

公立はこだて未来大学 2018年度 システム情報科学実習
グループ報告書

Future University Hakodate 2018 System Information Science Practice
Group Report

プロジェクト名
複雑系知能ロボットへの挑戦
Project Name
Challenge to Complex Intelligent Robot

グループ名
グループB
Group Name
GroupB

プロジェクト番号/Project No.
12

プロジェクトリーダー/Project Leader
1016130 平野香介 Kyosuke Hirano

グループリーダー/Group Leader
1016055 柳川大輝 Daiki Yanagawa

グループメンバ/Group Member
1016189 石崎智也 Tomoya Ishizaki
1016012 佐藤佑哉 Yuuki Satou
1016034 野中和馬 Kazuma Nonaka

指導教員
松原仁 鈴木恵二
Advisor
Hitoshi Matsubara Keiji Suzuki

提出日
2019年1月16日
Date of Submission
January. 16, 2019

概要

私たちは本学のゴミ問題を解決するために次世代ゴミ箱ロボットを開発することにした。そのためにメカ班はロボットの制御や外装の制作、画像の解析を行うことを目標とした。またソフト班では、そのロボットと連動するアプリケーション、シミュレーターを開発することを目標として活動している。

(※ 文責:柳川大輝)

Challenge to Complex Intelligent Robot

Abstract

In order to solve the problem of the garbage of our university we have tried to develop the next generation trash box robot. For that purpose, the Mecha Group aimed to control the robot, produce the exterior, and analyze the image. The software team also works with the goal of developing applications and simulator that work with the robot.

(*Responsibility for wording: Daiki Yanagawa)

Challenge to Complex Intelligent Robot

1. はじめに	
1.1. プロジェクト概要	6
1.2. グループ概要	6
1.3. 背景	6
1.4. プロジェクト目的	6
1.5. グループ目的	6
1.6. 課題	7
2. 到達目標	7
2.1. 問題の設定	7
2.2. 課題の設定	7
2.3. 到達レベル(目標)	8
2.4. 課題の割り当て	8
3. 課題解決のプロセス (概要)	9
3.1 前期のプロセスについて	9
3.1.1. 柳川	9
3.1.2. 野中	10
3.1.3. 佐藤	10
3.1.4. 石崎	10
3.2 後期のプロセスについて	10
3.2.1. 柳川	10
3.2.2. 野中	11
3.2.3. 佐藤	11
3.2.4. 石崎	11
4. 課題解決のプロセス (詳細)	12
4.1. 柳川	12
4.2. 野中	14
4.3. 佐藤	17
4.4. 石崎	21
5. 結果	24
5.1. 成果	24
5.2. 解決手順と評価	25
5.2.1. 前期	25
5.2.2. 後期	26
6. まとめ	26
6.1. プロジェクトの成果	26

Challenge to Complex Intelligent Robot

6.2. プロジェクトにおける各人の役割	27
6.3. 今後の課題	27
6.3.1. アプリケーション	27
6.3.2. シミュレーター	28
6.3.3. 発表に関する反省点	29

1. はじめに

1.1. プロジェクト概要

“複雑系知能ロボットへの挑戦”プロジェクトは、人工知能とロボットの融合による知能ロボット作成に取り組む。本大学ではゴミ箱周りの汚さが目立っている状況である。近年動画サイトで、自動でゴミの落下地点に移動するゴミ箱が有名になりました。本プロジェクトの目的は次世代知能ロボットとしてこれを参考に、自動で手元に来る、その道中障害物を避けることができるゴミ箱を開発することである。現在ソフト班、メカ班の二つのグループに分かれて活動を行っている。

(※文責：石崎智哉)

1.2. グループ概要

私たちは本学のゴミ問題を解決するために次世代ゴミ箱ロボットを開発することにした。そのためソフト班では、前期ではそのロボットと連動するアプリケーションを開発することを目指して活動している。後期ではソフト班はゴミ箱ロボットの『呼び出し』を行う補助アプリの開発と、ロボットの制御方法を模索するためのシュミレータの作成した。

(※文責：石崎智哉)

1.3. 背景

今日、公立はこだて未来大学のゴミ問題は深刻である。ゴミ箱のごみがあふれていたり、ごみをそのままにして机に放置してあったり、ごみの分別ができていないことが多い。そこで、ソフト班は次世代ゴミ箱ロボットを呼び出すための機能を搭載した個人用アプリケーションの開発とロボットの制御方法を模索するためのシミュレータの制作を目標として活動している。

(※文責：石崎智哉)

1.4. プロジェクト目的

ゴミ箱ロボットを開発することによりゴミの収集率を向上させ本学のゴミ問題の解決に少しでも貢献する。

(※文責：柳川大輝)

1.5. グループ目的

ソフト班は4つの課題のうちの『移動』と『呼び出し』を実装する役割を担っている。シミュレータを用いて仮想空間内での自動走行の『移動』を実装し、ライトを用いて『呼び出し』を実装することがソフト班の目的である。

(※文責：柳川大輝)

1.6. 課題

前期ではソフト班の活動にあたって以下の課題が挙げられる。

- ・ロボットと連携するアプリケーションの基礎部分の開発
- ・ロボットとの連携を行える機能の開発

また後期では以下の課題が挙げられる。

- ・ロボットを使用者の場所まで誘導する機能開発
- ・ロボットの行動制御をシミュレートするソフトの開発

(※文責：柳川大輝)

2. 到達目標

2.1. 問題の設定

本プロジェクト学習では、ゴミ箱ロボットを作ることを目指す。そのために我々ソフト班ではロボットの制御を行い、ロボットの活用を促進できるアプリケーションの開発を行うことを目指す。また、ロボットの行動を仮想空間上で模索するシミュレータを作成することで実際のロボットの行動制御をよりよく改良できるようにすることを目指す。

(※文責：野中和馬)

