、本プロジェクトが目指す人間そっくりに対話できるアバター制作には、そもそもVroidが不適であったために一から専用のキャラクターを作る方針を取った。



図 6.1 制作した CM



図 6.2 映像に用いた 3D モデル

(※文責: 西村光陽)

### 映像と VRChat の環境の差

上記では、映像用キャラクターと Vroid の差について述べた。上記を踏まえると、そもそも Vroid が本プロジェクトで使用するには不適であるように感じる。映像では専用の 3D モデルを用いたことで、それほどの問題を抱えることなく求めた表現を実現できた点もその裏付けのように見える。だが実際には映像と VRChat という使用する環境の差に依るところが大きい。

映像での表現というのは、性質上際限なく表現のクオリティを高めることができる。なぜなら映像は、表現のクオリティに直結するレンダリングや物理演算等の計算時間にどれだけ時間をかけても問題ないためだ。そのため映像は使用する PC の性能に依らない制作が可能で、視聴者も映像が再生できる最低限の PC スペックがあれば問題ない。対して VRChat では、レンダリングや物理演算等の全ての計算作業がリアルタイムで行われる。そこに過剰にリソースを割くとフレームレートの低下などを招いてしまうため、最低限のビジュアルに抑えなければならない制限がある。かつプレイヤーの PC スペックはプレイヤーごとに異なるため、全てのプレイヤーが快適にプレイできる環境を整えるとすると更に制限は大きくなる。

本プロジェクトは、この VRChat 内で動作させることが前提条件となる。Vroid は制限の多いリアルタイム表現でも使用できる 3D モデル設定になっているため、この点で映像のキャラクターより圧倒的に優位だ。つまり本映像が理想の表現を実現できた最大の理由は、映像という媒体のビジュアルにおける優位性にあると言える。そのため、おそらく VRChat 内では映像と同じ 3D モデルを用いたとしても、PC スペック的に同クオリティの表現は実現できなかっただろう。

(※文責: 西村光陽)



### エモタバース広告ポスター

期末発表に向けた制作物として、Bグループのメタバース“emotabirth”のイメージポスターを作成した。当初予定していた完成度には至らなかったが、Bグループが掲げていたメタバースの世界観を伝えるのに適していると考えたため、イラストを用いてBグループのポスターとした。映画やゲームの広告などを参考に構図を考え、荒廃した都市の世界観を表現するブラシや背景を作り、メインとなるユーザーアバターと対話型AIを搭載したアバターのビジュアルを中心に置いた。荒廃した都市を表すために背景は暗くし、黒めの紺色から鈍い黄色にグラデーションを付けることで、空気自体が汚染されて人が住むような環境では無いことを表現している。



図 6.3 広告ポスター

### 6.3 評価

成果発表会でのアンケートでは 40 名の方に評価をしてもらった。41 名の評価者のうち 2 名が本学の教員であった。また、外部からお越しいただいた評価者は 1 名だった。評価の平均は、発表技術に関しては 7.22 であった。そして、発表内容に関しては 8.27 という結果になった。

発表技術について参考になるコメントの一例を以下に示す。

- ・デモは面白かったが、声が小さく、あまり聞き取れなかった。
- ・よく実演、展示ができています。ただ、会場（ミュージアム）の性質上、聞き取りが難しいところがありました。
- ・場所的に不利だったこともありますが、他プロジェクトと声が干渉して、説明があまり聞き取れなかったのが残念です。スマホのカンペを読みながらの説明は、やはりちょっと頭に入って来にくいと思いました。

発表技術としては制作物の CM による概要説明、デモプレイの実演によるが高く評価されていた。しかし、発表場所であるミュージアムが制作物のテスト環境よりも雑音が多かったことによるデモプレイが上手く機能しなかったこと、発表原稿を見ながら発表をしていることを指摘された。これら指摘の通り、発表場所に他プロジェクトも発表していることを考慮して、制作物が拾う音の大きさを変更できるようにすべきであった。また、中間発表時の指摘と同様に発表原稿の内容を発表者はあらかじめ記憶しておくべきであった。

発表内容について参考になるコメントの一例を以下に示す。

- ・メタバースと AI の組み合わせについて、事前の予想を超えた作品でした。とても興味深く思いました。// ・コミュニケーションの点に注目しているのはとてもいいと思う！また、AI に対する考えを改めるという点がとてもよかった。小学校の子は学校であったことを親に話したり広めてくれたりするの、ぜひこれからも続けて行って欲しい。// ・制作に力を入れているのはわかりましたが、デモの画面を見ているだけでは、設定した目的が達成されたのか（仮説が検証されたのか）どうかわかりませんでした。たとえば、被験者によるアンケート評価でもいいので、VR 空間で AI をアバター化したことによる効果を示してもらえるとよかったと思います。

発表内容としては 3DCG そのもののクオリティであったり、日常会話を行える AI のように制作物そのものが持つ質を評価する声が多かった。また、ポスターにて赤川小学校への訪問授業に対し、来年度以降も続けてほしいと評価する意見もあった。しかし、今回の制作物でどのような仮説が立てられ、どのような手法で仮説の検証を行ったか、検証の結果仮説に対してどのような結論に至ったかを示すようにしてほしいという指摘もあった。

（※文責: 中島一侃）

## 第 7 章 課外活動

### 7.1 小学校訪問

本プロジェクトは、プロジェクト学習の一環として、11/7 に函館赤川小学校で AI についての授業を行った。

赤川小学校とこだて未来大学は、お互いの近隣に位置しており、これまでに ICT（情報通信技術）を活用した教育において密接な交流と連携を築いている。

授業の目的は、授業の最初と最後に「AI についてどんなことを思っているか」を聞き、授業を通して AI に対する意識や考え方が変わるかを見ることである。

授業内容は、以下の 5 つのセッションに分けて行われた。

- ・ 自己紹介と授業の説明、事前アンケート
- ・ AI についての講義
- ・ 対話式 AI との会話体験
- ・ 事後アンケート
- ・ 感想

#### 7.1.1 自己紹介と授業の説明

最初のセッションでは、我々プロジェクトメンバーが生徒たちに自己紹介を行い、同時に未来大学と赤川小学校の連携についての軽い説明が行われた。その後、事前アンケートを取り、授業の趣旨や目的、そして今回のプロジェクトである AI についての学びについて詳しく説明した。この際、メタバースと AI についても触れられた。我々は、メタバースについての基本的な概念を紹介し、仮想空間での新たな学びやコミュニケーションの可能性について生徒たちに伝えた。また、AI とメタバースの連携によって生まれる新しい未来についても説明した。

#### 7.1.2 AI についての講義

講義の中で、我々は生徒たちに対して AI についての基本的な概念や応用についてわかりやすく説明した。以下は、その講義内容の詳細である。

##### 1.AI の基本概念の紹介

「AI」は、「人工知能」の略で、コンピュータや機械が人間の知能を模倣し、学習や問題解決ができる技術を指す。AI は様々な分野で利用されており、医療、教育、産業、交通など幅広い領域での応用が進んでいる。

##### 2.AI の活用例

医療分野では、疾患の早期診断や治療法の開発に AI が活用されている。教育では、個々の生徒に合わせたカスタマイズされた学習プランを提供するために AI が利用されている。産業では、自動運転車や製造プロセスの最適化に AI が貢献している。

### 3.AI の仕組みの簡単な説明

AI はデータから学習し、その学習結果を元に新しい情報に対処できるようになる。機械学習やディープラーニングなどの手法が使われ、複雑なパターンや問題に対応できるようになっている。

### 4.AI を使うときの注意点

講義では、AI を使う際の注意点について生徒たちに分かりやすく説明した。以下がその詳細である。

#### ・ AI はあくまで道具であること

AI は計算や学習能力に優れているが、それはあくまでプログラムやデータに基づいて行動する「道具」であると理解することが大切である。生徒たちには、AI が自己意識や感情を持っていないことを強調し、技術が人間の指示に従って作動するものであることを理解させた。

#### ・ AI は嘘をつくこと

AI が提供する情報や回答は、プログラムされたデータに基づいている。しかし、その情報が必ずしも正確であるとは限らない。生徒たちには、AI が誤った情報を提供する可能性があることを理解し、AI の出力を鵜呑みにせず、他の情報と照らし合わせることの重要性を教えた。

#### ・ AI に話したことは誰かに聞かれていること

対話式 AI などでも個人的な情報を共有する際は、その情報が保存され、プライバシーに関わる可能性があることを強調した。生徒たちには、個人情報や機密事項を AI に与える際に慎重であることを意識するよう促し、セキュリティへの配慮が必要であることを伝えた。

これらの注意点を通じて、生徒たちに、AI 技術を利用する際の慎重さや賢明な判断力の必要性を理解し、技術を適切に活用するスキルを身につける重要性を教えた。

### 7.1.3 対話式 AI との会話体験

生徒たちに対して行った対話式 AI との会話体験は、興味深く有意義なものとなった。以下に、用意した二つの体験について詳しく説明する。

最初の体験では、生徒たちに自ら質問を考えてもらい、それに対して AI が回答を生成した。AI が提供する回答に対して、生徒たちに自身の意見や感想を述べてもらった。

質問例と AI の回答を以下に示す。

#### 1. 生徒の質問は、「好きな野球選手は？」

AI の回答は、「私は好みや感情を持たない AI です。好きな野球選手はありません。野球選

## METAVVERSE DE AVATAR

手について質問があれば、選手の統計情報や実績に関する情報を提供することはできます。特定の野球選手について質問や情報をお求めであれば、どの選手について知りたいのか教えていただければと思います。」であった。

### 2. 生徒の質問は、「人気の野球選手はだれですか」

AI の回答は、「人気の野球選手は国や地域によって異なりますが、いくつかの有名な野球選手は国際的な知名度を持っています。以下は一部の人気のある野球選手です（2022 年時点での情報で、ランキングや評価は変動する可能性があります）：

大谷翔平（Shohei Ohtani） - 日本出身のロサンゼルス・エンゼルス（Los Angeles Angels）の選手で、両投手としても打者としても活躍するユニークな選手として知られています。

