

公立はこだて未来大学 2025 年度システム情報科学実習 グループ報告書

Future University Hakodate 2025 Systems Information Science Practice Group Report

プロジェクト名

ロボットで未来大を世界にアピール
～エンターテインメントロボット&ドローン～

Project Name

Promoting Future University Hakodate to the world with robots
Entertainment robots & drones

グループ名

ソフトウェア班

Group Name

Software team

プロジェクト番号/Project No.

5

プロジェクトリーダー/Project Leader

吉田怜史/Reishi Yoshida

グループリーダー/Group Leader

柳杭田楓汰/Fuuta Yanaguida

グループメンバ/Group Member

安達涼太/Ryota Adachi

大倉寛晴/Kansei Ookura

藤田悠佑/Yu Fujita

指導教員

山内翔 長田純一 鈴木昭二 西沢俊広

Advisor

Sho Yamauchi, Junichi Osada, Sho'ji Suzuki, Toshihiro Nishizawa

提出日

2026年1月21日

Date of Submission

January. 21, 2026

概要

本学，公立ほこだて未来大学(以下，未来大と表す)は，AI，ロボット分野で強いことが知られており，「IKABO」や「いかみい」のように様々なエンターテインメントロボット(以下，エンタメロボと表す)が開発，発信されている．エンタメロボとは，人と関わり，感情を引き出し，楽しませることを目的としたロボットである．しかし，現状の問題点として未来大のロボットでは十分な知名度が得られていないということが挙げられる．そこで，本プロジェクトは未来大の新たな「顔」となるエンタメロボを開発し，動画やSNSを通じて世界に発信することを目的とした．そして，この目的を達成するため，以下の三つの課題を設定した．一つ目は，新たなキャラクター考案とロボット化をし，エンタメロボとして完成させる．二つ目は，ソフトウェアの改良とクオリティの向上をし，実装を行い，機能の追加を目指す．三つ目は，完成したロボットを世界中に発信するためにプロモーション映像の制作とSNSでの発信をする．以上の三つの課題を解決するためにデザイン班，ハードウェア班，ソフトウェア班にグループ分けをした．ソフトウェア班では，二つ目の課題を達成するためにモーションエディタの引継ぎを行った．このモーションエディタは，誰でも簡単にロボットの動きを編集できるように2024年度のエンタメロボ開発のソフトウェア班によって開発されたものである．ソフトウェア班は7月から9月はモーションエディタの開発をすすめ，基本的な機能の実装を行った．10月以降はモーションエディタのブラッシュアップを進めた．今後は，開発したモーションエディタを未来大の次年度以降の学生やロボット初学者が自由に活用できるようにオープンソースとして公開を行っていく．

キーワード エンターテインメントロボット，Arduino，シリアル通信，Unity，モーションエディタ，Python

(※文責:藤田悠佑)

Abstract

Our university, Future University Hakodate (hereafter, Future University), is known for its strengths in the fields of AI and robotics. We have developed and distributed various entertainment robots (hereafter, entertainment robots) such as "IKABO" and "Ikamii." Entertainment robots are robots designed to interact with people, elicit emotions, and entertain. However, a current issue is that Future University's robots are not yet well-known. So, we aimed to propose and create a new entertainment robot that would become the face of Future University and promote it globally. To achieve this goal, we set the following three tasks: 1) to create a new character, robotize it, and complete it as an entertainment robot; 2) to improve the software and quality, bring it to the implementation stage, and aim to add functions; 3) to create a promotional video and promote the completed robot to the world through social media. The team divided into three groups: design, hardware, and software. To address the three challenges above, the software team took over the motion editor to achieve the second challenge. This motion editor was developed by the 2024 Entertainment Robot Development Software Team to allow anyone to easily edit robot movements. From July to September, the software team worked on developing the motion editor and implemented basic functions. From October onward, they continued to refine the motion editor. Going forward, the developed motion editor will be released as open source so that it can be freely used by students at Mirai University from next year onward and by beginners in robotics.