2.2. 課題の設定

2.1節で述べた問題を解決するために前期では以下の条件を満たす機能を含むアプリケーションの開発を行う。

- ・ロボットを制御できる機能
- ・ロボット制御のほかにアプリケーションの使用を促進する機能

上記の条件下で具体的に挙げられた機能は以下の通りとなった。

Challenge to Complex Intelligent Robot

- ・ゴミ箱をアプリケーション使用者のもとへ呼び出すまたは、元の位置へロボットを戻す機能
- ・現在のゴミの量をアプリケーション上で確認できるメーター
- ・今日、何のゴミを捨てる日かを表示させる機能
- ・函館の地域に設定されたゴミ出しの日を確認できるゴミ出しカレンダーを表示する機能
- ・上記のカレンダーを設定する際に必要な地域の設定を行う機能
- ・1日毎のゴミの量を集計し、グラフとして表示させる機能

以上の機能をアプリケーション上で動くのように制作することを課題とした。

また、これらの機能を実装して稼働するアプリケーションの開発の手法を取得することも課題として挙げられる。

前期では上記の機能実装を行ったが、実装する機能が多く、制作が困難になることが予想されたため後期ではゴミ箱を使用者のもとへ呼び出す機能の作成に重点をおいて作成を行った。また、アプリ班内で機能作成とは別にシミュレータ制作を行うグループも設置し、シミュレータ作成に当たった。

(※文責：野中和馬)

2.3. 到達レベル(目標)

前期はロボットと連動するアプリケーションの完成を目標とした。そのために以下の項目を満たせることを目指す。

- ・2.2で挙げた機能の開発、実装
- ・アプリケーションの開発手順の理解

後期では呼び出し機能の作成とシミュレータの完成を目標とした。そのために以下の項目を満たせることを目指す。

- ・ロボットを呼び出すための手法の模索、作成
- ・仮想空間上でのロボットの行動制御シミュレータの作成

(※文責：野中和馬)

2.4. 課題の割り当て

前期では各々で異なる機能の開発を行い、アプリケーションに実装することとした。それぞれ1つずつ割り当てた。作業量が多いと考えられる機能には複数人を割り当てることにした。割り当ての結果は以下の通りとなった。

Challenge to Complex Intelligent Robot

石崎：函館の地域の設定を行える機能の作成
佐藤：ゴミ出しカレンダーを表示する機能の作成
下田：ゴミ出しカレンダーを表示する機能の作成
野中：アプリケーションのホーム画面の作成、機能の統合
柳川：ゴミの量を確認できるメーターの制作

割り当てていない機能の一部はロボットとの連携が必須であるために制作が難しいため、割り当てを保留とした。それ以外の機能については手が空き次第他の機能の作成をすることとした。

後期ではアプリケーションの開発とシミュレータの作成の2つに分かれて各々作成を行うこととした。アプリケーション作成とシミュレータ作成の分担は以下の通りとなった。

アプリケーション：野中 柳川
シミュレータ：石崎 佐藤

(※文責：野中和馬)

3. 課題解決のプロセス（概要）

各メンバーの割り当てられた課題の解決のためのプロセスについて述べる。

(※文責：柳川大輝)

3.1 前期のプロセスについて

3.1.1. 柳川

- (1) 開発内容を話し合い決定。
- (2) unityの参考書（参考書名）を用い勉強し開発の準備を行った。
- (3) ゴミ箱ロボットのアプリケーション開発
- (4) アプリケーション内のイラスト作成

(※文責：柳川大輝)

Challenge to Complex Intelligent Robot

3.1.2. 野中

前期では以下のプロセスを経て課題解決を行った。

- (1) アプリケーションの制作方法を学習し、ホーム画面の作成を行う。
- (2) ホーム画面上や他の画面に制作した機能の統合を行う。

3.1.3. 佐藤

前期は以下のプロセスを経て課題解決を行った。

- (1) グループでの話し合いによるアプリに実装する機能の決定
- (2) C#の勉強
- (3) 函館市内に対応しているゴミ出しカレンダーを地域の選択によって表示させるために必要であるカレンダーの種類ごとに地域をわけたハッシュマップを作成する。
- (4) カレンダー作成担当と地域選択画面制作担当と話し合い調整

(※文責：佐藤佑哉)

3.1.4. 石崎

前期は以下のプロセスを経て課題解決を行った。

- (1) 制作するアプリの機能決定。
- (2) Unityの勉強。
- (3) 函館市の地域に設定されたゴミ出しの日を確認できるゴミ出しカレンダーを作るために必要となる地域を設定できる画面を制作する。
- (4) 上記で作成した地域設定とカレンダーを連動させて地域別のゴミ出しカレンダーを表示できるようにする。
- (5) 上記の機能とアプリの本体との連結。

(※文責：石崎智哉)

3.2 後期のプロセスについて

3.2.1. 柳川

- (1) ゴミ箱を呼び出すための機能の作成
 - 1.1 LEDを起動させるための調査
 - 1.2 ライトレーサーについての調査

Challenge to Complex Intelligent Robot

- (2) Unityでの実行
- (3) XCodeでの実行

(※文責：柳川大輝)

3.2.2. 野中

後期では以下のプロセスを経て課題解決を行った。

- (1) ロボット班を協議を行いロボットの呼び出し方法を決定する。
- (2) 決定した呼び出し方法にあった手法を模索し、検証を行う。
- (3) 検証結果を元に機能の実装を行う。

3.2.3. 佐藤

後期は以下のプロセスを経て課題解決を行った

- (1) 本ロボットの実際の動きの大まかな想定、現在不可能である本ロボットの複数台を運転した際の各ロボットの動きの予想のためにシミュレータが必要になってくる。そのために必要な方法を模索した。
- (2) ロボット班に実際のロボットの動きを伺う。
- (3) Unityの勉強
- (4) 前述で模索した方法に必要な関数などを検索し、シミュレータに応用する形で使用し、動作を確認する。
- (5) シミュレータの動きをロボット班に聞いた内容とできるだけ沿っているかを確認してシミュレータを修正などをしていく。

(※文責：佐藤佑哉)

3.2.4. 石崎

後期では以下のプロセスを経て課題解決を行った。

- (1) 実際のロボットの移動のアルゴリズムの決定。
- (2) ロボットの動きを確認するためのタイヤなしの簡易的なシミュレータの制作。
- (3) より再現率を高めるためにタイヤ付きのロボットのシミュレータの制作。
- (4) 実際の稼働を想定した仮想空間ならではの複数台同時稼働の実装。

(※文責：石崎智哉)

4. 課題解決のプロセス（詳細）

ここでは各個人の課題解決のプロセスについて詳細に述べる。

（※文責：石崎智哉）

4.1. 柳川

前期

(1)