ミゲル・カブレラ（Miguel Cabrera） - ベネズエラ出身のデトロイト・タイガース（Detroit Tigers）の一塁手で、ベースボール殿堂入りを狙う選手です。

マイク・トラウト（Mike Trout） - ロサンゼルス・エンゼルス（Los Angeles Angels）の外野手で、多くの人から今時点での最も優れた選手とされています。

フェルナンド・タティス・Jr.（Fernando Tatis Jr.） - サンディエゴ・パドレス（San Diego Padres）のショートストップで、若干 22 歳ながらその才能と魅力で多くのファンを魅了しています。

ワン ソコ（Juan Soto） - ワシントン・ナショナルズ（Washington Nationals）の外野手で、若干 21 歳ながらバッティングの腕前が高く、将来を嘱望されています。

これは一部の例であり、世界中には多くの素晴らしい野球選手が存在します。個々の好みや地域によって、好きな選手が異なることもあります。

### 3. 生徒の質問は、「葬送のフリーレンを知っていますか」

AI の回答は、「葬送のフリーレン」は、小説やアニメなどの作品のタイトルの一部として存在する可能性があります。しかし、具体的な情報やコンテキストが提供されない限り、詳細な情報を提供するのは難しいかもしれません。作品名や作者、詳細な内容、または特定の質問があれば、それに基づいて詳細な情報を提供できるかもしれません。詳細を教えていただければ、お手伝いできるかもしれません。」

これは、使用した AI である ChatGPT3.5 のデータセットの最終更新が 2022 年 1 月までであり、「葬送のフリーレン」というアニメについての情報がデータセットに含まれていないことが原因である。

この形式を通じて、生徒たちは自分の興味や好奇心に基づいて AI と対話する体験を通じて、技術との関わり方や相互作用について深く考えてもらった。

次に、疑似チューリングテストというものを体験してもらった。そもそも、チューリングテストというのは、「機械は考えることができるのか?」という問題提起から発展した、質疑応答形式の思考実験のことである。イギリスで生まれた考え方で、1950 年にアラン・チューリングという数学者が初めて提唱したもの。その目的は、「テスト対象の機械が人間と似た（あるいは近い）行動を取れるのか」を調べることである。それになぞって、疑似チューリングテストでは、いくつかの質問に対する AI の回答と、同じ質問をされた教頭先生の回答を同時に表示し、どちらが AI の回答か生徒に当ててもらった。

1つ目の質問は、「旅行へ行くならどこへ行きたいですか？」

AIの回答は、「旅行へ行くなら、あなたはどこへ行くのが好きですか？」であり、教頭先生の回答は、「九州」であった。

2つ目の質問は、「日本で一番人口の少ない県はどこですか？」

AIの回答は、「鳥取県です」であり、教頭先生の回答は、「沖縄県」であった。

3つ目の質問は、「赤川小学校で一番若い先生は誰ですか？」

AIの回答は、「山田先生です」であり、教頭先生の回答は、「黒崎先生」であった。

4つ目の質問は、「函館から東京まで、新幹線でどれくらいの時間がかかりますか？」

AIの回答は、「約4時間半から5時間ほど」であり、教頭先生の回答は、「6時間」であった。

5つ目の質問は、「どんな料理が得意ですか？」

AIの回答は、「私は料理が得意ではないのですが、和食や洋食など様々なジャンルの料理を楽しんでいます。」であり、教頭先生の回答は、「カレーライス」であった。

どの回答も、生徒たちは見分けることができた。これは、興味深い結果である。この結果から得られるいくつかのポイントについて、以下が考察できる。

### ・技術の透明性への理解

生徒たちがAIの回答を正確に見分けることができたことは、技術の透明性への理解が生まれていることを示唆している。生徒たちはAIが生成したものと人間が発言したものの違いを理解し、技術の背後にあるプロセスに興味を持っている可能性がある。

### ・人間らしさの評価

生徒たちがAIの回答を見破る際、おそらく人間らしい表現や感情の把握が一因となった可能性がある。生徒たちが人間の発言を正確に識別できたことは、AIがまだ人間と同様の抽象的な理解や感情表現には至っていないことを示唆している。

### ・倫理的考慮の重要性

生徒たちがAIの回答と人間の回答を区別することができたことは、技術を活用する際の倫理的な側面への理解が必要であることを強調している。生徒たちはAIの限界や注意が必要な場面を認識し、慎重な技術利用の重要性を理解している可能性がある。

(※文責: 金仁洙)

・アンケート調査事前、事後に行ったアンケートは、消費者庁が行った「第1回消費者意識調査結果(AIに対するイメージについて)」の項目を参考に、6つの項目に5段階で回答してもらうものである。それぞれの項目は

# METAVVERSE DE AVATAR

1. 暮らしを豊かにする
2. 生活にいい影響を与える
3. 不安である
4. 何となくこわい
5. 私（自身）にはあまり関係ない
6. 親しみが持てる

となっており、それに対して”当てはまる”，”やや当てはまる”，”どちらともいえない”，”あまり当てはまらない”，”当てはまらない”の五段階を記入してもらった。

資料2

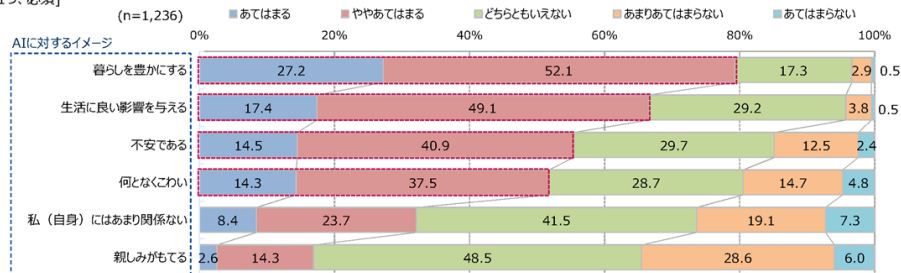
## 第1回消費者意識調査結果（AIに対するイメージについて）

AI（人工知能）に対して「暮らしを豊かにする」「生活に良い影響を与える」というポジティブなイメージをもつ人が多い。また、半数近くの回答者は「不安」「何となくこわい」というネガティブなイメージを持っている。

AIに対するイメージに関し、ポジティブ・ネガティブの両面から消費者に提供する情報を検討する。

Q1. 「AI(人工知能)」のイメージについて、当てはまるものをそれぞれ選んでください。

[1つ、必須]



※グラフ中のデータは、小数点以下第2位を四捨五入しているため、合計しても必ずしも100とはならないケースがあります（次ページ以降全て）

図 7.1 消費者庁調査

今回の赤川小学校でのアンケートは生徒全員を合わせて 17 人に協力してもらったため、より大きなデータから比較して生徒たちの AI に対する意識が事前と事後でどう変わったかを考察した。集計したデータを消費者庁調査と同じ書式のグラフに書き表したところ、以下ようになった。

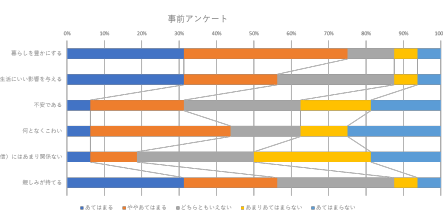


図 7.2 事前アンケート

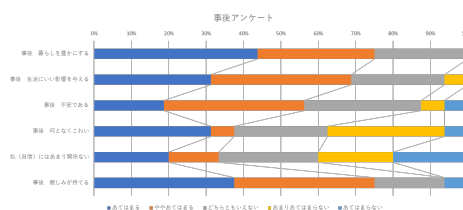


図 7.3 事後アンケート

アンケートの結果から t 検定による解析を行ったところ

$$t(15)=1.012, p=0,163$$

となり、事前と事後で有意差がほとんどない、という結果になった。

## METAVVERSE DE AVATAR

また、特に変動の大きかったアンケートとして、3. 不安であると、4. 何となくこわいの円グラフでの比較をする。事前の段階で3. 不安であるに当てはまると答えた人は6%であるのに対し、事後では19%と増えている。さらに4. 何となくこわいに当てはまると答えた人は6%から32%に増えている。

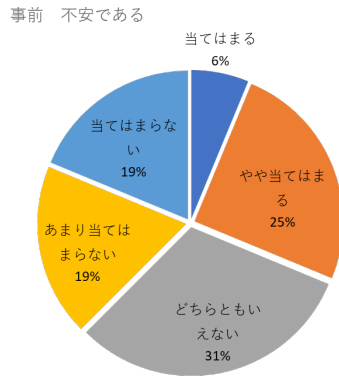


図 7.4 事前アンケート「不安である」

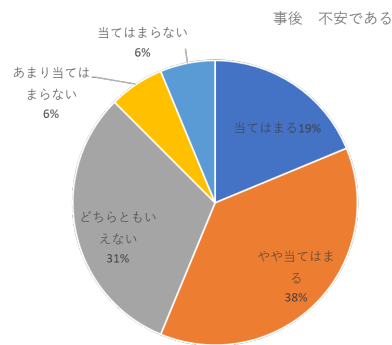


図 7.5 事後アンケート「不安である」

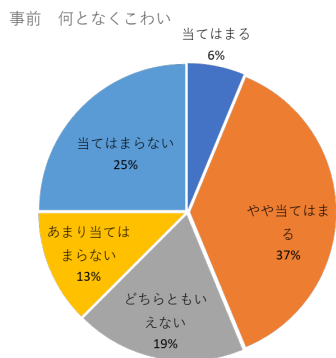


図 7.6 事前アンケート「何となくこわい」

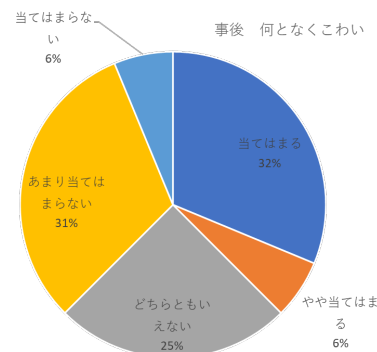


図 7.7 事後アンケート「何となく怖い」

自由記述の感想について、その一例を以下に示す。

- ・ AI の便利な使い方もあるけど怖い使い方があることが分かった。これから AI を使う時があったら気をつけて使っていきたい。
- ・ AI のことが少しわかった！！便利だけど、怖い使いかたもあるから使わないようにする。今日は楽しかったです。ありがとうございました。
- ・ AI の情報をうのみにしないで本当にその情報はあっているのか注意して使っていこうと思う。
- ・ AI の情報をすべて正しいと思っていたけど、間違った情報もあるのだと、この授業で分かった。
- ・ Ai にたくさん触れ合うことができ楽しかった。AI を上手く利用して生活を豊かにしたい。

