

デジタルキャラクター技術に対する関心

安田陽真

目次

1	関心の理由・きっかけ	2
2	music2dmx ～照明効果の自動選択～	4
2.1	概要	4
2.2	開発の動機・考え	4
2.3	仕様・技術	4
2.4	成果	7
3	VRMeet ～個人 VTuber 向け 3D コラボソフトウェア～	8
3.1	概要	8
3.2	開発の動機・考え	8
3.3	仕様・技術	8
3.4	成果	9
4	MRTalk ～AI キャラクターと現実世界で会話～	10
4.1	概要	10
4.2	開発の動機・考え	10
4.3	仕様・技術	10
4.3.1	初期バージョン(Unity)	10
4.3.2	Web 版 MRTalk	11
4.3.3	Web 版 MRTalk(未踏ジュニア応募版)	12
4.4	成果	13
5	aikyo ～AI キャラクター同士の会話基盤～	15
5.1	概要	15
5.2	開発の動機・考え	15
5.3	仕様・技術	16
5.4	成果	17
6	HoloTrip ～AI キャラクターを活用した地域振興ソフトウェア～	18
6.1	概要	18
6.2	開発の動機・考え	18
6.3	仕様・技術	18

6.4	成果	20
7	kokoro/rig ～エンジニア向け 2D メッシュ変形ライブラリ～	22
7.1	概要	22
7.2	開発の動機・考え	22
7.3	仕様・技術	22
7.4	成果	23

1 関心の理由・きっかけ

私は、小学生のころから様々なキャラクターコンテンツに接してきました(図 1)。私がキャラクターコンテンツに接し始めた 2018～19 年ごろは、以前から存在したアニメや漫画をメインに展開していくキャラクターコンテンツのあり方から、バーチャル YouTuber(VTuber)を始めとした、デジタル技術を中心としてコンテンツを展開していくキャラクターコンテンツのあり方への転換の時期でした。その中で私は自然とデジタルキャラクターを具現化させるための 3D 技術や、機械学習モデルなどの技術に興味を持つようになりました。中でも、LLM の台頭が私の興味を加速させました。LLM 登場以前にも、ルールベースや簡易な NLP などキャラクターとの対話実装を試みてはいましたが、クオリティには限界がありました。LLM を用いることで、こちらが想定しきれない返答ケースにも対応できるようになり、キャラクターとの会話体験が大幅に向上したことが、興味を加速した理由です。

本章では、こうした関心に沿った取り組みを時系列で示します。



図 1: キャラクターコンテンツに親しんでいる様子

2 music2dmx ～照明効果の自動選択～

開発時期: 高校 1 年(2024 年 9～11 月)

2.1 概要

music2dmx は、入力音楽の盛り上りを分析して、それに対応した照明効果を適用するプロジェクトです。バーチャルライブにおける照明選択の複雑さを自動化するために開発しました。

2.2 開発の動機・考え

私は 3D 技術を活用したバーチャルライブに興味があり、その中で Unreal Engine の DMX 機能を知りました。DMX とは、現実世界のライブなどで用いられる照明機材を操作するためのプロトコルです。Unreal Engine では実際の DMX 信号を Art-Net というプロトコルを用いることで Ethernet 越しに受け取り、バーチャルな照明機材を動かすことができます。そこで私は音楽に合わせて照明機材を動かそうとしましたが、Unreal Engine 側の打ち込み機能で照明機材を操作することはとても労力がかかります。ならば、音楽を分析して自動で照明効果を選択し、Art-Net パケットを Unreal Engine に送ってしまおう！と考え、実装しました。

2.3 仕様・技術

music2dmx は 図 2 に示す技術構成で動作しています。

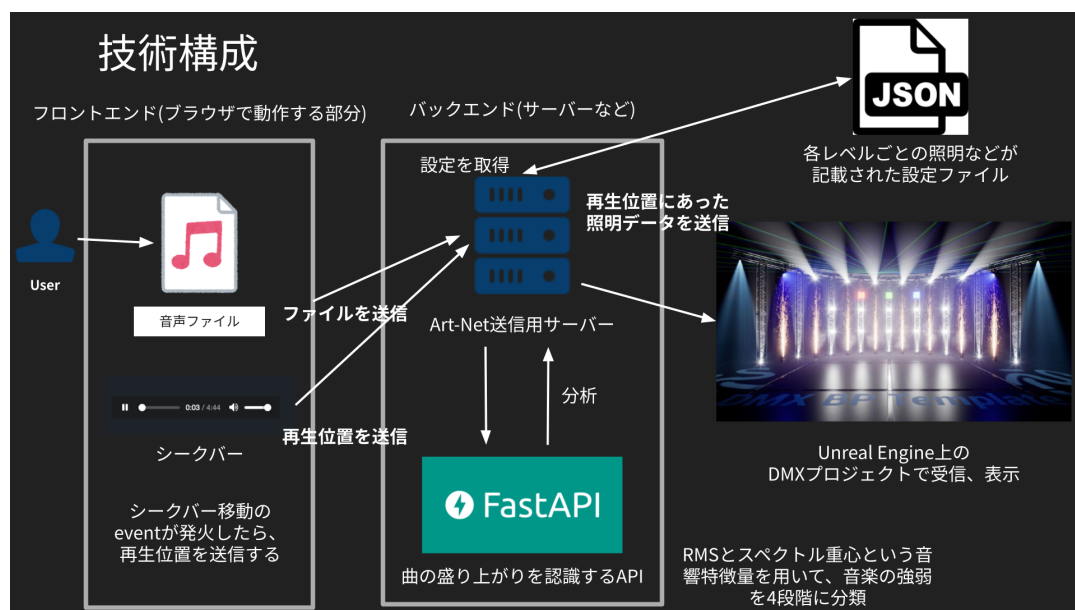


図 2: music2dmx の技術構成

まず、本ツールでは照明効果を以下の4つに区分します。

```
1 export type level = "low" | "mid" | "high" | "big_chorus";
```