プロジェクト作成物の決定後ソフト班とメカ班にグループを分けソフト班では、アプリケーションの機能面、デザインについての話し合いの結果、ロボットを制御できる機能、ロボット制御のほかにアプリケーションの使用を促進する機能を作成することに決定した。この、二点の条件下で具体的に挙げられた機能は以下の通りとなった。

- ・ゴミ箱をアプリケーション使用者のもとへ呼び出すまたは、元の位置へロボットを戻す機能
- ・現在のゴミの量をアプリケーション上で確認できるメーター
- ・今日、何のゴミを捨てる日かを表示させる機能
- ・函館の地域に設定されたゴミ出しの日を確認できるゴミ出しカレンダーを表示する機能
- ・上記のカレンダーを設定する際に必要な地域の設定を行う機能
- ・1日毎のゴミの量を集計し、グラフとして表示させる機能

これらの機能を開発する環境の候補に以下が上がった。

Python, Unity, Xcode, AndroidStudio

グループ内での話し合い、調査の結果Unityに決定した。理由は以下の3点である。

- ・Arduinoとの連携可能
- ・高い拡張性
- ・ロボットに主に使用される言語c#の使用が可能

(2)

開発環境の決定後、Unityの勉強をグループメンバーで行った。その際に、ライブラリーで借りた参考書を用いた。Unityの勉強をすることによって、初めてUnityを使用するメンバーもいたので開発に各々が取り組むときに困らない最低限度の知識を得ることができた。

(3)

Challenge to Complex Intelligent Robot

アプリケーション開発では、アプリケーションのホーム画面の作成をした。ゴミ箱ロボットを呼び出す際に使用するアプリケーションの画面である。

その際、Unityに初期状態からある画面の移動システムを用いた。これを用いることによって、アプリケーションのホーム画面からカレンダー、ゴミ収集グラフ、地域選択画面、ゴミの量を測るメーターへの移動を可能にした。

(4)

ゴミ箱のイラスト作成では、イラストレーターを用いた。その際、イラストレーターを使用するのは初めてだったのでイラストレーターの知識がある知人にやり方を教わった。

イラスト作成では、インターネット上にあるゴミ箱の画像を使用しその上をイラストレーターで型取り作成した。その後、ゴミメーターでイラストを使用するためにゴミの量が目視で分かるようにゴミ箱を縦に4分割し線を引き1/4に色を塗ったイラスト、1/2に色を塗ったイラスト、3/4に色を塗ったイラスト、すべてに色を塗ったイラストの4段階を作成した。

これによりゴミ箱のメーターをゴミの量に推移して表示することが可能になった。

以上が前期における活動である。

後期

(1)

1.1

後期では、ゴミ箱ロボットを呼び出すために使用するアプリケーション内の機能の作成を行った。ゴミ箱を呼び出す機能としてライントレーサーの機能を利用しゴミ箱ロボットにLEDライトを認識させて呼び出す方法をとった。決定までの経緯は、以下である。

ゴミ箱ロボットの呼び出し方法を考えている議論中にスマートフォンで呼び出す方法を考え、その中で何か目印となるようなものをスマートフォンで発信することはできないかと考えWi-Fiでゴミ箱ロボットとスマートフォン内の呼び出し機能を接続しようと考えたが、Wi-Fiだとスマートフォンを使用している発信者の位置が正確ではないので、その他の案を模索しているとライントレーサーの機能を知り、ライントレーサーを応用してLEDライトを使用しゴミ箱の呼び出しを行うことに決定した。

1.2

ライントレーサーとは紙の上に引かれた黒線に沿って自動的に走る模型自動車のことをいい3輪車である。左右の後輪がそれぞれのモーターで回転するので真直ぐ走ったり、右や左に方向を変えることができる。また、前方にラインセンサーが2個付いており、ラインセンサーによってどちらに曲がれば良いのかを識別することができ、電子回路で左右のモーターを回転させたり止めたりして黒線に沿って走行することが可能である。この機能を使用しゴミ箱ロボットにLEDライトを識別させゴミ箱ロボット内のセンサーの中心にLEDライトが来るように走行することによりゴミ箱の呼び出しを可能にした。

(2)

1.2の様に機能の設計ができたのでUnityを使用し、スマートフォン内のアプリケーションでLEDライトを点灯させることを目標にゴミ箱の呼び出しアプリケーションの開発を始めた。Unityで

Challenge to Complex Intelligent Robot

のアプリケーション開発を進めているとUnity内では、カメラを起動することは可能であったがUnity内の機能のみでスマートフォンのLEDライトを制御することが難しいと分かったため以下の方法で実行することを考えた。Xcodeを使用してiPhone内のLEDライトを制御することが可能なiOSのアプリケーションを作成しその後、Unity内にXcodeで作成したアプリケーションを移行させUnity内で作成したアプリケーションとの連結を行い、ゴミ箱ロボットの呼び出しを可能にする事ができるアプリケーションにしようと考えた。

(3)

(2)でXcodeを使用してゴミ箱ロボットを呼び出すためのLEDライトの点灯が可能なアプリケーションの作成が決定したので、Xcodeを使用しアプリケーションの開発を始めた。

LEDを点灯させるアプリケーションは、Xcode内の機能を利用し開発することができたが、プロジェクト内のタブレット端末が、iOSでなくアンドロイド対応だったためXcodeを断念しAndroid Studioでの開発にシフトした。AndroidStudioでの開発は同じグループメンバーである野中が開発を行ったがUnityとAndroidStudio内のLEDライト点灯アプリケーションの連結がいまだにできておらずその点が今後の課題である。

(※文責：柳川大輝)

4.2. 野中

前期

まずアプリケーションの機能についての話し合いを行い、2.2で挙げた計6つの機能をつけることを全体の目標とした。その後開発環境の選定を行い、Unityを開発環境として決定し実際の開発を行った。アプリケーションの中で私は機能選択を行えるホーム画面の作成をした。ホーム画面上には各機能の切り替えを行う切り替えボタン、ゴミ箱の容量が見えるゴミ箱メーターそしてゴミ箱の呼び出しを行える呼び出しボタンの実装を目標としていた。