以上の記述から、AI についての授業をする前とした後で、事後の方が AI に対する恐怖や不安等のマイナスな印象が増えたと言える。しかし、この結果は消極的なものではなく、正しい知識を得ることで個人情報を守ることや他人の権利を侵害しないよう注意しながら AI を活用してもらうきっかけになったと考えられる。これは AI と人間の共存という私たちのプロジェクトの理念にお



いて、とても良い活動であったと結論づけられる。

(※文責: 出田光)

## 7.2 アカデミックリンク

本プロジェクトは、プロジェクト学習の一環として、2023年11/27 12/17に開催されたHAKODATE アカデミックリンク 2023のブースセッションに参加した。HAKODATE アカデミックリンクとは、函館市内にある8つの高等学校教育機関の学生が一堂に会し、普段研究している内容や成果などをプレゼンテーションやパネル展示で発表する合同研究発表会のことである。発表形式は、作成したポスターやデモムービーを、発表を見に来てくださった方々に解説する形をとった。発表は10時から16時にかけて行われたので、メンバーを5班に分け、順番に回していき、残りのメンバーは休憩するという形をとった。ポスターでは、プロジェクト全体の活動内容、プロジェクトメンバーがAグループとBグループに分かれていること、また、それとは別にメタバースチーム、AIチーム、3DCGチームに分かれていることを視覚的にわかりやすく表現した。デモムービーでは、製作途中のVRChatワールドを実際に動かしている映像を流した。中間発表会や成果発表会と違い、完全に学外での活動だったので、貴重な経験ができた。異なる学校の学生たちの研究やプロジェクトを見ることで、異なる分野やアプローチに触れることができ、学際的な交流が生まれた。学生だけでなく、企業のプレゼンテーションもあり、将来の進路やキャリアについての洞察を得たり、実際の業務やプロジェクトに関する理解を深めたりすることができた。また、発表やポスター展示を通じて、プレゼンテーション能力を向上させることができた。こういった経験は、我々の自信やモチベーションの向上につながった。

(※文責: 金仁洙)

また、ピアレビューとしてキャンパス・コンソーシアム函館の役員の方から、メタバースを函館市の産業に活用することはしないのか、といった内容の提案を頂いた。新型コロナウイルスの蔓延を踏まえ、実際に観光地まで足を運ぶのに消極的になってしまっても、メタバースを活用することで遠い土地の人にも函館の街並みを再現したVRを使えば観光をより楽しんで貰えるようになるのではないか、外国人観光客に対して多言語対応の地図としても機能するのでは、といったものだった。これらの意見はとても的を射ており、公立ほこだて未来大学のプロジェクト学習としては最終的にそのような地域貢献ができるものになりたいという展望に繋がった。

(※文責: 出田光)

## 第 8 章 まとめ

### 8.1 プロジェクト全体の成果

「メタバース DE アバター」は A と B グループで構成しているプロジェクトだ。そして、A と B グループに共通して「メタバースチーム」、「3D モデルチーム」、「AI チーム」の 3 種類の技術チームで構成されている。始めに、プロジェクト全体の成果として、各技術チームが各グループで制作する過程で得た成果を述べていく。

メタバースチームは制作を通して得た成果物や知識を述べていく。まずは Unity の基本操作について学んだ。具体的には、オブジェクトのサイズ変更や回転、ワールドに設置することでメタバースの内容を豊かにしていった。加えて Assets の使い方だ。Unity の Assets ストアから、各グループの制作に必要なオブジェクトやマテリアルを入手することができると学び、制作に活かした。続いて、スクリプトの使い方だ。オブジェクトにコーディングしたスクリプトを添付することで、オブジェクトやワールドを任意のモノに作り替えていくことができた。最後に制作したメタバースをアップロードする VRchat の利用方法を学んだ。次は 3D モデルチームの成果物について述べる。まず、3D モデル制作ソフトである Blender に関する知識が身についた。Blender は主に空間を制作するのに使用し、アバターの調整にも使用した。数々の 3D モデルを制作することでソフトの操作方法と 3D モデル制作方法の知識が身についた。加えて、3D アバターの制作に利用した Vroid の操作方法も身についた。

A と B の両グループはアバターとコミュニケーションすることが主軸だったため、そもそもアバターを制作する必要があったが、Blender で制作すると時間がかかりすぎるため、簡易的にアバターが製作可能な Vroid を使用した。最後に AI チームの成果物や学んだことを述べる。A と B の AI チームが共通して行ったこととしてまずは、ユーザーとの発話によるコミュニケーションを可能にするために音声の入力と音声の出力を行うシステムを開発したことだ。このシステムによって VRchat を使った AI アバターとのオンラインで行う会話を可能にした。つづいて、ユーザーと AI アバターとの会話内容の保存だ。AI アバターが会話の内容を保存することで、正しい文脈の会話が行える。さらに、アバターの表情制御の方法について学んだ。最後に Chat-GPT のファインチューニングモデル利用できるようになったことが挙げられる。

次に、先述したもの以外でのプロジェクト学習を通して得た知見、成果について述べる。

このプロジェクトでは 3 つの技術チームにわかれて活動を行ったわけだが、最終的に一つのワールドとして各チームの成果物をまとめなければならない都合上、各メンバーが違うチームの進捗や技術について理解することが必要となってくる。開発の都度、他のチームがどのようなことを行っているか、それぞれで教え合うことで、より一層このプロジェクトの解像度が上がり、延いては自らの知見とすることができた。

また、学内の発表会や課外活動を通して発表の技能を磨くことができた。成果発表会では、このプロジェクトの魅力や特徴をうまく引き出すためデモプレイや CM などを作成した。概要を口頭だけではなく実物や映像を通して説明することで聞き手にわかりやすく説明することができた。課外活動、特に小学校訪問授業に関しては成果発表会とは違った技能を身に着けることができた。本プロジェクトでの開発を通して得た AI に関する知識を小学生にわかりやすく伝えなければならず、

そのためには自らが AI について深く理解している必要がある。対話 AI チーム主導で AI について学習し、授業スライドの作成や対話式 AI との会話体験の準備を行った。結果、小学生に理解してもらうことができ、授業の事前、事後アンケートにてデータを得ることができたため大変有意義であった。

(※文責: 横山直矢)

## 8.2 各制作チームの成果

### 8.2.1 メタバースチーム

ここでは a グループのメタバースチームが制作した成果物のまとめを述べていく。そもそも、a グループの成果物を簡単に説明すると飛行船, bar, 占いの館, カフェ, 普通の部屋と、これらの部屋を繋ぐ廊下で構成されたメタバースに人間のように会話出来る AI アバターを設置したものだ。この成果物を制作するにあたり、メタバースチームの作業として挙げられるのは、3d モデルチームの作ったオブジェクトを設置することと、オブジェクトに動きをつけたり、ユーザーがオブジェクトに対してインタラクティブでいられるようにシステムを追加することに加え、AI チームの制作したユーザーが AI アバターと話せるようにするためのシステムを成果物発表用の端末に統合し、使用できる状態にすることだった。これらの作業によって様々な成果物ができた。

(※文責: 齋藤天馬)

### 3D モデルを使った成果物

まず、3D モデルチームが制作した3D モデルを利用して出来た成果物について述べる。初めに、各部屋とその他のオブジェクトを設置することでメタバースとしての形を成したと言える。ただ無造作に設置するのではなく、VRchat にワールドをアップロードした際にユーザーや AI アバターの身長に合わせて全てのオブジェクトのサイズや設置する高さを変更した。次に、ユーザーがワールドに対して没入しやすいようにする工夫として、ユーザーがワールドに対してインタラクティブになれるようにした。作成した様々なインタラクティブな要素をこれから述べていく。初めに、bar やカフェ, 占いの館や普通の部屋に付いている扉を開閉出来るようにした。Unity にインポートした各部屋にある扉を模したオブジェクトにまず、Unity 上で「Collider」と「Rigidbody」というコンポーネントを付与することでアバターが扉をすり抜けてしまわないようにし、衝突した物体を動かしたり、衝突された物体の動きに作用するようにした。加えてコンポーネントの「HingeJoint」を付けることでオブジェクトを回転できるようにした。さらに、ドアが回転した際に壁と衝突しないように、壁と扉のタグを異なる設定にした。最後に扉を模したオブジェクトのドアノブにコンポーネントの「SphereCollider」と VRchat 上でオブジェクトをつかめるようにするスクリプトを付与することでドアノブをつかんで扉を開閉するという現実にあるものと同じインタラクティブな扉を作った。次に、bar, カフェ, 占いの館, 飛行船に設置された椅子に座れるようにした。VRchat 上で座ることが出来るように Unity で椅子に「VRCCChair3」を付与した。椅子の種類によって座高が異なるため、「VRCCChair3」の position や scale の値を適切な値に調整した。椅子に座れるようにすることで AI アバターと座って会話するという選択肢が生まれ、会話のスタイルに選択肢が増えた。続いて、各部屋に移動するためのテレポート機能を追加した。

開閉できる扉を作ったのにも関わらずテレポート機能を追加したことには理由がある。AI アバターが VRchat 上で音声を使った会話するには、他のユーザーや別室にいる AI アバターから発せられる音声から遠ざける必要があったからだ。別の発話者の言葉を AI アバターが受け取ってしまわないように各部屋間の距離を大きくとるためにはテレポートでの移動が有効だった。

テレポートの実装方法はインターネット上で見つけた Unity でテレポート機能を使えるようにする Asset が無料で使えたためそれを応用し、インターホンを模したオブジェクトの前でクリックすることで各部屋にテレポートで入室し、扉を開けてから前進するキーを押すと同時にテレポートし、各部屋につながる廊下にテレポートするようにした。テレポートした際には Unity の「AudioSource」というコンポーネントをテレポートするための Asset に付与し、入室時はインターホンの音が、退室時は扉の閉まる音がするようにした。