TypeScript

config ファイルで、それぞれの level に対応した照明効果を定義します。以下は config ファイルの一例です。

```
1  {
2    "fixtures": {
3      "spotlight": {
4        "universe": 1,
5        "baseChannels": [1, 12, 23, 34, 45, 56, 67, 78],
6        "channels": {
7          "color": 1,
8          "dimmer": 2,
9          "shutter": 3,
10         "pan": 7,
11         "tilt": 9,
12         "zoom": 11
13       }
14     },
15     ...省略
16   },
17   "levels": {
18     "low": {
19       "enableFixtures": ["spotlight", "staticPatch", "strobePatch"],
20       "fixtureSettings": {
21         "spotlight": {
22           "dimmer": "bpmSyncSlow",
23           "shutter": "bpmSyncSlow",
24           "pan": 128,
25           "tilt": 150,
26           "zoom": 255
27         },
28         "staticPatch": {
29           "r": 255,
30           "g": 255,
31           "b": 255,
32           "dimmer": 255,
33           "shutter": 255
34         },
35         "strobePatch": {
36           "dimmer": 200,
37           "shutter": 255
38         }
39       }
40     },
```

JSON

```

41    ...省略
42    "big_chorus": {
43        "enableFixtures": [
44            "spotlight",
45            "LEDWash",
46            "staticPatch",
47            "strobePatch",
48            "pyro",
49            "laser",
50            "fireworks"
51        ],
52        "fixtureSettings": {
53            "spotlight": {
54                "dimmer": "bpmSyncSlow",
55                "shutter": "bpmSyncSlow",
56                "pan": 128,
57                "tilt": 150,
58                "zoom": 255
59            },
60            ...省略
61        }
62    }
63 },
64 ...省略

```

music2dmx には管理用の WebUI が存在しており、そこに音楽の音声ファイルを入力します。この管理画面では、音声ファイルの再生位置と一時停止を制御することができます。管理画面で入力された音声ファイルは、Python で記述された音楽分析サーバーに送信されます。音楽分析サーバーは以下の手順で、音楽の盛り上りを分析します。

1. 音声ファイルを短いウィンドウに区切ります。
 2. ウィンドウごとの RMS、スペクトル重心を算出します。
 3. RMS とスペクトル重心を足し合わせた数値をしきい値と比較し、low, mid, high に分類します。
 4. 曲の後半 60% 以降の区間において合算した数値が最大となる区間を大サビ区間とします。
- この実装は librosa を用いており、この¹記事で示されている手法を使用しています。結果は JSON として返却され、その JSON が Art-Net 制御サーバーに渡ります。Art-Net 制御サーバーは、クライアントから受け取った音声ファイルの現在の再生位置と、音楽分析サーバーが返却した区間データを比較し、現在の盛り上がりラベルを取得します。config ファイルで事前定義

¹<https://www.wizard-notes.com/entry/music-analysis/highlight-detection-by-rms>

されているラベルごとの照明効果を Art-Net パケットに載せて、Unreal Engine に送信することで照明機材が動作します。

2.4 成果

本ツールは、所属している高校のイベントで展示し、観客の皆様から高評価をいただきました(図 3)。また、本ツールの開発を通して、Art-Net という比較的マイナーなプロトコルについて学習することができ、とても楽しい経験となりました。