ホーム画面作成にあたってまず最初にレイアウトを決め、適正なボタンの大きさや配置を決めた。そしてレイアウトを決めた後に3つの機能の内の切り替えボタン作成から取り掛かった。切り替えボタンの実装方法として以下の2つの方法が考えられた。

- ・ ボタンを押すことでオブジェクトの表示、非表示を切り替えてひとつのシーンにまとめる
- ・ シーンを切り替える方法を用いてそれぞれの機能を1つのシーンとして分ける

実装にあたって前者の方法は各個人がバラバラに作業する方法を取っていたため統合の際に時間が掛かり、変更する点も多く不具合の発生する可能性が高くなることが予想された。その

Challenge to Complex Intelligent Robot

ため切り替えボタンは後者の作った機能をそれぞれ1つのシーンとして区切り、Unityのシーン移動の機能を使い、それらを切り替えることで切り替えボタンを実装することとした。切り替えボタンと実際に動かすためのコードを作成し、他の機能が完成したのちにその画面にも実装が可能な状態にすることで切り替えボタンは完成した。

切り替えボタンの作成が終わった後には呼び出しボタン、及びゴミ箱メーターの作成に取り掛かろうとしたが、ロボットからの情報送信が行えなかったので機能事態の実装は行えなかった。その結果、見た目上でのボタンやメーターの配置にとどまった。

この作業が完了したころに他の機能が仕上がって来たので実際に切り替えボタンが作動するのかを機能を統合、ボタンの追加を行いテストした。テストの結果切り替えボタンは正常に機能することが確認できた。しかしながら統合した結果それぞれ作っていた規格が異なっていたためにボタンの配置や文字の大きさ等に大きな差異が発生し、ボタンが押せないという状況が発生した。その事態を踏まえて規格を統一し今後起こらないようにし、ボタンの再配置を行い無事に統合作業を完了しました。そのため、後期では前期で実装が出来ていない機能の開発を行っていくことを目標とした。

後期

後期の活動はまず最初にロボット班と呼び出し方法の模索を行うことになった。その中でロボットに何かスマホから目印となるものを送り、そこにロボットが近寄っていくという方法が提案された。そして最終的にライントレーサーの技術を利用した方法と呼び出す方法として使うことが決定した。この方法はライントレーサーが線を検知するように、スマホのLEDライトを用いて目印となる点を出し、そのライトをロボットが感知してその場所に向かって動くようにする制御を行うといった手法である。つまり、ロボットの正面カメラにスマホの背面LEDライトの光を検知してもらい、その光の点を中心とするように移動をしてもらう。そうすることでロボットはライトの光っているほうへ、つまり呼び出している場所へと誘導することができるといった方法である。

これによって課題解決のプロセスの1段階目の誘導方法の模索はクリアされた。

次にこの呼び出し方法を実現させるためにアプリケーションで行う機能の模索、検証を行った。ロボットを呼び出すためにスマホLEDライトを点灯させる必要があったのでLEDライトをアプリケーション上のボタンで点灯、消灯を行えるような機能の実装を行うこととなった。そのためライト点灯機能の実装方法を調査した。

まずUnityの機能で実装ができないかということを検討した。そのためにUnityの参考資料を探し、調べることにした。その結果としてUnityにはスマホの背面LEDライトを制御する方法を見つけることはできなかった。しかし、LEDを直接制御する方法を見つけることはできなかったがスマホのカメラを制御する方法を見つけることができたのでこの機能を利用してスマホ背面LEDを制御できないかと検証を行うことにした。カメラを使うことで撮影や周囲の様子を映すこ

Challenge to Complex Intelligent Robot

とができるのでそれらの機能を起動させることで間接的に背面LEDを制御することができないかと考えた。試行錯誤をした結果カメラの制御を行うことができる機能を実装することができた。しかしカメラの操作をすることで間接的に背面LEDライトを制御する方法を発見することはできなかった。

そのため新たな手法を模索することとなり、その結果以下の2つの手法を検討することとした。

- Xcode 上でLED制御を行う機能を作成してUnityに実装する
- Android Studio上でLED制御を行う機能を制作してUnityに実装する

どちらの手法を用いて開発を行うべきなのかを検証し、その結果Xcodeには以下の2つの問題点が上がった。

- XcodeはMacOSでしか扱うことができず開発を行うのに手間が大きくなる
- LEDライトの制御を行う機能制作にあたってテストを行う実機がなく、またエミュレータ上でも機能のテストを行えない

一方Android StudioではどのOSでも扱うことが可能なため開発環境の構築が簡単であり、またエミュレータ上ではXcode同様に動かすことは出来なかったが実機を用いることが可能であったためテストを行うことができた。以上の点からXcodeでの開発はもう一方のAndroid Studioを用いる方法に比べて手間が掛かるため開発にはAndroid Studioを用いての手法で行うことにした。Android Studioを用いてLED制御を行う詳しい手順としては以下の通りとなる。

- 1、Android StudioでLEDのオン・オフの切り替えが行えるアプリケーションを作成する
- 2、作ったアプリケーションをaarファイルにエクスポートし、Unityに実装できる形にする
- 3、Unityにaarファイルをアタッチし、アプリケーションに使用しているクラスを使用してLEDを制御する

まず最初の手順としてAndroid StudioでLEDの制御を行うアプリケーションの作成を行った。その為に参考となるコードを探し、その結果懐中電灯アプリのコードを参考として作成にあたった。しかし懐中電灯アプリに使用されていたバージョンが現在のバージョンとは大きく異なっていたため、書き方の参考にしながらAndroid Studioのリファレンスを確認し、変更されていた点を修正してアプリケーションの作成を行った。ソースコードの少なさはAndroidにLEDライトの点灯機能が標準で付いたためだと考えられる。結果、簡単なLEDライトのオン、オフが可能なアプリケーションを作成することができた。

Challenge to Complex Intelligent Robot

アプリケーションの作成が出来たため、次にUnityに実装できるaarファイルへエクスポートを行い、Unityへの実装作業へ取りかかった。しかし、作成したアプリケーションを実装しようとした際に実装が行えないという問題が発生した。その問題が起こった原因として以下の点が考えられる。

- Unityと作ったアプリケーションで動作可能なバージョンが異なる
- aarファイルを上手くエクスポート出来ていない
- Unityのコードに問題があり動かない