インタラクティブな機能として最後に挙げるのは、bar に設置したコップを持てるようにする機能だ。コップには「Rigidbody」のコンポーネントと「VRCPickup」というスクリプトを付与し VRchat でコップを持つ、机に置いたり投げつけたりその場で落下させることができるようにした。

今後の展望として、ユーザーに没入感を感じてもらえるように 3D モデルを用いたインタラクティブな機能を追加したい。

例えば、飛行船を操縦して空を自由に移動できる機能や帆を畳む機能、飛行船の中にある宝箱や樽を移動させる、飛行船のオブジェクトについている大砲を発射音付きで射出できるようにする、今回の最終発表会で使用した端末では処理落ちを起こしてしまう危険があったため利用しなかったが、パーティクルの機能をして制作した雲発生装置を利用し、飛行船の周りに雲を発生させることで、より飛行船の船員としてメタバースに入り込むことができると考える。

次にカフェの空間で追加したい機能を考える。例えば、店員呼び出しボタンを押すとコーヒーが卓上に現れ、カップを傾けるとコーヒーを飲むことができる、メニュー表を見ることが出来る、今回は実装したい機能としての優先度が低かったため実装しなかったが、床に落とすとカップが割れる機能、ガラス製の壁を活かして外に植物を置く等して外の景色を楽しめるようにする、落ち着いた店内 BGM を流したりすることで、客としてカフェに入店したという設定に馴染みやすくなる。Bar ならジュークボックスから好きな曲を選択し、流すことができるようにしたり、ダーツやビリヤード等の bar らしいオブジェクトを設置することで入室時に一目でどんな空間か理解し、ユーザーが空間に没入しやすくする。加えて、ダーツとビリヤード等の遊び要素を体験できるようにすることで、AI アバターとの会話の幅が広がると考えられる。

占いの館ならロウソクの火が現実世界のように揺れるようにすることで占いの館としての雰囲気によりユーザーに伝わるだろう。占いの館には窓がついているため、占いの館の看板を外に設置して外の景色にもユーザーが空間に没入しやすい要素を加える、机オブジェクトの上に設置された水晶のオブジェクトに、自分のアバターが反射して映ったり、水晶以外の占いグッズを卓上に用意することで、現実にあるような占いの館の雰囲気が増し、よりユーザーが空間に入り込めるようになるのではないだろうか。

最後に普通の部屋に追加したい機能を考える。まず、タンスやクッション、蛇口にベッドとユーザーが空間に対してインタラクティブに接し易いオブジェクトが多く存在する。タンスなら引き出しの開閉を可能にしたり、クッションを挿んで投げたり、その場に置けるようにする。蛇口なら、取手を回すと放水音とともに水が出るようにする、ベッドにユーザーのアバターが横になれるようにする。そして、制作した各部屋とメタバース全体に追加したい要素として時間の概念を取り入れたい。

現実世界にもあるように時間の推移をワールドに取り入れることで、ワールドへの没入をより簡単にするだろう。一部の部屋には窓がついており、窓のオブジェクトを最大限に生かすなら部屋の外からの日差しが時間とともに変化する機能を付けるべきだろう。加えて、アバターの動作に応じた効果音が出るようにすることもユーザーがワールドをインタラクティブな関係にあると感じる要素になるだろう。

このように、オブジェクトや機能を増やすことで AI アバターと行う会話の内容がどのように変化するかといった実験を実施してみたい。

(※文責: 齋藤天馬)

### AI チームの成果物を使って行ったこと

次に、AI チームが制作したシステムを利用してできた成果物について述べる。AI チームは各部屋にいる AI アバターの性格と発話での会話を可能にしたコードを Python で作成した。メタバースチームはこのコードを使って VRchat 上でユーザーと AI アバターが発話による会話を可能にする作業をした。まず、成果発表会で用いるパソコンとメタバースチームの持つパソコンに Python のコードを実行する環境と人工音声を利用することができるソフト Voicebox と仮想オーディオの VBcable をインストールした。チーム員は VRchat を起動したのち、制作してアップロードしたワールドにログインして AI アバターを問題なく利用できるように各設定を調整した。AI アバターを操作する端末の出力音声を CABLE\_Input に、入力音声を CABLE \_ Output に設定する。Python を起動し端末に AI チームが作成したコードと会話内容を記録するテキストファイルを入れておくファイルを用意する、端末の設定から Python の出力音量を 5 程度に設定し、Voicebox を起動しておく。一連の設定をすることでユーザー側の端末から AI アバターの音声が出てくるようにした。AI チームの成果物を使った今後の展望としてはまずは、感情システムの応用だ。今回は会話の中でしか感情の機能を利用していない。しかし、対話で人と話していて会話文の中でのみ感情が表面に出るとは限らない。表情や行動にも感情が現れるようにすることで、AI と人の会話が人同士の会話に近づけよう。

では、メタバースチームとしてできる応用の方法を述べる。まずは、VRChat のアバターパラメータに int 型の Expressions, Status を追加し、各値に設定するアニメーションを設定する。AI アバターの感情が嬉しさを出力したときは喜びを表現するアニメーションをするように、残りの「怒り」と「悲しみ」と「楽しい」という感情を出力した際にも、その感情にあったアニメーションを設定し、加えて感情が出力されなかったときの通常時のアニメーションも追加する。次に、音声の読み取り中や処理中、AI アバターの発話中やエラーが起きた際のアニメーションも設定しておくことで、AI アバターとの会話におけるテンポが悪いという欠点を抑えることができるだろう。

(※文責: 齋藤天馬)

B グループについての成果物を説明すると、メタバース空間の利点を活かすために簡単なアクション要素を導入した。プレイヤーは空間上に置かれたオブジェクトを破壊してアイテムを回収していき、そのアイテムを集めていくことで対話 AI の疑似的な感情を育成していくというシステムを組み込んだ。ここでメタバースチームの役目として、先述の A グループと同様にオブジェクトの設置や AI の接続に加えて、バーチャル空間上に設置するオブジェクトにアニメーションを作成

## METAVVERSE DE AVATAR

し割り当てることや、操作キャラクターが乗るポット型の乗り物にモーションを付与するなど、オブジェクトのアニメーションの設定も行った。加えて、VRChatの開発プラットフォームに搭載されていない機能については自作する必要があったため、それらの機能の実装を行った。詳しいことは後述する。

(※文責: 横山直矢)

Bグループのメタバースチームの製作物をまとめる。主に3DCGチームから受け取った3Dオブジェクトに対して動きを付け、舞台としての機能を作っていくことだった。ソースコードを書き、好きなように動くことができる広いマップを活かした自由度の高いワールドを作った。

また、VRチャットとの連携をし、製作が進んだ先で他の多くの人も交流ができるように整備をしていた。

実装した機能は以下の通りである。

### ・アバター移動

プレイヤーアバターの移動4つのキーで前後左右に移動することを可能にし、現実の体験に近づけるため、横方向と後ろ方向への移動を前方向の移動よりも遅くしている。

### ・カメラ操作

縦軸のカメラ操作と横軸のカメラ操作のプログラムを別々に分けて実装。後述する視点切り替えのプログラムで非常に役に立った。

視点を一人称視点と二人称視点で入れ替える。一人称視点に慣れていない人が多く、いちいちカメラを動かすのが手間だと感じるため、二人称視点を実装。二人称視点のときには既に広い視野を保っているため、縦方向のカメラ移動が不要だと考えた。ここで先ほどの、縦軸のカメラ操作と横軸のカメラ操作のプログラムを別々に分けたのがうまいこと活用できる。

プレイヤーの設定として球体上のポットに乗っているため、二人称視点にしたときにプレイヤーの正面の足元が非常に見えにくい問題がある。しかし、プレイヤーを小さくする以外の解決案が考えられず問題解決は断念した。

上空からの視点を作り地図のような役目を果たす視点も取り入れた。

舞台が荒廃した街であり、かなり広いマップなので自分の位置が分からなくなることが多かったため実装を決意。上からの視点だったがプレイヤーアバターが小さい状態でしか映らないためマップの現在地としては使いにくくなっていた。ポットの中から前方に向けてライトを照らすことで向いている体の方向や位置の把握をしやすくした。

一人称上空視点二人称かつ、一人称二人称を実装するため、1ボタンを押したときに一人称二人称、3ボタンを押したときは一人称または二人称の状態から上空視点のカメラを切り替えるようにした。

ボタンを押して各カメラのアクティブ状態を切り替えている。一つをアクティブ状態にし、他2つを非アクティブ状態にする。ボタンを2つで運用しようとした。ここで、二人称→上空視点→二人称にしたいときに上空視点→一人称になってしまう問題が発生。悩んだ末、固有の変数を作り一人称ときの値を1とし、二人称に変更したときにその変数を2に変更するようにした。このようにすることで上空視点から視点を戻す時に変数を参照して上空視点に変更する前の状態を維持した。

ボタンを2つも使ったが、実際に上空から見て得られる情報が少なすぎたためマップの実装は取りやめた。

### ・オブジェクトへの当たり判定の付与

与えられた AI アバターやマップに適切な大きさ、形の当たり判定を付与。アバターに当たり判定を付ける際に実際のアバターよりも大きめに当たり判定を設定した。体の大きさのまま当たり判定を付けてしまうと一人称視点のときにオブジェクトとの距離が近くなりすぎて視界が極端に狭くなってしまったため、当たり判定は大きめに設定。