Keywords: Entertainment robot, Arduino, serial communication, Unity, motion editor, Python

(※文責:藤田悠佑)

目次

第1章 はじめに	1
1.1 プロジェクトの背景	1
1.2 ソフトウェア班の位置付け	1
1.3 現状の問題点の分析	1
1.4 ソフトウェア班の目標	1
第2章 開発の詳細	3
2.1 モーションエディタの機能の提案	3
2.2 開発環境の構築	3
2.2.1 GitHubの共有	3
2.2.2 GitHubでのUnityの共有	3
2.2.3 開発の進め方	3
2.3 モーションエディタ概要	4
2.3.1 システム解説	4
2.3.2 システム図	6
2.4 モーションエディタとロボットの連携の構築	6
2.4.1 シリアル通信	7
2.4.2 ロボット動作制御	7
2.4.3 アクチュエータ制御	7
2.4.4 esp32対応	8
2.4.5 配線	8
2.5 試験用模型作成	9
2.5.1 設計	9
2.5.2 レーザーカッター加工	9
2.5.3 組み立て	9
第3章 結果	11
3.1 中間までの結果	11
3.2 最終結果	11
3.3 アンケート概要	11

3.3.1 アンケート結果	12
3.3.2 アンケート考察	13
3.4 プロジェクトにおける各個人の活動詳細	14
第4章 おわりに	15
4.1 まとめ	15
4.2 展望	15

第1章：はじめに

1.1 プロジェクトの背景

公立はこだて未来大学(以下、未来大と表す)は、AI、ロボット分野で強いことが知られており、「IKABO」や「いかみい」のように様々なエンターテインメントロボット(以下、エンタメロボと表す)が開発、発信されている。エンタメロボとは、人と関わり、感情を引き出し、楽しませることを目的としたロボットである。しかし、現状の問題点として未来大のロボットでは十分な知名度が得られていないということが挙げられる。本プロジェクトの目的は未来大の新たな「顔」となるエンタメロボを開発し、動画やSNSを通じて世界に発信することである。そこで、エンタメロボ「ミラガラ」の開発を決定した。

(※文責:藤田悠佑)

1.2 ソフトウェア班の位置づけ

本プロジェクトは「ミラガラ」を制作し、発信するという最終目標を達成するため、以下の三つの課題を設定した。一つ目は、新たなキャラクター考案とロボット化をし、エンタメロボとして完成させる。二つ目は、ソフトウェアの改良とクオリティの向上をし、実装を行い、機能の追加を目指す。(理由を1.3, 1.4にて後述)三つ目は、完成したロボットを世界中に発信するためにプロモーション映像の製作とSNSで発信をする。以上の課題を解決するためにデザイン班、ハードウェア班、ソフトウェア班にグループ分けをした。本グループ(ソフトウェア班)はソフトウェア面での問題点(1.3 節にて後述)を解決するために活動した。

(※文責:藤田悠佑)

1.3 現状の問題点の分析

1.1節で述べたように、未来大のロボットでは十分な知名度が得られないという問題がある。そこで、ソフトウェア班は原因を昨年度までのロボットは人とロボットをつなげるインタラクションの仕方がわかりにくかった点にあると分析した。

(※文責:藤田悠佑)

1.4 ソフトウェア班の目標

ロボット開発において、ユーザが直観的に操作が分かるユーザインターフェースを実装することは難しい。そこで、我々はユーザとロボットのインタラクションを円滑にすることを目的

として、ユーザがロボットの動きを直観的に操作できることを要件とした、ロボットの動きの編集のプラットフォーム（以下、モーションエディタと表す）を設計しミラガラに搭載させることを目標とした。本稿の目的は、モーションエディタとそれに付随する外部機器の開発の手法と、ユーザからのフィードバックを元にした考察を提示することである。