しかしながら原因が絞り込めなかったため1つずつ修正を行い、実装ができないか試行錯誤をしている状況である。これを今後の目標としていきたいと考える。

(※文責：野中和馬)

4.3. 佐藤

- 前期

ゴミ箱を動かすために、プロジェクト班全体でアプリが必要と考えたが、それには何で開発していくのが妥当かとグループ内で話しあった。その結果、次の点で優れていると考えたためUnityで開発していくことになった。まず、今回のロボットに使用していく予定であったArduinoとの連携が可能である点、高い拡張性がある点、これを使ったアプリをよく見かけるため、アプリが作りやすい可能性があるという点の3つだ。

次に、ゴミ出しのアプリに何が必要になってくるかをグループ内で意見を出し合い、カレンダー機能が必要などといった意見が出た。そこで、アプリのホーム画面作成班とカレンダー班に分かれた。カレンダーは函館のゴミ出しカレンダーを使うことにした。しかし、函館のゴミ出しカレンダーは複雑になっていた。具体的にいうと以下のような点があった。

- ごみの収集日が毎週あるものもあったり、隔週のものもあった。
- そのパターンが一部適用できない週があった。(年末年始)
- 年度末に切り替わる、つまり新年度になるとパターンが変更される可能性がある。
- その収集日のカレンダーのパターンが地域によって違い、16パターン存在した。
- また、地域といっても「〇〇町1丁目28～45」のような同じ町名であっても番地等で収集日パターンが違うところも存在した。
- それに加えて、町名、番地以外で分けていた地域も存在していた。

Challenge to Complex Intelligent Robot

これらを考慮してカレンダー機能を作成するために、ハッシュマップが必要になった。これは2年次までに習っていた内容にあったので、Unityで使うプログラムに使う言語のC#にも対応できるかを調べて、できるようだったので使用した。地域ごとに選択するカレンダーパターンをキーに設定し、地域別に振り分けていった。この後、地域選択画面の担当者、カレンダー画面作成の担当者と連携させるために文字の半角全角の修正など多少の修正をした。

・後期

前期の中間発表での評価シートなどを見て全体で反省した結果、プロジェクトの進め方や内容にいろいろと穴が存在していたため、ソフト班はロボットの呼び出し方法の変更やシミュレーターの作成などアプリの機能の変更だけでなく、アプリ以外のものも作成していく必要になった。シミュレーターの作成が後期で湧いて出た理由は、シミュレータはロボットの最適な行動を模索するのに必要になってくるためである。つまり、ロボットの動きをシミュレーター内で再現していくことで、ロボットの移動の課題が解決されると考えられた。

そのシミュレータを作っていくにあたり、ソフト班内のシミュレータ班で話し合った結果、何も作れないよりは簡単な法から段階を踏んでいき最終的に現在のロボットと似たようなロボットが画面上で動くシミュレータを作っていくという方針になった。具体的に言うと、まずタイヤなしのロボット、次に2輪走行のロボット、最終的にロボット班が使用している3輪走行のロボットが画面上で動くようにしていくといったものだ。このシミュレータを作成するに必要な開発環境は、アプリ開発の時に使用していたUnityが妥当と考えた。そして、個人で開発していくことでそれぞれがプログラム言語の勉強がしやすいなどといった理由で、別々に開発していき、相談しあい作成していくことになった。

まず、私はUnityの実行の方法や、画面の作成方法を知らなかったため、後期の授業で使おうとしていたUnityの教科書や、インターネットを駆使して、Unityの中での図形の設置の仕方などを自主的に学んでいった。まず、私は直方体を駆使して、部屋の大枠を作成していった。次に、実際のロボットは障害物を判定して、動きが変わっていくため、直方体の障害物だけでなくリアルな環境を再現していくことで、実際の環境になるべく近づけていこうと考えた。そこで、私はインターネットで調べて、Unity内のアセットストアで無料の3Dの人型のサンプル、家具のサンプル（テーブル、椅子）が存在したため、それらもシミュレータ内に設置した。

部屋が完成した後、シミュレータ上のゴミ箱ロボット作成に取り掛かった。ゴミ箱ロボットは前述のとおり言語はC#で書いた。ロボットに以下のような機能が必要だと考え順番に作成していった。

- ・ゴミ箱ロボットを前に動かす
- ・ゴミ箱ロボットに視点のカメラの設置
- ・ゴミ箱ロボットの回転
- ・ランダム走行

Challenge to Complex Intelligent Robot

- ・ 障害物判定
- ・ 視点カメラによる目標の検知
- ・ 目標への追尾
- ・ 複数台の操作

最初に、ゴミ箱を動かそうとした。「X座標を2秒で+3動かす」といったようにC#にもともとあるのをそのまま使おうとすると現実離れした動き方になってしまう。これを、どうしようかと模索した。その結果、インターネットで調べ、動画なども見たところ、適切なアセット、ライブラリがあった。等速ではなく、最初は徐々にスピードを上げ、目的地に近づいたら徐々にスピードを落とし、指定した秒数で目的地まで移動するものであった。これを実装してみたところ、自然な動きとなった。

次に、ロボット班に聞いて、実装していることが分かったランダム走行である。まず、違う方向にカメラと一緒に動かす、自然な動きに近づける動作が必要だった。これらは前述のアセットが活躍した。そのアセットは、自然な移動をC#のプログラミング上で実装させる、だけでなく、ほかにも回転などの機能を持っていた。そして私はその機能の中にあつた、回転、1つの動作が終了したら呼び出される文、前述の移動の機能により違う動きを再現できるようになった。ちなみに、カメラの動きについてはゴミ箱と関連付けさせる等をして解決に至った。ランダム化は、そもそもC#にあるランダム関数を用いてようやくランダム走行が実装ができた。

次に障害物判定の実装を目指した。実際のロボットは超音波センサを用いて、前方に物体があるかの判定をして、あつたら引き返すように設定をしている。しかし、シミュレータ上にはセンサから読み取って動くという動作は再現不可能であつた。センサーとなるものが存在しないためである。その後、考えたところ、最初は障害物を検知して避けるAIを使い、再現しようとした。しかし、自身の経験不足のため、仕組みを理解することができず、使用することができず、途中であきらめ、別な方法を模索した。