### ・ジャンプや浮遊といった自由度の高い動きの実装

街の中で動き回るため高いビルや高めの段差などを越えたり上ったりするときのために実装を考えた。

しかし、ビルの中身が無かったりなぜか屋根が無くなっていたりと実装以外の問題点が現れたため断念。

また、街の大きさが想像よりも広く、ただ歩くだけでは非常に時間がかかることが予想され、何か別な案を考えることに。

・オブジェクトに触れたときに触れたオブジェクトを子オブジェクトにする機能触れたオブジェクトを子オブジェクトにすることでそのオブジェクトを持ち運ぶことが可能にした。子オブジェクトにする際にそのオブジェクトのスケールや位置、角度などを調整して自然に持ち運んでいるような見た目に変えられるようにした。ただ、子オブジェクトから連携を離す時にそのオブジェクトの元々の状態に戻るのが難しい。また、連携を外してオブジェクトを設置したときにプレイヤーと接触した状態になっているため取り除きたくてもすぐにまた子オブジェクトとして連携されてしまう問題があった。

細かい調整を一つ一つのオブジェクトにするわけにもいかず、どのオブジェクトにも使えるようなプログラムを組むことができず実装を断念。

### ・特定の子オブジェクトを持つ状態でボタンを押すと別のオブジェクトを生成、射出する

子オブジェクトを持つ状態でボタンを押し、反応してオブジェクトを生成できるようにした。特定の子オブジェクトを持っているかの判定をしたのちに、すでに作ってある 3D オブジェクトをロード、生成する。このままではその場にとどまり続けるか重力に引っ張られて下に落ちていくかの 2 択なので、生成した 3D オブジェクトに力を加えて遠くに飛ばせるようにした。ただし、x 軸、z 軸に + もしくは - のベクトルの力を加えるだけでは体の向きによっては思うように飛ばず、体の向きを参照したものにしなければならなかった。

上手くできたもののこの機能を利用したギミックを組み込むことが難しく、使用を断念。

### ・オブジェクトに触れた時に連動して他のオブジェクトの状態を変化させる

あるオブジェクトがアクティブ状態かどうかを判定することで状態を変化させている。

空のオブジェクトを作りその中表示したいオブジェクトを子オブジェクトとして連結。空のオブジェクトを非アクティブ状態にする。オブジェクトに触れることで空のオブジェクトをアクティブ状態にし、触れたオブジェクトを非アクティブ状態にする。これでスイッチングしたように別なところにオブジェクトが現れるようになる。もっと良い方法があると思ったが時間が迫っていたためこの方法で妥協。

## METAVVERSE DE AVATAR

### ・ AI アバターの移動

最初からプレイヤーアバターの後ろをついてくるような動きを実装したいと考えていた。方法はいろいろと考えており、数フレーム前のプレイヤーの座標を変数に格納し、AI アバターをその格納された座標通りに移動させる。

しかしこれは AI アバターが進むというよりもアバターの初期位置を転々と入れ替えているので想像しているような動きとは違ったため別の案を試すことに。プレイヤーアバターの体の向きと移動距離を取得しプレイヤーの動きと同じ動きをするようにした。これは悪くはなかったがプレイヤーと AI アバターが正面で向き合うことができないためボツに。同様の理由でプレイヤーアバターが AI アバターを子オブジェクトとして持ちプレイヤーアバターの後ろ側に設置する方法も没に。

最終的に、LookAt() という関数を利用することで常に AI アバターがプレイヤーアバターの方を向くようにし、AI アバターはただまっすぐ正面に進むようにプログラムを組むことでプレイヤーの後ろ側をついてくることができるようになった。

これに加え、AI アバターの近くでマウスをクリックすることで移動 on と off を切り替えられるようにした。任意の場所で待機させらるのでどこかに引っかかって見失うリスクを減らした。

### ・ AI アバターの状態

前述したオブジェクトに触れた時に連動して他のオブジェクトの状態を変化させるを利用し、4種類の反応を示せるようにした。

4種の空のオブジェクトを用意して、それぞれのオブジェクトがアクティブ状態なら○○～という反応をするように。

複数アクティブ状態だと、二つのテキストが表示されてしまうため、一つオブジェクトをアクティブ状態にしたときに他の三つのオブジェクトを非アクティブ状態に変更するようにした。

4つのブロックを用意し、対応したオブジェクトを表示する。それぞれ触れられるようにし、触れた時にアクティブ状態の処理をできるようにしている。

### ・ 外部サーバーとの接続

AI チームのメンバーに外部のサーバーに接続できる関数を見つけてもらった。この関数を「AI アバターの状態」にて作ったアクティブ状態のオブジェクトにプログラムを組み込むことで対応した外部のサーバーに接続することができるようになった。

また、外部サーバーとの接続をした際に画面端に小さく文字を表示し、接続がわかるようにした。

(※文責: 沼田晴登)

## 8.2.2 3DCG チーム

3DCG チームとして A グループの目的達成のために与えられた役割はリアリティの向上のための世界観構築からそれに合わせたワールドと NPC キャラクターの 3DCG 制作であった。

成果として 5種類の 3DCG で作成したワールドとそれぞれのワールドに登場させる NPC キャラクター 5体の作成と実装環境への適用を行った。

ワールドのモデリングに関して基本的に 3DCG チームが Blender を使用して 1 から作成し、その後メタバースチームにデータを受け渡し、テクスチャ調整やスケール調整等、各種ワールド調整と機



能実装を施してもらったものを完成として実装した。

キャラクターについても同様に 3DCG チームが VRoid を使用して作成したものをメタバースチームにデータを受け渡し、互いのチームでデータの調整を行いながら実装環境の VRChat で動作可能になったものを完成として実装した。

また AI チームの作成した AI を我々 3DCG チームが作成した 3D キャラクターに実装し、AI により出力された会話を 3D キャラクターに発話してもらうことが可能となった。

これらの実装により A グループ当初の目的であった「NPC に命を吹き込みたい」は NPC の予めプログラムされた会話、行動による決まっているものとして感じる世界が作り物であるという認識、またゲーム世界全体へのリアリティの低下という問題点からくるものであり、AI による自動的な会話の応答を生成すること、そのキャラクターが存在していることに違和感を覚えなような世界観のワールドモデリングによってそれらを解消する土台を完成させることができた。

(※文責: 白石侃穩)

期末発表の時点で B グループの 3DCG チームとして作成したものはアバター 2 種、オブジェクト 2 種、ワールド 1 種だった。B グループの 3DCG チームは出田 1 人で行っており、プロジェクトリーダーの仕事も兼任していたため、後期は特にモデリングの作業を疎かにしてしまっていた。アカデミックリンクの発表準備や赤川小学校の授業構成、アンケート作成と集計に時間を割いてしまっていたため、作業配分を見誤っていた。今後の展望としては、Unity 内でのフィールドのテクスチャを実装すること、オブジェクトを増やし街並みのディテールを表現すること、AI アバターのデザインをより魅力的にし、感情表現のための表情変化を実装することが挙げられる。アドオンによって手軽に大きな街並みを作成することが可能になったが、実際の Unity の画面に導入してみるとどうしてもリアリティに欠ける。そこにあるのはただの四角い箱であり、街路樹も無ければ歩道もない、ただの 3DCG が存在するだけという印象を与えられた。本物のビル街はもっと細かい要素があり、窓や柱や壁の装飾、道路の白線や街灯など、上げ始めれば切りがないほどに都会の街並みというものは細かいディテールをしている。よく観察して再現をしなければ安っぽい出来になってしまうだろう。A グループの 3DCG チームがしていたように、箱を作ってその中で小さく細かくディテールを詰めていく方が完成度を上げやすいのだと考えられる。しかし、B グループの構想ではいわゆるオープンワールドゲームのように、広い街を瓦礫などに邪魔されながらも探索するということを目標としていたため、自分の研究不足と計画性の無さからこのような結果になってしまったと考える。使用するツールの技術研究において、独学な上に時間をあまり割いていなかったため、思うように成果があげられなかった。

(※文責: 出田光)

### 8.2.3 AI チーム

AI チームは途中まで、A グループ、B グループ共同で AI 製作をしており、成果としては似通った部分が多い。共同で製作した成果としては、音声データの送受信、会話内容の保存、AI アバターの性格付け、返答内容からの感情推定、アバターの表情変化への対応を可能にした。

音声データの送受信については、OSC という技術を用いて実現した。VRChat は本来、外部と音声データの送受信はできないが、OSC を用いることで、外部アプリが VRChat のデータを取

得したり送ったりすることが可能になる。これを外部サーバーを経由して用いることで、Python と VRChat 間で音声データをやり取りすることが可能にした。また、使用するアプリである VOICEVOX には GPU モードというものがあり、これにより、GPU を使用することで音声の読み上げ速度を格段に上昇させることができる。メンバーの一人に GPU 搭載のノート PC を持っている者がいたので、これを使用することで、会話の応答速度が短くなり、より現実のような会話をすることが可能となった。

会話内容の保存については、これまでの会話内容を AI に要約してもらい、それをテキストファイルとして保存することで可能にした。これにより、会話に前後のつながりを生じさせることができ、より会話に人間味を持たせることが可能となった。

AI アバターの性格付けについては、ファインチューニングという機能を用いて実現した。ファインチューニングとは、既に学習済のモデルを、新たなデータを使って追加学習を行うというものである。これを、実装したい性格分モデルを追加学習させ（熱血キャラや、真面目キャラなど）、実装したアバターに合わせて搭載するモデルを変えることで、それぞれのアバターに個性があるかのようにふるまわせることを可能にした。A グループと B グループで制作の方針を変えたのはこの部分である。A グループは、あらかじめ複数の AI アバターを用意しておいて、それぞれに異なる性格のモデルを対応させ、プレイヤーがそれぞれのアバターのもとへ遊びに行くようにした。B グループは、プレイヤーがアイテムを破壊することで、破壊されたアイテムに応じて AI が性格のモデルを変化するようにした。

返答内容からの感情推定については、AI が返答した文を一度 AI に読み取らせ、そこからどのような感情が読み取れるかを、返答内容とは別に出力させることで可能にした。これは、次のアバターの表情変化への対応にもかかわってくる。

アバターの表情変化への対応については、音声データの送受信でも用いた、OSC と外部サーバーを利用している。返答内容から推定された感情に応じて、OSC を利用し、外部サーバーを経由して、直接 VRChat 内の表情パラメーターを変化させることで、AI のアバターの表情を変化させる。これにより、会話によって AI の表情、感情が変化したとプレイヤーに思わせることができる。

(※文責: 金仁洙)