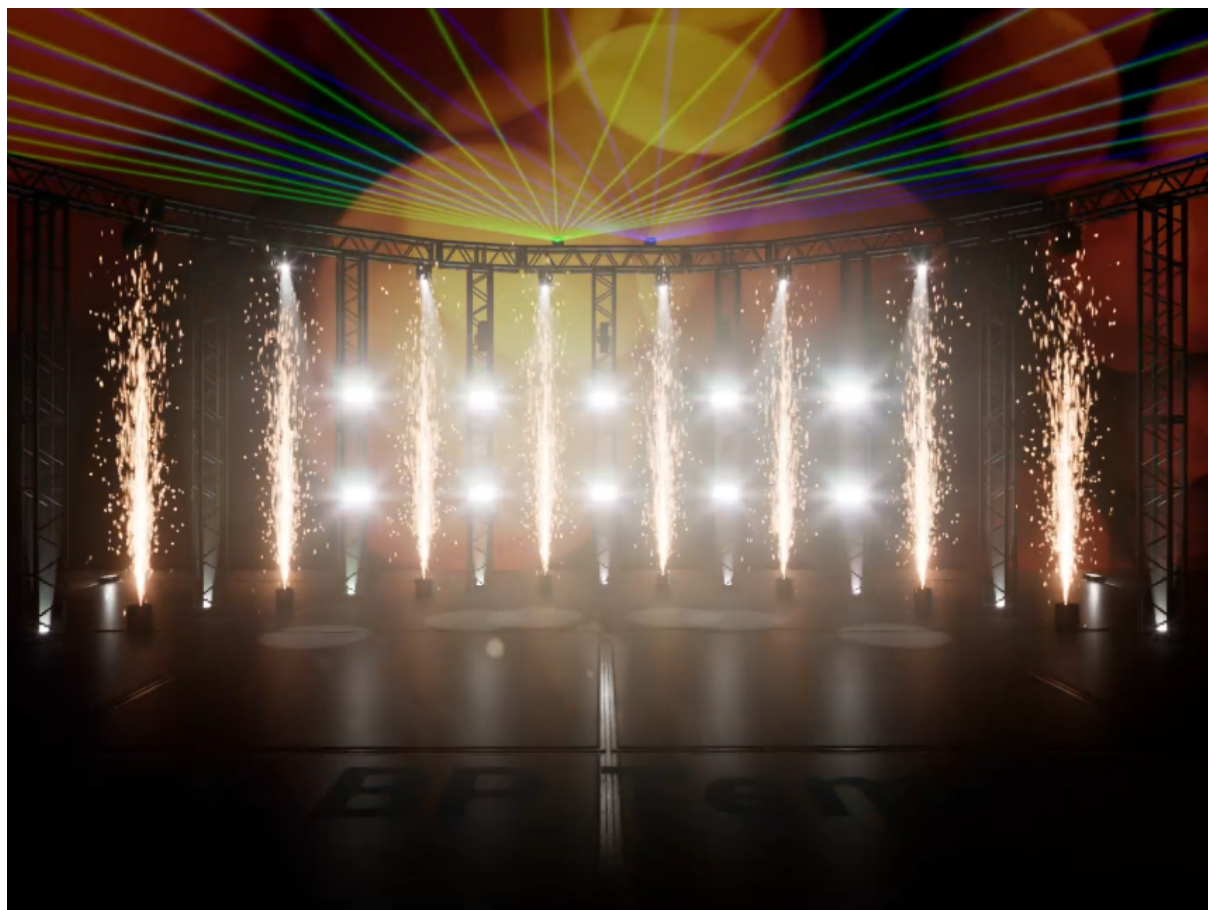


図 3: music2dmx を用いて照明機材が動作している様子

3 VRMeet ～個人 VTuber 向け 3D コラボソフトウェア～

開発時期: 高校 1 年(2024 年 10 月～2025 年 1 月)

3.1 概要

VRMeet は、個人のバーチャル YouTuber(VTuber)向けの、3D コラボソフトウェアです(図 4)。高価なデバイスや複雑なセットアップ無しで、遠隔 3D コラボが可能です。このアプリケーションで第 4 回 Joyo High School テックコンテスト「テック特別賞」を受賞しました。

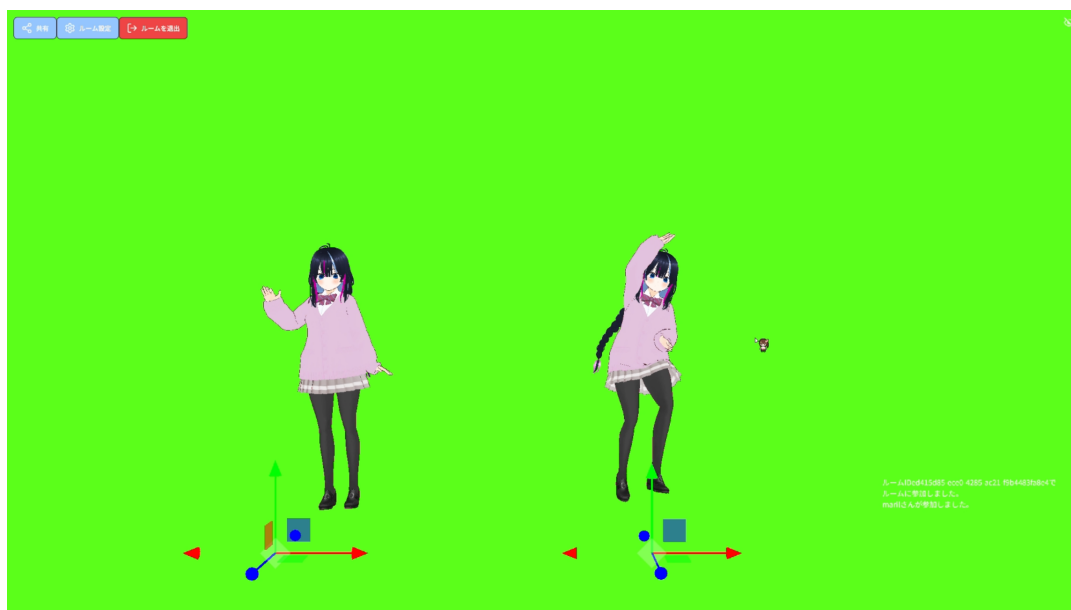


図 4: VRMeet が動作している様子

3.2 開発の動機・考え

3D コラボとは、画面上にリアルタイムで動作する VTuber の姿を並べ、それを配信する行為を指します。既存の 3D コラボレーション手段は、複雑なソフトウェアの設定や高価な機材が必要であったため、個人 VTuber には敷居の高いものでした。そこで、VRMeet では MediaPipe と WebRTC を用いてブラウザのみで完結する 3D コラボを実装しました。

3.3 仕様・技術

VRMeet は、図 5 で示す技術構成で動作しています。

システム構成

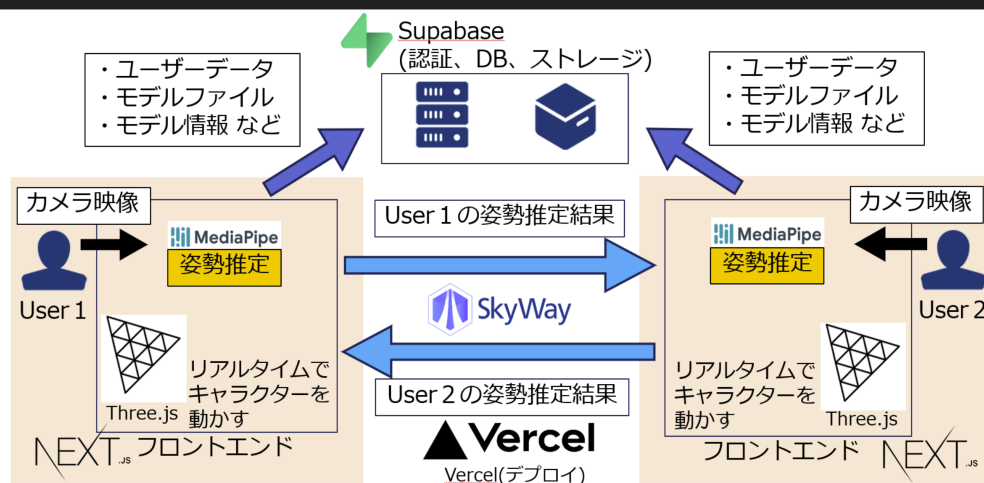


図 5: VRMeet の技術構成

ローカルでの姿勢推定には MediaPipe を、キャラクターの描画には Three.js を用いています。MediaPipe の姿勢推定結果を、SkyWay という STUN や TURN サーバーを提供してくれる WebRTC プラットフォームを用いて同期しています。また、モデルファイルやユーザーデータは Supabase で管理しています。