次に考えたのが、物体の当たり判定を利用したものであつた。Unityには当たり判定を認識する範囲の指定ができるためそれを利用して、判定をしようとした。これは互いにその範囲に余裕を持たせて直接ぶつからないようにした。インターネットで調べたところ、こちらのほうは、範囲を少し調整し、プログラミングでそれを用いて当たり判定を行わせる方法が存在した。それを行ったところ、しっかりと障害物として設置したものや壁をしっかりと判定して、ぶつからず引き返すようになった。しかし、一部の障害物、例えば低い位置にある障害物の当たり判定を行った際、ゴミ箱が跳ね返って、宙に舞って元の位置に戻るという不具合が発見された。これに関しては、原因が不明なため修正するまでには至ることができなかつた。また、障害物には円柱状のものや、自分より背が低い障害物の判定、動く障害物に対しての判定ができていないため、今後その判定も必要になってくる。

次に目標、つまり現実のロボット操作でいうライトの検知、追尾だ。こちらはライト代わりに、球体を用いた。シミュレーション上でとある球体を見たらそれを正面にして前進し、追尾していこうと考えた。まず、カメラでの球体の認識を行う。インターネットで調べたところ、

Challenge to Complex Intelligent Robot

カメラでの特定の物体の検知が可能であったため、その関数や方法を取り入れてみたが、障害物の向こうでもそれが認知できてしまったのである。調べたところ、カメラの視野の範囲に入ってさえいけば、間に障害物が入っていて目標が見えなくとも、見える、検知してしまう判定になったのだ。このことをどうにかして、カメラに写っていれば検知という風にしていけないだろうか、と調べた。すると、カメラと目標の間に障害物が入ってしまい、目標が見えなくなったら、障害物を透明化するというプログラミングの見本があったのでそれを応用していこうと考えた。これには、もし目標とカメラの間に障害物が入っていたらこうするといったようなプログラムソースがあったのでそれを利用していった。今回のシミュレータ上では、障害物がカメラと目標の間に入ってしまい、目標が見えなくなったら検知しないという風に変えればいいので、そういった風になるようにソースコードを書き直した。そして、わかりやすいように文字でしっかり判定しているかを見たところ、障害物の向こうの判定がしっかりと出ていたためこれで障害物判定の部分は完成したと考えられる。

次に障害物を、カメラの中心にとらえつつ近づくというものだ。これはどうしようもできないので互いの位置と三角関数を用いて、回転させる角度を計算し、回転途中であった場合そこでストップさせてそのまま前進させていこうと考えていた。これに関しては、カメラの中心にとらえ、回転をストップさせることには成功したが、追尾していこうというところで不具合を起こしそのままロボットは動かなかった。そのまま、追尾する部分に関しては不具合の修正や解決方法を模索していたが、本プロジェクトの期間中には実装までには間に合わなかった。

最後に、追尾する部分以外を搭載したロボットを複数台動かしてみても、互いの動きに支障が無いかを試すために、小さな部屋を用意し、障害物を多少配置し、その中で2台動かしてみることとした。ここでは、障害物が目標であるかどうかをタグを用いていたため互いのロボットにも障害物と同じタグをつけた。また、カメラの判定がおかしくならないように、それぞれのカメラで判定したものをそれぞれのゴミ箱で反応するように、関数名等を変更した。そして動かしてみたところ、互いのゴミ箱の障害物判定、目標判定は問題なく行われていた。しかし、シミュレータ内で互いのゴミ箱が接近することがなく、ゴミ箱同士の判定がどう行われていくのかを期間中に見ることができなかった。今後、確かめていく必要がある。

最後の過程を除いた物を、ロボット班が以前、撮影した物などを見て見比べて、かけ離れていないかの確認も行った。障害物判定は以前撮影された実際のロボットの障害物判定の動画と見比べた。カメラによる目標の認知、ランダム走行の分野に関しては、実際に目の前で動いているものを見て、比較していった。大きくは変わってはいないので大丈夫だと判断した。しかし、ゴミ箱ロボットが宙に舞う不具合の修正、追尾機能、ゴミ箱同士の障害物判定の確認が、本期間中にできなかった。

(※文責：佐藤佑哉)

4.4. 石崎

・前期

前期の活動では、ゴミ箱ロボットを作るということが決まったため、ソフト班でどのような機能を作成するかを話し合い決めた。その中で出てきた案が以下のものである。

- ・ゴミ箱を呼び出す機能
- ・ゴミ箱に入っているゴミの量を表示するメーター
- ・その日に捨てるゴミの種類を表示
- ・ゴミ捨てのカレンダー
- ・上記のカレンダーを表示する地域を選択画面
- ・1日にどれだけゴミが溜まったかを示すグラフ

ソフト班では以上の機能を一つのアプリケーションとして制作することとした。これらはゴミ箱の稼働環境を学校だけでなく、将来も見据えて一般家庭等も対応できるようにしたため、カレンダーのような一般家庭向きの機能を作ることとなった。まだメカ班の進捗もなかったため、この中の直接的にロボットとの連携を必要としない機能を優先的に制作することとした。私はカレンダーを表示する地域を設定できる画面を作ることとなった。

前段階のプロセスで決めた機能を作る上で、開発環境としてUnityを用いることとした。その他の案としてXcodeやAndroid Studio、Pythonなども上がっていたがUnityではiOSとAndroid OSの両方に対応でき、固有のUIもありアプリの作り込みはやりやすいと判断した。また、Unityを使う上でC#を使うことになるがこれはロボットの制御には比較的向いている言語で、Unityにはarduinoのライブラリがあるためロボットの制御とアプリ開発の両方をこなすのに適していると考えた。アプリケーションの開発に入る前にグループメンバーでUnityを扱ったことのある人がいなかったため、借りてきた本でUnityの基本的な使い方を学ぶこととした。