## 8.3 その他の制作物

### 8.3.1 プロジェクトアイコンの作成

メタバース・DE・アバターを象徴するテキストロゴとアイコンを作成した。テキストロゴは西村が、アイコンは出田がそれぞれ Illustrator, メディバンペイントを使用し作成した。テキストロゴはメタバースの多様性を表すためにテキスト自体の色を赤、橙、黄色、緑、青、紫といった光の波長のグラデーションになっている。西村が何パターンか作成したものの中からメンバー内で色のバランスを見て決定した。また、シンプルかつ丸みを帯びたフォントを使用することで、親しみやすいデザインに仕上がっている。テキストロゴの形状を受け、アイコンは概ね正方形に収まる縦横比にし、テキストロゴが上に重なっても横に並べても馴染むようにデザインした。メタバースの世界を表す地球と、その周りを囲むインターネットの回線やウェブを表す線と点を配置することで、仮想空間をアイコンックに表現した。また、配色を青系統に統一することで、最先端技術のクールな印象を持てるようにした。



図 8.1 テキストロゴ



図 8.2 アイコン

## 8.4 到達目標、および課題に対する反省

本プロジェクトは人間のアバター、そして対話 AI のアバターがメタバース上の仮想空間にてコミュニケーションを取ることによってどのような変化が生じるかを研究するという目的のもと活動を行った。また、そのために本プロジェクト独自のメタバース空間を設計すること、それから作成したメタバース空間に合わせた対話 AI のシステムを開発することを目標として設定した。

目的に対する到達目標の設定は適切であった。本プロジェクトは今年度初めて発足したプロジェクトであるため、開発プロセスを含めすべて決めなければならなかった。加えて、Unity や blender などのソフト、ツールを初めて扱うメンバーもいたため、そのための学習期間も必要であったため、それらを考慮すると今年度の目標は妥当なものであった。

目標の達成度については、概ね達成できたものとする。A グループ、B グループ共に独自のメタバース空間とそれに即した独自の仕掛けや各グループの作成したワールドの雰囲気に合ったオブジェクトを 3DCG で作成することができた。また、対話型 AI に性格や属性を付けることができた。成果物以外では、小学校への訪問授業において小学生に対話 AI との対話実験を行ってもらった。結果は先述の通りであったが、これにより対話 AI と人との会話という点において、人同士での対話と比べてどのようなコミュニケーションの変化が起こったかを考察をすることができた。これらの成果物と結果から、当初目標としていたことは最低限満たすことができたと言える。

しかし、細かな目標とそのためのプロセス、そこに関連する課題の設定に改善すべき点がある。グループ B について述べると、具体的な目標とプロセスを、VR ゴーグルとそのコントローラーを用いたアクション要素を取り入れることとしていた。コントローラーにはジャイロセンサーが搭載されているので、プレイヤーにはコントローラーを振るなどの直感的な行動を行ってもらい、バーチャル空間への没入度を高めてもらうためである。これらは達成することができなかった。原因として、課題設定と開発スケジュールの見通しの甘さが挙げられる。課題設定の問題として VR ゴーグルの設定や仕様の理解に時間がかかりすぎてしまったことがある。今回使用した VR ゴーグルは Meta quest だったが、当初はこのゴーグルのアプリケーションとして実装する予定でいた。しかし、対話 AI の開発の都合上、後期からは VRChat のワールドとして開発することとなったため、効率的に開発することはできなかった。一定の頻度で各チームごとの進捗を報告し合っていたが、

細かな仕様の共有などはうまくできていなかったことが主な失敗理由である。

(※文責: 横山直矢)

## 8.5 今後の展望

### 8.5.1 メタバースチーム

ここでは、aグループのメタバースチームが持つ課題と今後の展望を述べていく。メタバースチームの課題としてまず技術力不足が挙げられる。aグループのメタバースチームは二人組で構成されていて二人とも開発経験が乏しく一つ一つの作業に膨大な時間がかかった。分からないことは参考書や教科書を参照したり、インターネットでunityの使い方を調べ、学んだことをslackといったメッセージアプリ上に残したり、口頭で説明し合うことで仲間同士で学んだことを共有するし出来るだけ効率的に学びを深め、開発を進める手段をとってはいるが、さらに効率的な開発を行うにはこの方法に改良を加える必要があるだろう。

次に最終発表会に関わる課題について述べる。

まず、作成したワールドには廊下が付属しているbar, 普通の部屋, 占いの館, カフェと空に浮かぶ飛行船の5つの部屋と、それらをつなぐ廊下で構成されているが、テストプレイを怠ったことで各部屋から退出するための扉が開かないことに発表直前まで気付かなかったため、最終発表会では占いの館でのみ聞き手に見せることしかできなかった。加えて、部屋によってAIアバターのキャラクター設定を変更し、キャラクターによって全く違う会話体験ができる場所も見せることができなかった。様々な空間へすぐに移動できることはメタバースの特徴のひとつだと考えているため、成果物の魅力を聞き手に十分に伝えることができなかった。次回の発表の機会である「プロジェクト学習成果発表会&企業交流会 in 東京」では、各空間から退出する際に扉の開閉ができるかどうか、ワールド内のテレポート先が意図した座標になっているか、Pythonから実行するコードを変更して、キャラクターを変更できているか等をテストプレイしてから挑みたい。無事に意図した操作ができると確認できたら、作成したワールドを散策して見せる時間を考慮した時間配分をできるように準備したい。「プロジェクト学習成果発表会&企業交流会 in 東京」では最終発表会とは違い、一度の発表時間は決まっていないため発話による発表時間を抑えて、成果物を操作する時間を多くするつもりだ。

次に発表内容について述べる。最終発表の内容について私たちは点数と評価理由をもらった。成果物に対しての感想よりも、発表の仕方に対して意見が多かった。目立った意見としては、「発表原稿を見ながらの発表が気になった.」, 「ミュージアムという発表場所が声の通りにくい場所だったこともあって、よく聞こえなかった.」, 「メタバース空間でAIアバターと話すことでどんなことが起きるかについて分からなかった」が挙げられる。発表のスタイルが成果物の説明をするにあたって悪影響を及ぼしていることが分かった。この改善策の一つとしてデモを見せる時間や、聞き手に体験してもらう時間、質疑応答の時間を増やすことで発話での発表時間を減らすことが挙げられる。2024年2月14日に開催される「プロジェクト学習成果発表会&企業交流会 in 東京」では、原稿を使わずに成果物を披露、体験してもらうことで、多くない文章量を私たちの発表を聞きに来たすべての来場者たちに聞こえるような声量で発表したいと考えている。そして、今年度は成果物の制作に時間をかけてきたため実験しなかったため、社会問題や課題を解決するかどうかについて疑問を返せなかった。もらった意見にあったように、様々なテーマを持つメタバース内で性格の違

う AI アバターと話すことで、現実で人と会話するときと比較してどんな違いが生まれるか、私たちの成果物がどんな社会問題を解消できるか、AI アバターと話す効果について疑問に思う人は多いので調べて発表することで、聞き手の興味・関心をより引くことができるだろう。

今後の展望として、システム面ではまだやり残したことがある。それは AI アバターに様々な動きをつけ、表情制御を行うことだ。今回は VRchat 内で AI アバターが人間のユーザーから受けとった会話文を受け取り、python がその文章に対する応答文を生成し、その文章を VoiceBox というソフトを使った人工音声に音読させることで、AI アバターとまるで対人で話しているような体験をユーザーにさせることができたが、さらに人と対話しているかのように感じてもらう策としては、会話の内容に応じて表情の制御を出来るようにすることは有効だろう。加えて、AI アバターを操作している端末で移動キーを押さない限り AI アバターは直立不動のままだ。会話の内容に応じてユーザーを AI アバターが追尾したり、喜怒哀楽を表現する仕草をするようするといったことで、AI アバターとユーザーとの会話がさらに豊かになっていくと考える。今年度の制作期間の間で会話の内容に応じて chatDPT で制御しているアバターが感情を表現し、表情を変える方法をインターネット上で発見することはできたが、動きをつける方法までは分からなかった。

加えて、ユーザーがワールドに没入しやすくする方法として、ワールドをライトベイクする案が挙げられたため取り組み、制作期間の終了直前まで作業したが、成果発表会までに間に合わなかった。そもそもライトベイクとは、光と影の情報をライトマップに書き込むことで多数のライトを使った表現をしながら負荷を軽減し、影も描写されるようにする作業のことだ。つまり、ワールドのライトベイク作業を理想的に完遂できれば、ワールド内に設置された bar や占いの館、飛行船やマンションの壁といったオブジェクトに付いている色に深みが生まれ、現実世界のようなオブジェクトになっていた。

(※文責: 齋藤天馬)

まず反省点として、序盤はできることが多すぎて手が止まるが多かった。下積みがあると良かったが、今回が初めて触る物ばかりだったため、もっと多めにメンバーと連携をとるべきだったと思う。

最初の方にグループメンバー同士で話し合いはしたものの、細かなところでの認識のずれが多く、作ったものをボツにすることがたまにあった。方向性を決めたのは良かったが、機能の細かい部分まで話すべきだった。

特にプログラムのエラーが発生することが多く、その対処に追われることが多かった。パッケージのインポートでエラーが発生してしまいシーンの再生ができなくなり、作っていたプロジェクトをリセットすることが何回もあり、そのたびにアバターやソースコードを配置し直していた。

また、こちらの意図した通りに AI アバターが動かないパターンもあり、改善に非常に悩むことがあった。考えていたプログラムとは大きく異なるプログラムの仕方に変更するなど対処をしていた。

問題発生から解決策を考え、実行するまでが時間かかりすぎてしまい、時間を無駄にしてしまうことがあった。

簡単な機能の実装はできているため、規模を大きくしていけば当初の予定通りのものができるはず。しかし時間が足りず、規模が大きいものは実現できなかった。マップの大きさなどを見誤っていたため予定通りに進まなかった。

ある程度作業を進めたところで話し合う時間を作り、今実装したいものと実装が厳しそうなことを洗い出す機能の取捨選択のタイミングを設けるべきだった。作ったプログラムをボツにすることが