3.4 成果

Web カメラと Web ブラウザのみで姿勢推定結果を同期することができるシステムを開発しました。また、開発当時は WebRTC についてほぼ触れたことがなかったため WebRTC のシグナリングや TURN など、P2P 特有の概念を理解する助けになりました。さらに、本アプリケーションを第 4 回 Joyo High School テックコンテストで発表し、「テック特別賞」を受賞しました。

4 MRTalk ～AI キャラクターと現実世界で会話～

開発時期: 高校 1～2 年(2024 年 4 月～2025 年 7 月)

4.1 概要

MRTalk は、AI キャラクターと現実世界で会話することができる WebXR アプリケーションです。Meta Quest3 と WebXR を使って 3D モデルを描画、裏側では LLM が動作し、ユーザーの音声入力に対して返答をします。本アプリケーションは 1 年をかけてバージョンアップを繰り返し、その後未踏ジュニアに採択されました。

4.2 開発の動機・考え

MRTalk 開発のきっかけは、Meta Quest3 を購入したことでした。これまで妄想していた AI キャラクターを現実的に召喚して会話する体験を、Meta Quest3 の MR 機能と LLM を組み合わせることで実装できるのではないかと考え実装を開始しました。

4.3 仕様・技術

MRTalk は、1 年半を通して開発され、3 回のバージョンアップをしました。

4.3.1 初期バージョン(Unity)

MRTalk の初期バージョンは Unity で開発されました。キャラクターの描画部分、MR 実装には Meta Quest の Unity 向け SDK を使用しました。本バージョンでは、基本的な音声対話に加え、Meta Quest の深度センサーを用いた部屋の形状検出と Unity のナビゲーションエージェントを組み合わせ、キャラクターが家具を避けながら部屋を動き回る機能が実装されています(図 6)。LLM には GPT-4o を使用し、音声返答の生成には VOICEVOX の API を使用しました。本アプリケーションは、2024 年のゲームメーカーズスクランブルというイベントに出展し、参加者からキャラクターの見た目・性格をカスタマイズしたい等のフィードバックをいただきました。

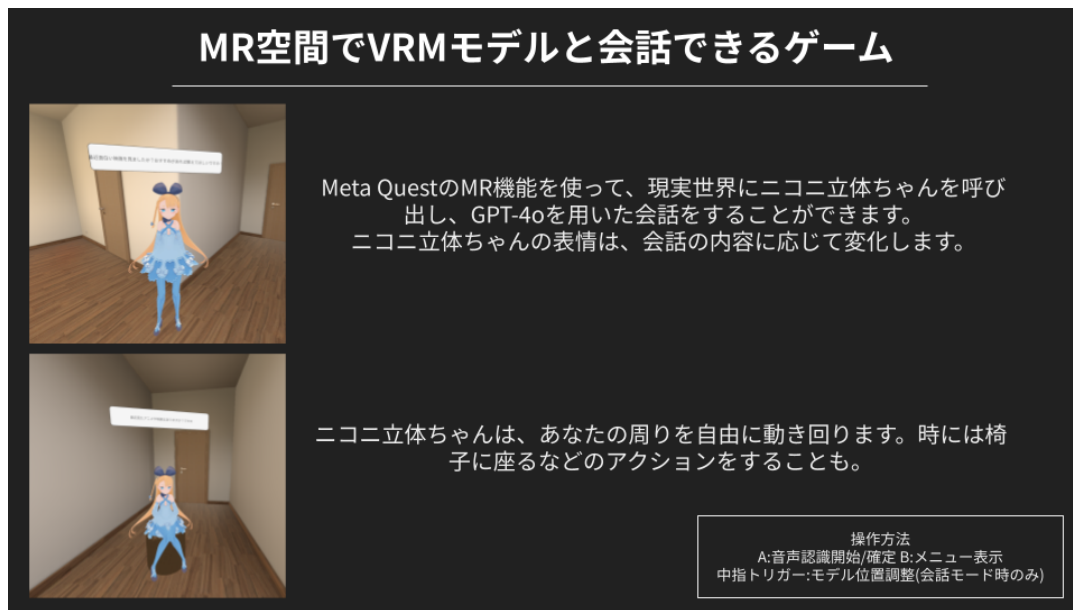


図 6: MRTalk(初期版)

4.3.2 Web 版 MRTalk

ゲームメーカーズスクランブルでのフィードバックを受け、MRTalk は Web ベースの技術スタックに完全移行しました。Web ベースに移行することで、Unity ではやや実装が複雑であった認証・アカウント管理周りの機能に関して Web の豊富なエコシステムを活用し、キャラクターカスタマイズが可能な Web サービスとして展開する狙いがありました。結果的に Web への移行は成功し、ユーザーが自由にキャラクターの性格・見た目を登録、他のユーザーと共有し、様々なキャラクターと会話することができるプラットフォームとしての MRTalk が実現しました。

技術的には、これまで Unity の Meta Quest SDK を介して取得していた部屋の形状検出結果などを WebXR Device API というブラウザ API を用いて取得し、その上に Supabase を中心としたプラットフォーム機能を実装しています。ユーザー体験としても、アプリケーションのインストールが不要となり、Meta Quest の標準ブラウザですぐに利用できるようになりました(図 7, 図 8)。