（※文責：藤田悠佑）

第2章：開発の詳細

2.1 モーションエディタの機能の提案

作成するモーションエディタの要件はロボットの動きを自由に編集できることかつ、ロボット工学初学者でも扱いが簡単なことである。モーションエディタの機能としてタイムラインに沿ってロボットの動きを編集できるエディタアプリを要件とした。タイムライン型のエディタアプリは視覚的な表現がしやすい。そのため、ロボット工学初学者でも理解が簡単で、モーションエディタに必要な要件を満たしているといえる。

(※文責:藤田悠佑)

2.2 開発環境の構築

モーションエディタのプラットフォームであるunityとそれを編集するためにVisual Studio Codeを各メンバーで導入した。その後、シリアル通信のためにPython, ロボットを動かすためにArduino IDE, コードの管理の効率化のためにGitHub Desktopを導入した。

(※文責:安達涼太)

2.2.1 GitHubの導入

事前にリポジトリを共有することで、全メンバーが同一のデータを基に作業を開始できる環境を整えた。これにより、ファイルの不足やバージョンの違いによる混乱を防ぎ、作業開始時のトラブルを最小限に抑えることができた。また、GitHubを通じて最新版のデータを取得するという手順を統一することで、作業フローの明確化にもつながった。

(※文責:柳杭田楓汰)

2.2.2 GitHubでのUnityの共有

Unityはプロジェクト単位での管理が必要なため、個別にデータを受け渡しする方法では管理が煩雑になる。その点、GitHubを利用することで、変更履歴を残しながら複数人で同時に開発を進めることが可能となった。各メンバーは担当部分を分担して作業し、更新内容を共有することで、進捗状況の把握や修正点の確認が容易になった。結果として、チーム内の連携が強化され、効率的なグループワークの実施につながった。

(※文責:柳杭田楓汰)

2.2.3 開発の進め方

昨年度に開発されたモーションエディタを改良するにあたり、まず当該モーションエディタをダウンロードし、スクリプトコードを読み込んで理解を行った。コード量が多かったため、メンバー間で担当箇所を分担し、それぞれが内容を読み込んだうえで、理解した内容を他のメンバーに説明する形で作業を進めた。

(※文責:安達涼太)

2.3 モーションエディタ概要

作成したモーションエディタ(※図1参照)はUnity上で動作するタイムライン型制御エディタである。ユーザが画面上のアイコンをレーンにドラッグアンドドロップしてロボットのモーションの再生開始時間、再生時間、値をコントロールできる。モーションエディタは入力されたデータを元に3Dモデルでモーションのシミュレーションを提供する。また、ミラガラ本体に接続することで、モーションエディタで入力されたデータを元にミラガラを動かすことが可能である。