あったからこそ先に話し合っただけで決める形を取るべきだった。

広大なマップを活かしきれていないため、もう少しコンパクトなマップにして、複数人で動くことを前提とした構成にする。

今の状態ではプレイヤーの移動以外の機能が少ない。できることが限られているため、プレイヤーアバターにのみできることを増やし、操作をしていて楽しい体験ができるようにしたい。

外部サーバーとの接続はできたものの、こちらの音声を見分ける部分などはまだ実装していないため、音声入力の仕組みを組み込む。会話をする中で感情に合わせた AI アバターの表情を変更できるようにする。

また、感情の付与としていた AI の機能周りの部分に着目し、on または off のみだったものを 5 段階くらいのパラメータに変えることでより会話の幅が広がると考える。

(※文責: 沼田晴登)

### 8.5.2 3DCG チーム

今後の展望として 3DCG チームとして実装したい機能として NPC キャラクターの表情変化とモーションの実装があげられる。現状実装できていないのはこの機能は NPC キャラクターに実装される AI がどんな感情であるか。それに対してどのような表情変化、モーションを実装すべきかという課題があり、中間時点での AI チームとの相談の結果、この機能の実装の優先順位を低く見積もったため実装に至らなかった。しかしこれは現実のコミュニケーションにおいて感情や意思を伝える上で重要な役割を持つコミュニケーション機能であり、私たちの目的である NPC とのリアルな会話への必須条件になるためこの機能の実装如何によってキャラクターとの会話のリアルさというものが格段に向上すると思われる。

また制作した各ワールドをつなぐ空間として廊下を制作したが、イメージとしてホテルの廊下をサンプルにしたようなデザインを通して各部屋ごとに扉に特徴を持たせ、その横にプレーヤーテキスト表示による部屋の紹介から各部屋への誘導を促すような効果を意識したが、各部屋の配置についても考える余地がある。現状作成したワールドの数が少ないため感じることは少ないが、制作した順に配置した方が制作を経て上達している部分のクオリティの差やが浮き彫りになりづらいのと上達過程に関して作成者毎に分類して配置することができるのでさらに分かりやすく視覚的にも見やすくなる。これはホテルの廊下のような空間配置からある程度容易にできることなので今後の拡張要素としてワールドの種類が増加が見込める場合には適用していきたい。

もう一つ見直したい点として現状達成できている点はそれぞれの空間に入る前の扉のデザイン。空間に入る前にその空間がどのようなコンセプトで制作されたのか、内部の様子がどのような物なのか、扉を見ればある程度理解できることである。ただし全体を通して、そもそもこの空間はどのような目的で制作されたのか等が、VRChat に公開されているため、その関係をまったく知らない外部の人間が見た際にも理解可能なものにならなければならない。そのために廊下に入る前にもう一つ空間を用意し、そこでメタバース DE アバターとは何なのか、コミュバースとは何なのか、この場にいる NPC 達は AI で制御されているなどを文章や図、もしくは入口に説明用のナビゲーター NPC を設置し、口頭で説明させる事ができれば好ましい。また、発表会専用の簡易的な説明用空間を制作することができていれば初めてこの空間を訪れる、またこのプロジェクトの意義について

まったく知らない第三者が見た際に分かりやすくすることができると思う。

(※文責: 白石侃穩)

### 8.5.3 AI チーム

中間時点では3つの課題が挙げられた。応答時間の改善、VRChat 内での動作の最適化、実装する AI の個性付けである。1つ目の課題である、応答時間の改善については、制作期間と時間配分を考えた上で、今回は実装を断念した。しかし、方針の検討は2つほどついている。1つは、得られた応答文を、文節や句読点などで細かく分けた上で、都度音声化するというものである。音声化を細かく段階分けすることで、音声化にかかるラグを低減させる目的である。この手法は長文で特に有効であると考えられる。しかし、応答文が比較的短文であった場合、複数回プログラムを実行することになる関係上、かえって処理が長くなる可能性については留意したい。もう1つは、応答文の生成待機中に、「えっと」や「うーん」といった発言をさせることで場をつなぐというものである。こちらは、応答生成までのラグを低減させることが可能である。しかし、根本的な問題として、現状のシステムは OpenAI 社の提供する API に頼っているため、抜本的な解決は難しいということが挙げられる。我々にできることは、OpenAI 社の提供する API をいかに使うか、そこから得られる結果をどのように処理するかといった、いわば工夫だけである。OpenAI 社により提供される API、あるいは AI そのもののアルゴリズムを改善することは、当然ながら技術的に不可能である。しかし、少なくとも当プロジェクト発足時点においては、OpenAI 社の提供する API および AI がもっとも高いパフォーマンスを発揮するとの結論に至り、これを利用する方針とした。もし今後、これ以上のクオリティを求めるのであれば、応答文生成に用いるエンジンの見直しも、併せて考えるべきである。

2つ目の課題である、VRChat 内での動作の最適化については、会話可能なアバターがただそこにいるだけ、というのが現状である。とはいえ、必要最低限の動作自体は実装できた。今後の展望としては、たとえば、AI がプレイヤーに追従する、すなわち AI を連れて歩ける、というものが考えられる。グループ A においては、複数のロケーションで、AI とその場に合った会話を楽しめるというコンセプト上、AI アバター自体は、ワールド内を移動する必要がなかった。そのため、前述のようなシンプルな実装でも問題なかった。しかし、グループ B においては、AI の成長をコンセプトにしており、その成長にはプレイヤーの行動が介入する。そのため、AI アバターはプレイヤーを追従することが望ましかったが、実装は困難であった。結果的に、実装には至らなかった。その原因は、AI アバターは、あくまで、システム上一人のプレイヤーとしてメタバース上にログインする形を取っているためである。したがって、プレイヤーの挙動そのものに、メタバース上の機能が関与することは難しく、実装は困難を極めた。本来、人間が手動で行うはずの操作を完全に自動化することが困難なためである。このような手法を取った理由は、VRChat 特有の問題として、外部との通信が困難であることが挙げられる。このため、AI の稼働に必要な通信が、自由に行えないという問題があった。これを解決するために、AI アバターはあくまで1人のプレイヤーとしてメタバース上に存在するという手法をとった。AI の稼働に必要な環境を、VRChat からの通信によってではなく、ログインしている端末からの通信で賄えるためである。当然この手法にも課題はあり、ログインするアバターの数だけ端末が必要である点、また、前述した通り、AI の挙動を制御するのが難しい点等が挙げられる。他にも、AI との会話に際して行われる人間の発話は、AI に入力するためにメタバース上で録音されており、プライバシーの問題も懸念点である。利用者には

事前に許諾を求めることが必要となるだろう。上記のように、VRChat 上で対話可能な AI を実装することには、さまざまな問題点が存在する。しかし、「メタバース上で AI と会話する」というコンセプトにおいて、メタバース分野で最も強いプラットフォームの 1 つである VRChat を切るとは考え難い。今回の制作においては、あくまで最低限会話ができるものにとどまった。しかし、今後の発展を考える上では、VRChat というプラットフォームを利用する上での諸問題を念頭に置いたうえで、それらを解決する様々なアプローチを検討する必要がある。

3 つ目の課題である、実装する AI の個性付けについては、ファインチューニングという手法で解決した。ファインチューニングとは、AI の学習データに、ユーザーの目的に合ったデータを追加することで、目的とする応答を得られやすくするというものである。それぞれのメタバースの目的に合った AI となるよう、学習データを用意し、ファインチューニングを行った AI を実装した。ファインチューニングという手法を取った理由としては、前述したように、AI から理想的な回答を得られる可能性が高くなること、また、こちらが期待する挙動を直接学習データに組み込むことで、プロンプト上での指示を減らすことができることが挙げられる。プロンプトとは、人間が AI に与える指示のことである。プロンプトは、AI との通信リクエストの度に、都度送信される。回答して欲しい問いかけのほか、このような口調で回答して欲しいといった指示も可能である。プロンプト上での指示を、効果的に行うことができれば、それぞれのメタバースに合った個性を AI に持たせることは可能である。しかし、現実的には、プロンプトには文字数制限がかけられている。そのため、大量のプロンプトを一度に送る事はできない。また、AI の傾向として、プロンプトの量が増えれば増えるほど、再現率は低くなる。さらに、プロンプトの量が増えれば、前述の通り、毎回の通信でやりとりされるデータ量が増え、結果として応答生成にかかる時間が長くなる。このように、プロンプト上での制御のみに頼ると、様々な問題が発生する。しかし、ファインチューニングを導入することで、これらの問題は一気に解決すると見込まれる。よってファインチューニングを導入を決定した。実際に導入したところ、たしかに、AI の回答の傾向が、こちらが意図したものに近くなった。しかし、徹底されているとはいえ、求める水準には遠かった。最終的には、プロンプト上での指示も併用しての実装となった。このようにすることで、ファインチューニングのみの場合や、プロンプト上での指示のみの場合と比較して、AI の回答に、意図した傾向が安定して現れた。ファインチューニングのみでの個性付けは困難であったが、ファインチューニングを用いるという試み自体は成功したと言える。また、ファインチューニングは、AI に感情をもたせるという点でも、大いに機能した。AI に感情をもたせ、その感情がアバターの表情として表れるようにするために、応答文と、その応答文に込められた感情をセットにして、JSON 形式で返すようにファインチューニングした。以下は想定される AI からの応答の一例である。

```
{emo:glad; message: 楽しかったよ！ありがとう！}
```

このように、“emo” に格納された感情と、“message” に格納された応答文のセットを、AI が返すようファインチューニングを行った。上記の文がそのまま応答文として出力されるようになっている。この JSON 形式の応答文から、“emo” の値を抜き出し、表情変化に用いた。また、“message” の値を抜き出し、音声化した。このようにすることで、文字や音声のやり取りだけでなく、表情という情報も加わった、AI とのコミュニケーションが可能となった。仮に、上記のように回答するよう、プロンプト上で指示しようとしても、求める結果を得ることは困難である。実際、プロンプトだけで AI に個性と感情を両立させる試みは難航した。しかし、ファインチューニングを用いることで、簡単に実装することができた。総じて、ファインチューニングを利用するという試みは上手くいったと言える。今後の展望としては、学習データの充実が挙げられる。今回の制作では、ファインチューニングのデータ量が不十分であったと言える。具体的には、AI アバター 1 体につ