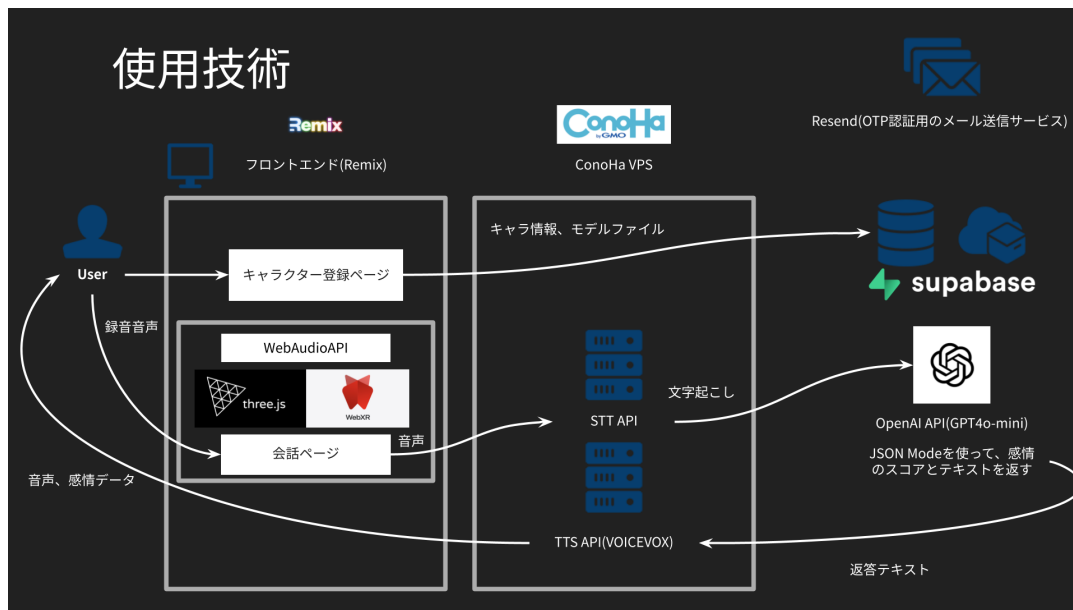


図 7: Web 版 MRTalk の技術構成



図 8: 実際にキャラクターと対話している様子

4.3.3 Web 版 MRTalk(未踏ジュニア応募版)

MRTalk を未踏ジュニアに応募するため、改善を進めました。まず、部屋の形状検出について従来では平面検出 API を用いていましたが、改善版ではメッシュ検出 API という部屋のより詳細な形状を検出できる API を使用しています。また、その API から取得できる家具ラベル等のデータを用いて、キャラクターが現実の家具にインタラクションする機能を実装しました。

さらに、キャラクターが現実のオブジェクトの裏に回ったときにキャラクターが隠れて見えるオクルージョンも実装しました。また、従来では返答ボイスの生成に VOICEVOX を用い

ていましたが、より感情的な音声を生成するため、Style-Bert-VITS2を採用しました。プラットフォーム面においても、従来ではバックエンドをSupabaseに依存していましたが、よりモジュールな構成にするため Prisma ORM + Neon DB の構成に移行しました。また、未踏ジュニアの応募時には、同高校の友人を誘い、二人体制で開発・応募しました。

4.4 成果

MRTalk の開発を通して、これまで妄想でしかなかったバーチャルキャラクターとの会話を、現実のものとすることができました(図 9)。



図 9: バーチャルキャラクターが現実の椅子に座っている様子

未踏ジュニア採択後も、リアルタイムアニメーション生成モデル²と組み合わせたプロトタイプを開発し、大阪・関西万博の「デジタル学園祭」で展示を行うなど、改良と発表を進めました(図 10)。その後、MRTalk は aikyo というフレームワークの誕生につながります。

²<https://github.com/zkf1997/DART>



図 10: 大阪・関西万博での発表の様子(左はチームメンバー)

5 aikyo ～AI キャラクター同士の会話基盤～

開発時期: 高校 2 年(2025 年 8～12 月)

5.1 概要

aikyo は、相互につながる AI キャラクターを作成するバックエンドフレームワークです。独立したキャラクターが相互につながり、自然な会話のキャッチボールができるネットワークを簡単に構築することができます。さらに、Action や Knowledge と呼ばれるシステムにより、LLM による自律的なキャラクターの操作を可能にします。本フレームワークの開発で、未踏ジュニアスーパークリエイター認定を受けました。

5.2 開発の動機・考え

前述した MRTalk の開発を未踏ジュニアで継続する中で、MR キャラクターの構築という単一の体験を提供することだけでなく、だれもが自分好みの AI キャラクターを開発できる基盤の制作に興味を持ち、プロジェクトの方向性を転換、aikyo を開発しました。既存の AI エージェントは、特定のタスクをこなす補助的な役割として設計されていますが、ただ人間らしく会話する、人間のようにふるまうだけの自律的エージェントが存在してもいいのではないか、という考えが aikyo の根幹です。

aikyo は、エンジニアが自律的な AI キャラクターを構築する際の共通基盤となることを目指しています。これまで AI キャラクターの実装手法は開発者ごとにバラバラで、確立されたアプローチが存在しませんでした。aikyo はその知見を一つのフレームワークとして整理することで、エンジニアがすでにある技術を再び開発することなく、キャラクター設計そのものに集中できる環境を提供します(図 11)。