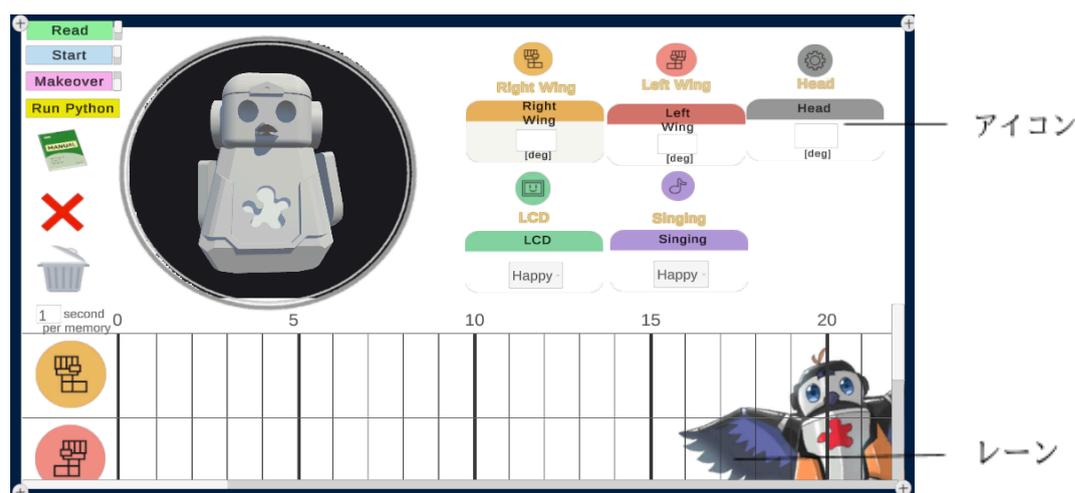


図1モーションエディタ画面

ユーザーはアイコンをレーンにドラッグ&ドロップしてモーションを編集できる

(※文責:藤田悠佑)

2.3.1 システム解説

本システムは28個のクラス(基幹となるクラスは15個ほど)から成り立っている。以下ではそれぞれのクラスの役割について説明する。

①UI操作・編集

- UISetting : EventSystemを用いて, ドラッグ&ドロップ, 拡大, 縮小などの機能を担う抽象クラス.
- Spawner : アイコンの複製を担うクラス.
- DeleteAllIconOnLane : アイコンの一括削除を担うクラス.
- Memory : レーンの一メモリあたりの秒数を管理するクラス.

②データ保存

- IconData : UISettingを継承したアイコンの座標を再生開始時間, 再生時間のデータに変換する抽象クラス.
- RightWingIcon/LeftWingIcon/HeadIcon/LCDIcon/SingingIcon : IconDataを継承したクラス群. IIconインターフェースで管理している.

③再生制御

- Playback : 3Dモデルの再生開始, 停止, 再開, 時間管理を担うクラス.
- Reset : 3Dモデルを初期位置にリセットする機能を担うクラス.

④実行部

- Motors : 3Dモデルの右羽, 左羽, 頭を回転させるクラス.
- Audio : 感情に即した音楽を流すクラス.
- LED : お腹の未来大マークを感情に即した色に点灯させるクラス.

⑤表示系

- Redline : 3Dモデル再生時に赤線でレーン上の再生位置を指し示す機能を担うクラス.
- SevenSegmentLED/SevenSegmentLEDManager : 3Dモデル再生時に7セグメントLEDで再生時間を表示するクラス.
- ScreenManager : 矢印キーを押したときに3Dモデルを回転させるクラス.
- Manual : 本システムの説明書を表示させるクラス.

⑥ファイル入出力

- FileAbstract : ファイル名入力などのファイル操作を担う抽象クラス.
- FileInput : 入力されたファイル名に記されたCSV形式のモーションのデータからUIのセッティングを行う.
- FileOutput : 現UIの状態をCSV形式のファイルに変換する.

⑦外部連携

- PythonManager : CSV形式のモーションのデータをミラガラに送信し, 動かすクラス.

⑧エフェクト機能

- Background : モーションエディタ画面の背景の切り替えを行うクラス.
- Background_emo : モーションエディタ画面を虹色にするクラス.

- Rechord_Gear : 歯車を回転させるクラス.
- ScrollView : 画面構成を変換するクラス.

(※文責:藤田悠佑)

2.3.2 システム図

モーションエディタの全体システム図を提示する。(※図4参照)

システムは5層で構成されている.

1. UI制御層 : ユーザ操作を受け付ける.
2. 3Dモデル制御層 : ロボットの動作のシミュレーションを行う.
3. ファイル入出力層 : CSVデータの読み書きを行う.
4. 動作実行層 : モータ、LED、音声の制御を行う.
5. ミラガラ制御層 : ミラガラ本体へデータを送信する.

情報は上から下へ伝達される.