き、学習データは 50 程度しか用意できていない。学習データの用意は全て人力であり、AI に自動生成させる方法も試したが、結果は良好とは言えなかった。このような事情から、AI アバター 1 体あたりの学習データは非常に少ないものとなった。しかし、学習データを増やすことができれば、ファインチューニングしたモデルだけで、プロンプトでの指示なしに、意図した応答が得られる可能性は十分に考えられる。そうすれば、応答の高速化も見込め、全体的な質の向上も望めるであろう。しかし、繰り返しになるが、応答文生成に OpenAI 社の API を用いている以上、技術的に介入できる範囲は限られる。さらなる質の向上を求めるならば、用いるエンジンの見直しを考える必要が出てくるだろう。

ここまで述べてきたように、本プロジェクトで用いた AI は、外部のソフトウェアに頼る部分が多い。応答文生成は OpenAI 社の提供する API に、音声化は VOICEVOX に、メタバースのプラットフォームは VRChat に依存している。VOICEVOX に関しては、あくまで利用する上で簡便だったから、ということが採用理由に過ぎず、より適した選択肢は存在すると考えられる。例えば、CoeFont や COEIROINK など、より肉声に近い質感の合成音声ソフトは無数にある。今回は、制作期間の関係上、VOICEVOX での実装となったが、今後の発展を見据えて、より適した選択肢を探っていきたい。応答文生成に、OpenAI 社の提供する API を引き続き使用するかは吟味が必要である。本プロジェクトでは、まず AI の中心に何を据えるか検討するところから始まった。当初は RASA を使用する案もあったが、自然な対話と呼ぶには、およそ期待できない精度であったため却下された。以下に RASA の特徴と問題点を説明する。RASA とは、ドイツの RASA 社が開発する、オープンソースのチャットボット構築フレームワークである。言語は、python で記述される。拡張性が高く、また、軽量であることが特徴である。与えられた文章に対し、自然言語処理を用いることで、関心対象などを分析し、適切な応答文を返すというものである。この際、ユーザーから与えられた情報に不足などがあった場合、それを補足するよう、ユーザーに返すといった挙動も可能である。実際に使ってみると、大きく 3 つの問題点が見えてきた。まず 1 つ目として、RASA には、想定される問答とそれに対する回答を、開発者が逐一用意しなければならないという点である。RASA は、あくまでインプットされたやり取りしかできず、想定外の会話には弱い。例えば、実装した AI アバターとプレイヤーとのやり取りとして、自己紹介を求められる、というシーンは容易に想定される。こういったシーン・問いかけに対する回答を予め用意しておけば、違和感なく会話は成立する。しかし、開発者の想定しないやり取りがあった場合、例えば、ユーザーがいきなり「アラン・チューリングについて教えて」と問いかけてきても、RASA の場合、会話が成立しなくなる。開発者は、そのような会話は想定しておらず、これに対する応答は当然用意されていない。その場合、何も返さない、あるいは回避策として「それについては知らない」といった回答を用意していた場合、それを返す。ここで問題なのは、“対話”において、話題を予め想定しそれらに対する回答を一つ一つ用意し網羅するのは、現実的に不可能であるという点である。対話において、想定される話題はあまりにも多く、そもそも想定するという表現自体が不適切であるとすら言える。そしてこれは、2 つ目の問題点に繋がってくる。2 つ目の問題点は、RASA が自然言語処理を用いて解析するのは、あくまでユーザーからのメッセージのみであるという点である。RASA 自身の回答は、前述の通り、あくまで開発者によって用意されたもののみで、それ以外のやり取りに遭遇した場合、自然言語処理によって回答が自動生成されるということはない。さらに、想定されたやり取りにおいても、RASA は、開発者によって入力された回答を一字一句同じに出力するのみである。具体的には、何度自己紹介を求めても、返されるのは全く同じ回答のみである、これでは自然な対話とはとても呼べないであろう。3 つ目の問題点は、RASA の閾値設定が非常に難しいということが挙げられる。具体例を挙げて解説する。例えば、ユーザーからのメッ

セージを解析し、その興味対象が天気であるとわかった場合、「いい天気ですね」と返すようプログラミングしたと仮定する。閾値とは、上記の「興味対象が天気であるとわかった場合」という部分を、どれだけ厳密に判断するかというパラメータである。閾値が高すぎると、どんな聞き方をしても、例えば「今日はいいい天気ですね」と、明瞭なメッセージを入力しても、「いい天気ですね」とは返してはくれない。逆に、閾値が低すぎると、どれだけ意味のない文字列であろうと、例えば「ああああ」などと入力しても、「いい天気ですね」と返すようになる。これを踏まえた上で、実際に使ってみると、デフォルトの閾値ではあまりにも低く、意味のないメッセージでも、さも理解しているかのように振る舞った。閾値を挙げてもこれは改善されず、およそ対話とは呼べるものではなかった。ここまで挙げてきた、RASA の問題点をまとめると、想定されたやりとりしかできず、想定されたやり取りであっても一言一句おなじ回答を繰り返すのみで、そもそも興味対象の判定があまりにもずさんである、ということである。総じて、RASA の役割は、例えば再配達依頼などの、一定のフローが想定される業務用のやり取り程度が最適であり、対話には到底向かないという結論に至った。そのため、プロジェクト発足当初は、十分な精度を有するほぼ唯一の AI として、ChatGPT に白羽の矢が立ち、これを採用する運びとなった。しかし現在は、プロジェクト発足当初と比べると、多数の選択肢がある。また、OpenAI 社の提供する API も、今後バリエーションや機能が増えていくことが予想される。本プロジェクトでは GPT-3.5Turbo を利用した。これは、自然な対話と呼ぶに十分な精度を有し、かつ比較的高速であると認められたためである。しかし今後は、より高速・高精度なモデルの登場が予想されるほか、GPT-4 のファインチューニング対応も見込まれる。あるいは、究極的には、大規模言語モデルを独自に開発・利用できることが望ましい。また、前述した通り、RASA は自然な対話には向かないが、逆に言うと、ある程度の規則性のあるやり取りでの信頼性はあるといえる。この点を活かし、メタバース内で発生するイベントでのやり取りとして、ピンポイントでの起用はできそうである。例えば以下のようなものである。

(喫茶店というテーマのロケーションで)

AI アバター「何を注文する？ コーヒーかカフェラテがおすすめだよ」

プレイヤー「コーヒーにしようかな」

(RASA で興味対象が「コーヒー」であると判定する)

(興味対象が「コーヒー」であるときの応答を出力する)

AI アバター「いいね。すみませーん。コーヒー2つお願いします」

このように、プレイヤーからの応答を極度に絞った環境下で、ある地点に到達した際に一度だけ発生するイベントとして、実装することはできそうである。懸念点として、これはゲーム的なアプローチであるといえ、自然な対話と呼べるのかということには留意したい。こうした可能性を踏まえ、今後も OpenAI 社の提供する API を使用するのか、使用するのならばモデルは何を採用するのか、あるいは別の技術を使用するのか、独自開発するのか、つねに AI に処理を任せるのか等、さまざまな方策を検討し、AI との自然な対話という目標を、より高度なものとしていきたい。また、プラットフォームに関しても、順次検討することが望まれる。VRChat 内に AI アバターを実装するというのは簡単なことではなく、拡張性という点でも疑問が残るものとなった。しかし、VRChat はメタバースプラットフォームとしてもっともメジャーであるといえる。利用者の数は抜きん出ており、今後の発展や正式にリリースする可能性などを視野に入れた際、もっとも強力な選択肢であることは間違いない。さらに、あくまで AI アバターの実装という点で相性が悪いのみであり、ワールドの構築やアバターのモデル利用等、メタバースの構築において必要になるリソー

## METAVVERSE DE AVATAR

スや情報など、バックアップの豊富さも魅力である。AI アバターの実装は困難な点が多くあったが、それ以外は優秀であり、決して開発が困難であるということを指摘しているわけではない。さらに、AI アバターにとっても強力なツールが存在するのもまた事実である。リップシンクや豊富なアバターの挙動プリセットなど、AI アバターにより人間味を持たせるための諸機能が存在する。これらは、メタバースプラットフォームとして VRChat を利用することを決定した強力な要因である。しかし、メタバースプラットフォームは他にも多数存在する。次点で利用者が多く、間口の広さが特徴的な cluster や、配信・広報に特化したバーチャルキャストなど、検討すべきメタバースプラットフォームはいくつも存在する。また、メタバースとは、VR 空間に限ったものを規定する言葉ではない。Unity 内に一大空間を構築することも、十分に検討に値する。総じて、メタバースという多人数が交流できる空間を構築すること、アバターを用いて AI と人間の視覚的な垣根を撤廃すること、AI がより人間的で自然な振る舞いができること、これらの、本プロジェクトが示す原点に立ち返って、何が最適な選択肢であるか、多角的に検討することが求められる。

(※文責: 大原拓己)

## 付録 A 中間発表で使⽤した発表スライド

スライド全体は以下の URL を参照.

[https://docs.google.com/presentation/d/1PdKemj\\_zM7dG5ADW6eJrUxWEJKTsK3nk/edit?usp=sharing&ouid=107562026529821251431&rtpof=true&sd=true](https://docs.google.com/presentation/d/1PdKemj_zM7dG5ADW6eJrUxWEJKTsK3nk/edit?usp=sharing&ouid=107562026529821251431&rtpof=true&sd=true)

# 付録 B 中間発表に向けて制作したポスター

中間発表で使用したメインポスターを掲載する。各グループのポスターは以下の URL を参照。  
[https://docs.google.com/presentation/d/1PdKemj\\_zM7dG5ADW6eJrUxWEJKTsK3nk/edit?usp=sharing&oid=107562026529821251431&rtpof=true&sd=true](https://docs.google.com/presentation/d/1PdKemj_zM7dG5ADW6eJrUxWEJKTsK3nk/edit?usp=sharing&oid=107562026529821251431&rtpof=true&sd=true)



図 B.1 中間発表で使用したメインポスター