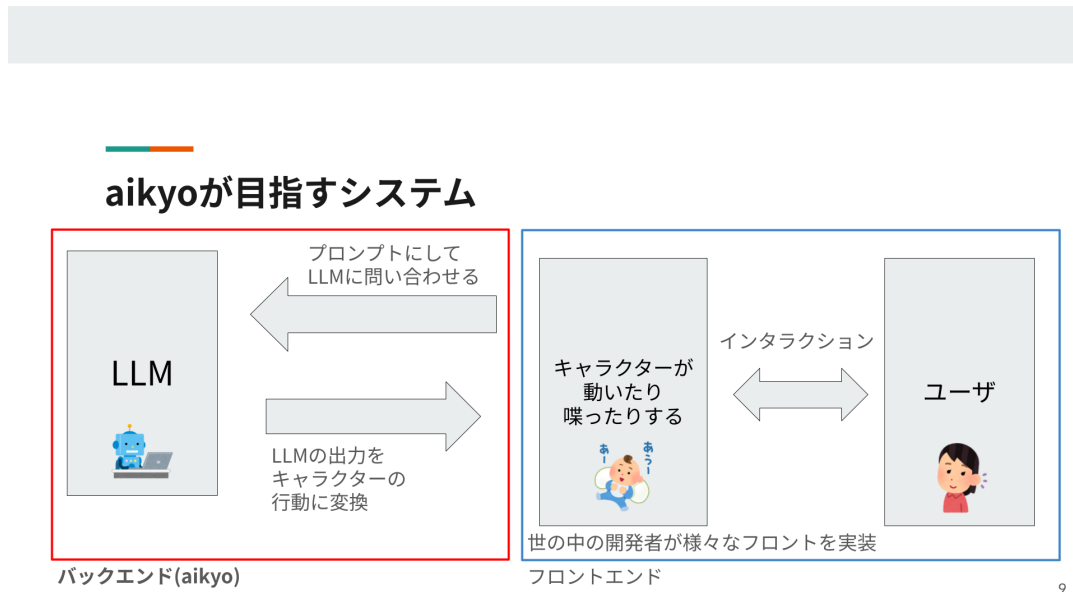


図 11: aikyo が目指すシステム

5.3 仕様・技術

aikyo では、libp2p を用いたトランスポート非依存の Gossipsub によって、AI エージェント同士の自律的なコミュニケーションを実現しています。特に注力したのは、会話分析における「隣接ペア」の概念をメッセージスキーマに取り入れた点です。隣接ペアとは質問 -> 応答、挨拶 -> 挨拶のように発話が対になって成立する構造のことで、これをスキーマレベルで定義することで、より自然なターンテイキングを実現しています。このアイデア自体は先行研究「Who speaks next?»³に基づくものですが、先行研究では単一プロセス内の単純なループとして実装されていたものを、libp2p の Gossipsub を介したネットワーク越しの通信に発展させることで、複数のエージェントが物理的に異なるマシン上で自律的に動作できる分散構成を実現しました。この取り組みは先行研究の著者である宇都宮大学の森先生からも意義深いとのフィードバックをいただいています。また、「Opening Up Closings」⁴に基づき、会話の終了プロセスを none・pre-closing・closing・terminal の 4 状態で表現する実装も行いました。これにより、エージェントが自然なタイミングで会話を締めくくれるようになっています。加えて、CEL という式評価言語を組み込み、エージェントの挙動をプログラマティックに制御できる設計としています。

³<https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1582287/full>

⁴<https://web.stanford.edu/~eckert/Courses/l1562018/Readings/SchegloffSacks1973.pdf>

5.4 成果

結果として、異なるキャラクター設定を持つ AI エージェント同士が人間の介在なしに自律会話するデモを実現しました。実際に触れてもらった知人からは「自分好みの AI キャラクターを手軽に作れる」という反応を得ました。また、本フレームワークの開発で、未踏ジュニアスーパークリエイター認定を受けました。

6 HoloTrip ～AI キャラクターを活用した地域振興ソフトウェア～

開発時期: 高校 2 年(2026 年 1 月)

6.1 概要

HoloTrip は、AI キャラクターを用いて地域振興を行うソフトウェア群です。自治体等が観光地に設置することを想定した HoloTrip Business と、観光客がスマホ等に導入して一緒に旅することを想定した HoloTrip Personal があります。HoloTrip Personal では、観光客それぞれの AI 同士を会話させることもできます。このアプリケーションで第 5 回 Joyo High School テックコンテスト「テック特別賞」を受賞しました。

6.2 開発の動機・考え

MRTalk、aikyo を通して獲得した AI キャラクター会話の知見を、何らかの分野に応用できないかと考え、観光を思いつきました。私はつくば市出身で、現在つくば市に住んでいます。隣の牛久市が「ラブライブ！スーパースター!!」というアニメ作品とコラボしたことをきっかけに、キャラクターを応用した観光アピールのアイデアを思いつきました。