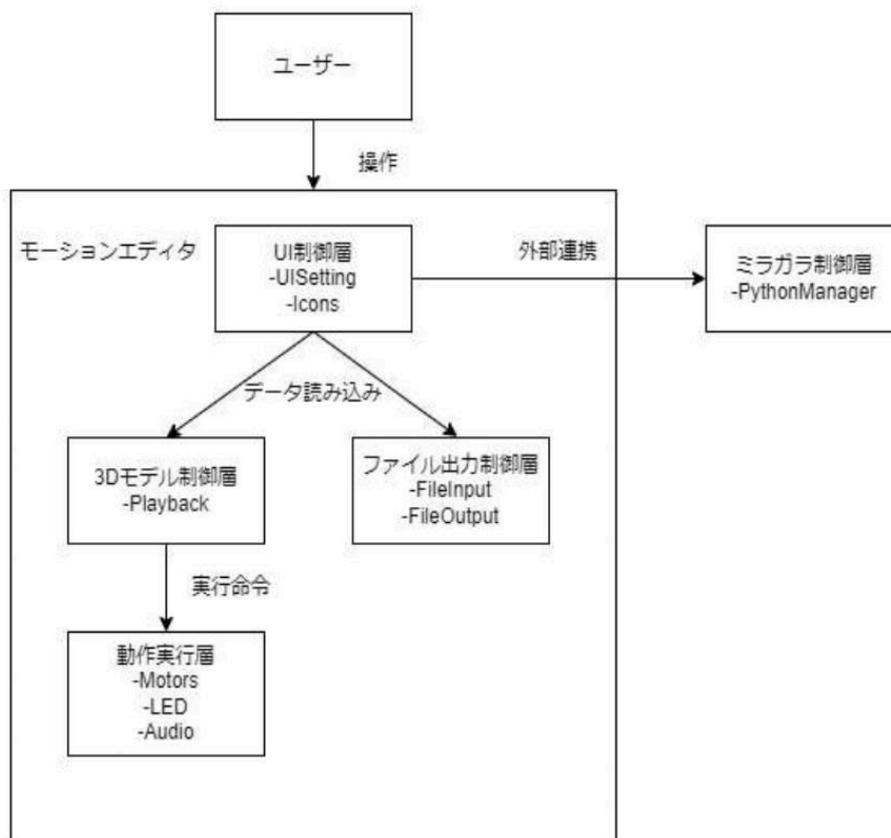


図2モーションエディタの全体システム図

(※文責:藤田悠佑)

2.4 モーションエディタとロボットの連携の構築

次に、モーションエディタで編集したデータをミラガラに転送して、ミラガラを動かすための機構とプログラムの製作に取り掛かった。

(※文責:藤田悠佑)

2.4.1 シリアル通信

モーションエディタで作成した各モーションの情報をテキストデータとして記録し、そのデータをArduinoに送信する仕組みを構築した。送信されたテキストデータはArduino側で解析され、サーボモータなどの制御に反映されるため、エディタ上で想定した動きを実機上でも再現できるかを確認した。この過程では、データ形式や送信タイミングの調整を行い、動作のずれや誤差が生じた場合には原因を特定して修正するなど、想定どおりに動作させるための補助を行った。

(※文責:柳杭田楓汰)

2.4.2 ロボット動作制御

M5Stackを用いて、ミラガラに必要となる目や腹部の未来大のマークをモニターに表示させるためやミラガラの羽を動かすためにコーディングを行った。コーディングだけでなく配線もロボットがしっかり動いてくれるかどうかにかかわってくるため、適切に配線されているのかどうかを確認しながら活動を行った。

(※文責:大倉寛晴)

2.4.3 アクチュエーター制御

Arduinoを用いて各モータの制御を行った。まず、モーションエディタから出力されるデータをもとに、各モータが動作を開始するタイミング、動作の経過時間、および最終的な到達角度を整理・解析した。これらの情報をもとに、モータが滑らかかつ正確に動作するように、動作全体を1秒単位に分割し、各時刻における目標角度を計算した。

その際、開始時間と終了時間の差から総移動時間を求め、角度の変化量を時間で割ることで、1秒ごとに回転させる角度を算出し、Arduino上で逐次制御を行った。これにより、複数のモータがそれぞれ異なるタイミングと角度で動作する状況においても、意図したモーションを正確に再現することができた。