前段階のプロセスでUnityの基本的な使い方を学んだので、本プロセスでは事前に決めた担当の機能を作ることとした。私の担当する地域選択の画面はカレンダーの担当と話し合い函館市内に限定して作ることとした。基本的な構造としてはホームから地域設定の画面を押し、設定画面に飛ぶようにする。そこから自分で住んでいる地域を選択しそれをアプリ内に反映できるようにし、カレンダーやその日捨てるゴミの種類を表示する機能で使えるようにするというものである。その中で設定画面から制作し始めた。デフォルトでは選択されていないと表示するようにし、その下の選択ボタンで選択画面に遷移するようにした。遷移先では地域をただ羅列するだけだと探しづらいので50音の行ごとに分け、それぞれのボタンを押すとその行に対応した地域名が出てくるようにし、スクロールして探し選択できるようにした。また、間違えて押したときのための戻るボタンは画面の一番下に配置した。ここで選択した地域名のログをとり、最初の設定画面に遷移させ、とったログの地域名を選択ボタンの上に表示するようにした。

Challenge to Complex Intelligent Robot

私の作った地域選択の画面とカレンダー、またそれらを合わせるためのハッシュマップを結合することとした。まず、先に完成していたハッシュマップと地域選択画面を結合することとした。カレンダーと合わせる前の確認として地域を選択するとその地域と対応した函館市のゴミ捨てカレンダーの番号を出力するようにした。半角と全角や漢字の間違いやタイプミスなどがあるとうまく反映されなくなってしまうので二人掛かりで入念にチェックした。ここでハッシュマップとの連携ができたので、ハッシュマップ側で出力したカレンダー番号と対応するよう色のついたカレンダーを表示できる仕組みを作ることとした。出来上がった無地のカレンダーを地域が設定されていないときのデフォルトのものとして、選択された時にハッシュマップ側で出力された番号をカレンダー側で受け取るようにする。その番号に合わせて日付に色をつけるようにした。年末年始は不規則になるため、試験的に2018年4月からの一年分のみのカレンダーとした。

ここで出来上がった機能をアプリ本体の外枠とくっつけ、一つのアプリケーションとした。製作時のアスペクト比の違いで文字のサイズやボタンの位置などにずれが生じたが、ホームにあるボタンからカレンダーや設定の画面への遷移は問題なくできた。

・後期

後期の活動として、ロボットの機能である移動と呼び出しの課題を解決するために、前期までの活動とは別にロボットのシミュレータを制作することとした。このシミュレータは実際のロボットにいきなりプログラムを組み込み動かすのが大変であり、問題が起きる可能性があるため、実験できる場としてロボットの最適な行動を模索するという役割や現状では現実的にできない複数同時稼働のようなものを仮想空間で実現させ将来像を完成させるという意図で制作することとした。まず、メカ班と話し合い、実際のロボットの動きとして以下の行動が必要であるということがわかった。

- ・ランダム走行
- ・障害物の検知と回避
- ・対象の検知と接近
- ・対象の近くで停止

これらを順序立てて組み立てた場合、途中で何らかの障害が生じた時に最後まで実行できなくなるため、これらを並列的に組み立てた階層構造であるサブサンクションアーキテクチャのアルゴリズムを適用することとした。

シミュレータの制作にあたって前期のアプリケーション制作で使っていて新たに勉強する必要のなかったUnityを使用することとした。初めは上記で挙げたアルゴリズムの動きを確認するためにタイヤをつけずに簡易的な形でシミュレータを制作することとした。まず、仮想の部屋としてシンプルな長方形の部屋とし、テーブルや椅子など一般的な部屋にあるものを障害物として配置した。タイヤをつけていないためサブサンクションアーキテクチャの第一段階であるランダム走行に関しては、直進とその場での旋回をランダムで呼び出すことで容易に実装する

ことができる。また、第二段階である障害物の検知に関しては実際のロボットと違い、超音波センサが無いが、指定した位置から指定したベクトルの方向に、自ら指定した長さの光線を飛ばせるRaycastを用いてロボットの正面方向に光線を飛ばし、その光線が障害物に触れたら右または左に約90度回転するようにし、実装することが可能である。対象の検知はカメラを仮想ロボットに搭載して、カメラに対象物が映っているかどうかの判定を行うことで可能とした。次の段階である対象への接近は、カメラの中心に対象が来るように機体を制御するライントレースの技術を適用し、その制御方法として対象物とカメラの中心のずれを徐々に減らしていくPID制御を用いて行うとしたがこの時点ではタイヤをつけていないため実際の制御とは異なり、対象物の座標を検出してそこに向かって直進するような仕組みになった。現段階では実物のシミュレータとしてはかけ離れてしまっているため次のプロセスに移ることとした。

前段階のプロセスで基本的な動きや大まかなアルゴリズムへの理解が深まったため、このプロセスでは再現率を高め、より実際の動きに近いシミュレータを制作することとした。前段階のプロセスから、対象物の検知と接近においてタイヤを取り付けることでPID制御が可能になると考えられた。今回のプロジェクトでは三輪駆動のロボットを使用することとなっているが、本プロセスでは次の段階として二輪駆動のロボットを制作することとした。ロボットの稼働空間は前段階のプロセスで作成した部屋を使用することとした。ロボットの構造はゴミ箱となる本体とその両脇にタイヤを一つずつ取り付けた。また、このままだと機体が安定しないため本体の底にキャスターを取り付けた。それぞれのパーツに重さを与えることで一つのロボットとして機体を安定させることができた。サブサンクションアーキテクチャの第一段階であるランダム走行は前段階のプロセスと同様に直進とその場での旋回を一定時間ごとに呼び出すことで可能とした。第二段階である障害物の検知と回避も前段階のプロセスと同様にRaycastを用いて障害物を検知し、右または左に回転させることで回避できるようにした。このRaycastを用いることで超音波センサの搭載できないUnity上での擬似的な障害物検知を可能とした。また、本体にカメラを取り付けることで対象物が映っているかを判定し対象の検知ができるようにした。この時、Unityのカメラでは、対象が物陰にあって視覚的には映っていないくても、カメラの範囲内にあれば映っているという判定になるため、障害物検知の時に用いたRaycastを向きをカメラの正面方向として本体と対象の間に飛ばし、その間に何らかの障害物を検知したらランダム走行に切り替えるようにして実際の挙動の同じく文字通り映っている時のみ反応するようにした。対象への接近に関してはライントレースを行うため、PID制御を用いて機体を制御させる。まず、対象物とカメラの中心のずれを求めなければいけないがカメラから読み取ることができない。そこで、カメラの正面方向のベクトルと本体から対象物へ向かうベクトルの内積を求め、そこから2つのベクトル間の角度差 θ を求める。それに加え、本体と対象物の距離 l を求める。求めた距離 l と求めた角度 θ から三角関数を用いて以下のようにずれを求める。

$$\cdot \text{ずれ} = L \sin \theta$$

ずれを求める際、対象物がカメラの右にあるのか左にあるのかによって制御するタイヤが変わってくる。この時点では角度差 θ は正の値にしかならないため、左右の判定ができない。そこで、Vector3.Crossを使ってカメラから見て対象物が左側にあると0度～-180度で出力、右側にあると0度～180度で出力するようにし、この角度が0度より小さくなっていた場合、ずれの値を