6.3 仕様・技術

HoloTrip Business は、図 12 に示す技術構成で動作しています。バックエンドで LLM 呼び出しと知識・記憶の管理を行うシンプルな構成です。

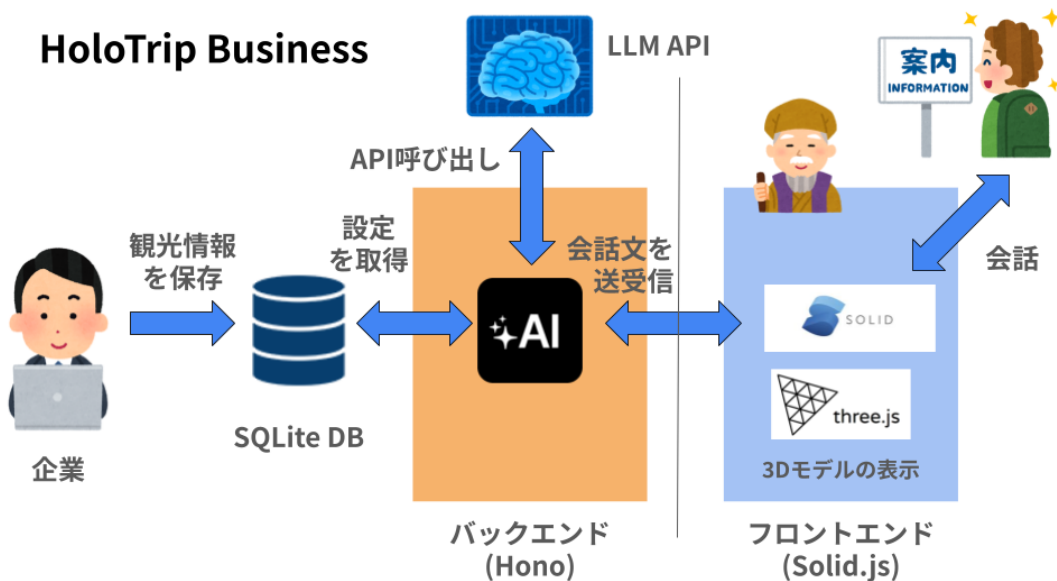


図 12: HoloTrip Business の技術構成

HoloTrip Personal は、図 13 に示す技術構成で動作しています。ユーザー管理と記憶管理、LLM 呼び出しはバックエンドで行いつつ、ユーザー設定のキャラクター情報はブラウザ側(OPFS)に保存しています。自治体等は、バックエンドの DB を操作することでキャラクターに観光情報等をインプットすることができます。

さらに、HoloTrip Personal では、同じ場所に訪れた観光客の AI キャラクター同士が会話する機能が実装されています(図 14)。キャラクター同士の会話実装では、トランスポートを抽象化し、都度生成される一時鍵を用いた簡易的なハンドシェイクを行うことでなりすましを防止しています。

HoloTrip Personal

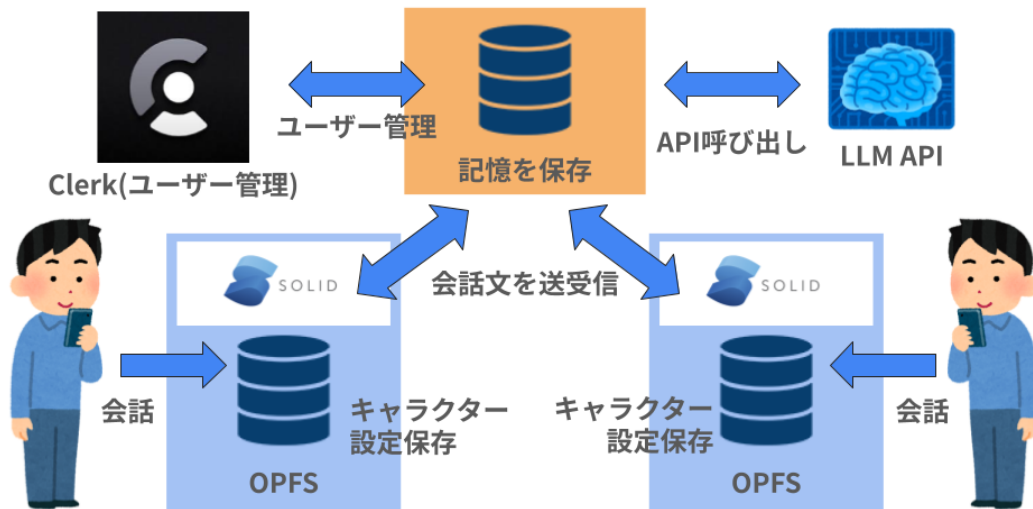


図 13: HoloTrip Personal の技術構成

②AI会話プロトコル

AI同士の通信ルールを策定

- 特定の通信手段に固定せず、様々な通信手段を利用可能に
- 電子署名アルゴリズムECDSAを用いることで、なりすましを防止
- このプロトコル部分をオープンソースとして公開

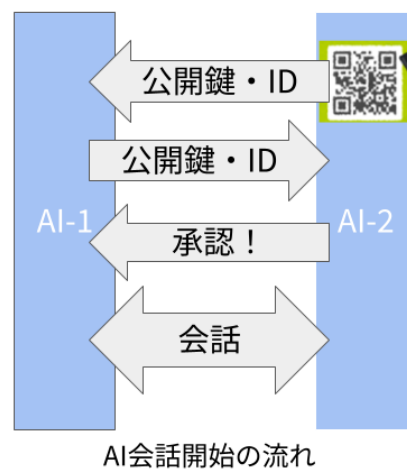


図 14: HoloTrip Personal の AI 会話プロトコル

6.4 成果

本アプリケーションの開発を通して、これまで取り組んできた AI キャラクターを他分野に応用する可能性について探ることができました。また、本アプリケーションを第 5 回 Joyo High School テックコンテストで発表し、「テック特別賞」を受賞しました(図 15, 図 16)。