最終的には、モーションエディタで設計した動きと実機の動作が一致していることを確認し、モータが想定どおり正確に動作することを実証した。

(※文責:柳杭田楓汰)

2.4.4 esp32対応

もともと本プログラムはArduinoを用いて各モータを制御することを目的に作成していたが、処理性能や今後の拡張性を考慮し、ESP32への対応を行った。ArduinoとESP32ではPWM制御方式やピン構成が異なるため、そのままでは既存の制御プログラムを動作させることができず、制御方法の見直しが必要となった。

そこで、ESP32ではanalogWriteが使用できない点に対応するため、ESP32Servoライブラリを導入し、PWM周期を50Hzに設定したうえで各サーボを制御する方式に変更した。また、サーボごとに使用するピンを明示的に指定し、初期位置を統一することで、ハードウェア差による動作のばらつきを抑えた。

さらに、モーションエディタから送信される開始時間、動作時間、目標角度のデータを構造体として管理し、モーション再生時には秒単位で設計された動作をもとに、一定間隔で角度を補間する方式を採用した。この設計により、ESP32上でも処理負荷を抑えつつ、複数のサーボを同時に安定して制御できるようになった。

その結果、Arduino環境で作成したモーションをESP32上でも正確に再現することができ、ハードウェアの違いを意識したプログラム設計の重要性を理解することができた。

(※文責:柳杭田楓汰)

2.4.5 配線

配線については、開発初期段階で全体構成を十分に計画せず、ジャンプワイヤを用いてその都度接続を行ったため、配線が複雑になり視認性が大きく低下してしまった。その結果、配線の経路や接続状態を一目で把握することが難しく、本番環境において一部の配線が断線するトラブルが発生した。

このトラブルにより、ソフトウェア自体は正しく動作しているにもかかわらず、モータが意図したとおりに動作しない状況が生じ、ハードウェア面の信頼性がシステム全体の動作に大きく影響することを実感した。特にジャンプワイヤは抜けやすく、振動や取り回しの影響を受けやすいため、実運用には不向きであるという課題が明らかになった。

今回の経験から、回路設計の段階で配線計画を立て、配線の整理や固定を行うことの重要性を学んだ。今後は、ブレッドボード上での整理や配線色の統一、コネクタやユニバーサル基板の使用、ケーブルの固定などを行うことで、視認性と耐久性を向上させ、本番環境でも安定して動作する構成を目指したいと考えている。

(※文責:柳杭田楓汰)

2.5 試験用模型製作

中間発表前の時点で他班で模型を作成する予定や余裕が無かったため、進行状況に十分の猶予があったメンバーで中間発表に向けた簡単な模型を作成した(※図3参照)。

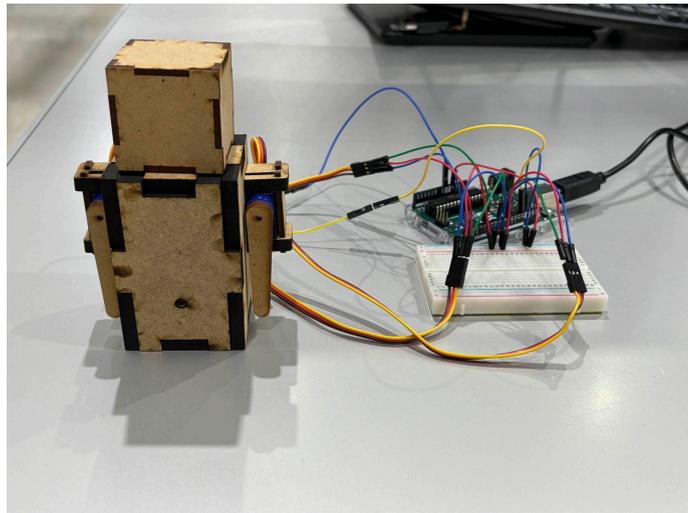


図3試験用模型

(※文責:安達涼太)

2.5.1 設計

実装する機能は首と両腕を制御する三つのサーボモータとして、本体はMDFで作成した。具体的な設計はAdobe Illustratorで行い、MDFに切り抜き線を印刷した。

(※文責:安達涼太)

2.5.2 レーザーカッター加工

レーザーカッター加工は学校の設備を利用し、監督者のもとで行った。MDF上の切り抜き線に沿って切断し、監督者に手本を見せてもらった後に自らで行った。

(※文責:安達涼太)

2.5.3 組み立て

組み立ては接着剤等を使用せずにはめ込むことのできるように、サーボモータも凹凸を利用してはめ込めるように設計した。想定通り勝手に外れないように組み立てることはできたが、サーボモータの形状に認識漏れがあったため、背中部分が完全にはめ込むことが出来なかった

Entertainment robots & drones

. 作り直しても中間発表に間に合わないことと大きな問題は無く利用できるため、これで完成とした。

(※文責:安達涼太)

第3章：結果

3.1 中間までの結果

中間までの活動では、モーションエディタの解析，Unityの技術取得，シリアル通信の技術取得を行い学習した技術を用いて，モーションエディタの基本的な機能の完成，テキストデータを用いてモーターを動かせるロボットの開発を行った。

(※文責:大倉寛晴)

3.2 最終結果

プロジェクト全体の目標に向けて，モーションエディタのUIデザインやミラガラの5つの機能に対応する3Dモデルの作成および機能の実装を行った。（※図4参照）また，モーションエディタからミラガラ本体を動かせるようにシリアル通信のPythonのコードを実行できるようにした。

Arduinoでは，モータの動作を可能な限り滑らかに動くようにし，モニタで目が表示されるようにした。



図4完成したモーションエディタ

(※文責:大倉寛晴)

3.3 アンケート概要

12月5日に実施された成果発表会にて成果発表評価シートにモーションエディタについての項目を追加した。アンケートを未来大生の生徒，教員に対して実施した。追加した項目は「モーションエディタに表示された情報は理解しやすかったか」，「モーションエディタの操作の流れは自然だったか」を10段階のリッカート尺度によって評価する項目と，モーションエディタの不満点改善点についての自由回答形式の項目の計3つである。被験者に対し，モーションエ

ディタについての簡単な説明と，デモンストレーションを行いアンケートに答えてもらった。回答者は65名である。3.3.1にその結果，3.3.2に結果から得られる考察を記す。

(※文責:藤田悠佑)

3.3.1 アンケート結果

以下にアンケート結果を記す。

質問内容 項目	モーシヨンエディタに表示された情報は理解しやすかったですか？ / Was the information displayed in the Motion Editor easy to understand?	モーシヨンエディタの操作の流れは自然でしたか？ / Was the workflow of the Motion Editor natural?
10	12人	10人
9	3人	8人
8	11人	6人
7	14人	12人
6	8人	11人
5	9人	11人
4	6人	1人
3	0人	3人
2	0人	0人
1	0人	0人
無回答	2人	3人

表1アンケート結果表

「モーションエディタについて不満点・改善点があれば教えてください。 / Please let us know if you have any complaints or suggestions for improvement regarding the Motion Editor.」の回答結果