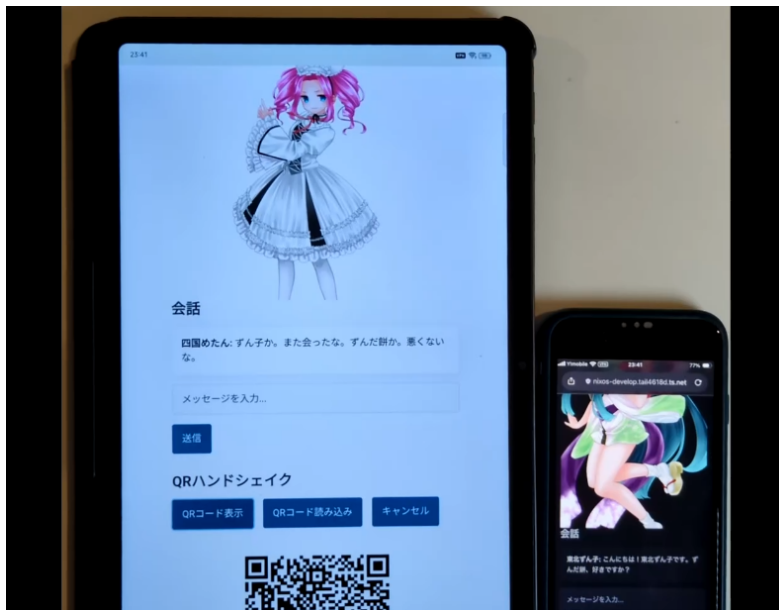


図 15: HoloTrip Personal で 2 キャラクターが会話している様子

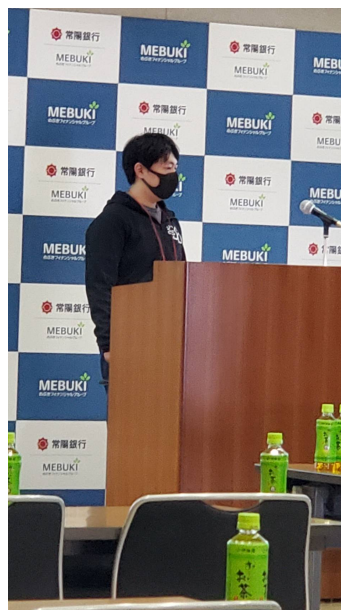


図 16: 第 5 回 Joyo High School テックコンテストでの発表の様子

7 kokoro/rig ～エンジニア向け 2D メッシュ変形ライブラリ～

開発時期: 高校 3 年(2026 年 4 月～)

7.1 概要

現在は、サイボウズ・ラボユースにて「kokoro/rig」という 2D キャラクターのメッシュ変形ライブラリを開発しています。Live2D のような GUI ベースのツールとは異なり、 $(u, v) \Rightarrow$ 変形量のような関数でキャラクターの変形を記述できる設計です。Web フロントエンドエンジニアが TypeScript の資産を活かして直感的に 2D キャラクターを動かせることを目指しています。

7.2 開発の動機・考え

Live2D のような GUI ベースのツールは、Free-Form Deformation のような技術を用いて、ユーザーがキャラクターのメッシュ頂点を明示的に編集することで、変形を表現していました。しかし、この作業には変形量を手で動かすコツを掴む必要があり、専用のスキルが必要になります。kokoro/rig は、この GUI による手作業の編集ではなく、エンジニアにとってより直感的な関数型モデルを使用します。

7.3 仕様・技術

このライブラリにおいて、キャラクターを動かすために必要な概念は一つだけで、 $(u, v) \Rightarrow$ 変形という関数のみです。この関数をポーズ関数と呼びます。引数として取られている (u, v) は、変形対象の頂点の UV 座標、つまり左上を $(0,0)$ 、右下を $(1,1)$ とした座標です。変形量としては、 x, y だけでなく rot (回転)や $pivot$ (回転の中心)も指定できます。例えば、体を左向きに傾ける `left` というポーズは、以下のような関数で定義されます。

```
1 left: (u: number, v: number) => { ts TypeScript
2     const { fromTop } = getSpatialParams(u, v);
3     const w = curve.power2(fromTop);
4
5     return {
6         tx: -200 * w,
7         ty: 0,
8     };
9 },
```

UV 座標が引数として取れることで、相対的な頂点の位置に基づいた重み付けをすることができます。この実装例で言えば、UV 座標から算出される `fromTop`(対象となる画像の一番上を 1,

一番下を 0 とした値)を 2 乗の曲線に渡し、その値を移動量-200 にかけることで、上半身ほど大きく動き、下半身はあまり動かないという自然な動きをたった 2 行で記述できます。

もし、胴体や胸だけを動かしたい場合は、sin などの山形の曲線を用いて、同じように UV 座標から重みを求めればよいです。

さらに、立体表現のための実装もしています。この基本的な概念を用いて、キャラクターを斜めに傾けたり、一部分を移動させる、といったことは可能ですが、立体表現をするために必要な複雑な変形を表現する曲線の組み合わせを探することは困難です。ここで視差、つまり手前のものは大きく動き、遠くのはあまり動かないという概念を導入し、UV 座標から曲線を使って上半身ほど大きく動く動きを表現したように、UV 座標から手前/奥という情報を判別すれば、移動量をスケールして頂点単位の視差を作り出すことができると考えました。そのため、ブラウザで動作する Depth Anything V2 をライブラリに組み込み、立ち絵から自動で深度情報を推定、UV 座標から手前/奥という情報を判別できる仕組みを実装しました。

7.4 成果

実際にキャラクターが立体的に動くデモを、こちらで公開しています。⁵本ライブラリは開発中で、現在は表情切り替え等の機能に取り組んでいます。

また、本ライブラリに関する発表を第二回全国中高生学術審査会で行い、優秀賞を受賞しました(図 17)。

②: マウス追従

マウスの動きに合わせて、上下左右ポーズを切り替える

- 4 点をなめらかに切り替えるには?
→ 4 点をそれぞれ線形補間した新しいポーズ
- 関数 `lerpPose(a, b, t) ⇒ Pose`
→ ポーズ a, ポーズ b を補間係数 t で補間する高階関数

指示内容

- 上下左右を `lerpPose` で切り替える

```
1 window.addEventListener("mousemove", (e) => {
2   mouseX = e.clientX / window.innerWidth;
3   mouseY = e.clientY / window.innerHeight;
4 });
5
6 app.ticker.add(() => {
7   rig.apply([
8     lerpPose(TEMPLATE.left, TEMPLATE.right,
9       mouseX),
10    lerpPose(TEMPLATE.up, TEMPLATE.down, mouseY),
11  ]);
12 });
```

マウス座標を補間係数 t として
毎フレーム線形補間を実行

11

エンジニアのための2Dキャラクターメッシュ変形ライブラリ
安田陽真 (S高等学校3年)

i2 - 独創する知性 -
第二回全国中高生学術審査会

図 17: 第二回全国中高生学術審査会での発表の様子

⁵<https://marukun712.github.io/kokoro/depth.html>