- UIが見にくい
- あまり画面が見えなかったので書けないのですが少し情報量が多い気がしました。
- 実行しているラインが進むとき、画面の端に揃えて動いているので見にくい。画面は実行しているラインよりも少し先を見せるべきだと感じた。
- 操作の流れはわかりませんでした。
- モーションエディタについてもう少し詳しい説明が欲しかったかもしれない。
- 面白かった。
- 面白かったが、何がなんなのか遠目だと分からなかった。
- 日本語が欲しい。
- それぞれのコマンドについての説明が欲しかったです。
- 動いてるのを見たかった。
- すみません、まったくわからなかったです。
- パッと見ちょっと分かりにくいかも。説明受けたり使い慣れると使い勝手は良さそう。
- 特になし、すごかった。
- 腕ごとに背景色が違うが、なぜその色にしたのか。
- ボタン自体が小さいのと左側にまとまっているせいでぱっと見で操作がわかりづらい。
- 左側のボタンが楕円で余白がありすぎるので孤立したボタンに見えてつながりが見えない。
- いつ説明したか分からなかった、ごめんなさい。
- あまりよく見るできませんでした。
- いつ説明があったかわかりませんでした。
- モーションエディタの説明をよく聞き取れなかった。
- Scratchなどのよく使われがちなUIに寄せた方が分かりやすそうな気がしました。
- 独自のUIは使い方を把握するのに時間がかかりそうです。
- 設定段階ではスムーズだったとのこと。

(※文責:大倉寛晴)

3.3.2 アンケート考察

UIが見にくいや実行状況の可視化方法についての意見、日本語が欲しいやScratchのようなUIに寄せた方が分かりやすいといった親しみやすさに対しての意見が多く見られた。また、説明

に関しても詳しく伝えることができなかつたため、初めて見る人たちにとってどのようなものなのか理解してもらえなかつた。このような意見から、モーションエディタは完成度は高くできたが、初めて使う人に対しての配慮が不足していることが考えられる。

(※文責:大倉寛晴)

3.4 プロジェクトにおける各個人の活動詳細

本グループは以下のように作業分担を行い、活動を進めた。

柳杭田楓汰

- ソフトウェア班リーダーを担当
- 毎プロジェクト学習時の作業内容の指示
- テキストデータからArduinoを動かす係
- 小型模型の製作

大倉寛晴

- モーションエディタのスキプトの解析
- 「ミラガラ」の3DモデルのUnityへの導入
- 3Dモデルを動かすプログラミングの実装
- モニターとモーターが正しく動くためのコード作成

安達涼太

- モーションエディタのスキプトの解析
- 小型模型の製作
- pythonでのシリアル通信のコード
- arduinoでのシリアル通信の受信のコード

藤田悠佑

- Unity の技術習得
- モーションエディタのコーディング
- モーションエディタのUI作成

第4章：おわりに

4.1 まとめ

本班の目的はユーザとロボットのインタラクションを円滑にすることであり、目標はユーザがロボットの動きを直観的に操作できることを要件とした、ロボットの動きの編集のプラットフォームを設計しミラガラに搭載させることである。本班は、活動期間内に要件を満たしたモーションエディタを完成させることができた。しかし、成果発表時にはモーションエディタからミラガラを動かす機構に不具合があったことが反省点としてあげられる。また、完成したモーションエディタはアンケート結果からリッカート尺度の項目からはロボット工学初学者でも扱いやすいモーションエディタというコンセプトを満たしていると読み取れる。しかし、不満点・改善点を書いてもらう項目からは現状のモーションエディタには改善点がいくつかあることが分かる。

(※文責:藤田悠佑)

4.2 展望

モーションエディタ部分においてはUIやデザイン等の使いやすさを向上させる改良が求められる。現状は機能の実装を優先したために不完全であるが、モーションエディタのコンセプトはユーザが直感的に操作が分かることであるため重要性は高い。また、来年度以降もモーションエディタを継承することも想定し、コードのコメントを書くことも必要である。

Pythonのシリアル通信用のコードでは必要な機能は実装できた。改善点があるとすれば、コードやシリアル通信で送信するテキスト、これらの最適化である。

Arduinoではモニタでの動画再生、モータの動作の円滑化が優先される改善点である。現状、モニタでは描画をプログラミングで指示している状態、モータは整数でしか角度を制御できないために動きが滑らかでなくなっている状態となっている。また、初期案にあったが技術的に実装を諦めた、ロボットが人の感情を認識するインタラクションもゆくゆくは実装したいところである。

(※文責:安達涼太)