Challenge to Complex Intelligent Robot

負の数で出力できるようにした。まずは対象に向かって機体を進ませる仕組みを作らなかったため、今回はPID制御の中の比例を用いるP制御を実装することとした。比例ゲイン k_p とこれらの値を使い、タイヤの制御量を以下のように表した。

- ・タイヤの制御量= k_p *ずれ

この制御量を左タイヤの速度から減算、右タイヤの速度に加算することで機体を左右にずらしながら対象物へ向かっていけるようになる。この時の k_p の値でしっかり対象物を追従できるかが決まるため、少しずつ値をずらし調整した。 k_p を10以下に設定すると、しっかり追従するようになったがシミュレータのカメラでは左右に機体をずらしながら制御しているように見えなかったため、少し値を大きくし13とした。この値にすると機体の進む速度が少し遅くなってしまいが、機体が左右にブレながら動いていることが目視で確認できたため、P制御はできていると判断した。これで機体の機体の制御ができたので、制御中に取り付けたカメラから対象物が見えないと判定されたらランダム走行に切り替えるようにした。障害物にぶつからず対象物の近くにきたらタイヤの速度を0にして機体を止めるようにした。

二輪駆動での機体の制御ができたため、次のプロセスとして実際の稼働を想定して複数台の同時稼働を実装することとした。実際の使用場面としては、ゴミの分別が必要となってくるため、複数のロボットが同時に動いていないといけない。しかし、実物を複数用意するのは現実的ではないため、シミュレータの仮想空間で実装することとした。基本的にはこれまでのプロセスで作ったロボットをプレハブ化して増殖するようにした。また、呼び出しの際に全てのロボットが同じもので呼び出されてしまったら困るため対象物の色を適宜変えて、それぞれのロボットを個別に呼び出せるようにした。画面のサイズの関係上、全体を写すカメラの他にそれぞれのロボットに取り付けたカメラによるロボット視点のものを用意したため、配置場所の関係でロボットの数自体は2つほどで止めることとしたが、同じ作業を繰り返すことでもっと個数を増やすことは可能であると考えられる。

(※文責：石崎智哉)

5. 結果

5.1. 成果

- ・前期

ソフト班はアプリの機能の一部である地域別のゴミ出しカレンダー、ごみのメーターの外枠、地域選択画面機能を作成した。

・後期

ソフト班は呼び出し機能の一部実装と、車輪なしと、2輪走行のゴミ箱ロボットシミュレータの開発を行った。

(※文責：佐藤佑哉)

5.2. 解決手順と評価

5.2.1. 前期

プロトタイプの完成、試用までには至っておらず、アプリの一部の機能しか実装できていないため評価はまだ不可能である。今後、プロトタイプができた際、ソフト開発にかかわっていないロボット開発班によって試用していき評価を記述していく予定である。

- ・ごみを捨てる活動の促進のために作成しているアプリの利用率の向上のための機能の一つとして、ごみを出す日は見るのが必要と考えられるごみの地域別カレンダー、地域選択画面、それらを連携するためにハッシュマップを内部に実装した。今回は、函館市内で使うことを想定し、函館市の地域に配布されているごみ収集カレンダーを参考に2018年度分を作成した。

- ・ごみ捨て活動の促進させるための機能、アプリの利用率の向上のための機能の一つとなっている、ごみのメーターを搭載し、わかりやすい図に示すことにより、ゴミ箱ロボットの現在の容量を知れるようにしていく。ロボットとの連携は未着手のため、今後連携していき、評価していく予定である。

- ・ロボットを動かし、ごみを捨てる活動を促進させるために必要な機能の一つである、ロボットの呼び出し、戻るボタンの実装をしていく。現在、メーターと同様にロボットとの連携が取れていないため外枠のみが完成しており、後期に評価していく予定である。

- ・後期にごみを捨てる活動やアプリの利用率の向上のための機能の一つである、ロボットと連動し毎日更新されるごみの量のグラフ、今日出すごみの種類の表示をする機能を作成していく予定である。

(※文責：佐藤佑哉)

5.2.2. 後期

後期は今では必要でないと思われる一部のアプリ機能の開発をやめ、シミュレータと呼び出し機能の作成の2つの点について重点的に開発を行った。

その結果

- ・シミュレータは実際のロボットのように3輪走行のシミュレータを作成するまでには至らなかった。しかし、車輪なし、2輪走行のシミュレータが完成に近い状態までもっていくことができた。この2つのシミュレータは実際のロボットと同じような反応に近づいて行った。完成形まで持っていくことができたら、実際のロボットと比較しロボットの動きの改善などがしやすくなったかなどを評価していく予定であったが間に合わなかった。

- ・呼び出し機能に関しては、呼び出すための方策までまとまり、そのために呼び出し機能を開発していたが、途中までしか進めず、実際のロボットとの連携するかどうかまでの実証、評価はできなかった。

(※文責：佐藤佑哉)

6. まとめ

6.1. プロジェクトの成果

前期期間において、ソフト班ではアプリ内の構成とデザインについて決定しアプリ内の以下の4点を制作した。

1. ゴミ箱の呼び出しボタン
2. ゴミの量を示すメーター
3. 函館市内における町名などを設定する地域設定画面
4. ゴミの分別を表示させるゴミ捨てカレンダー

Challenge to Complex Intelligent Robot

後期期間において、ソフト班では以下の2点を制作した。

1. ゴミ箱ロボットのLEDライトを用いた呼び出し機能
2. ゴミ箱ロボットのシミュレーター

(※文責：柳川大輝)

6.2. プロジェクトにおける各人の役割

前期

- 石崎 アプリ内の地域設定画面制作
佐藤 アプリ内の機能結合のためのハッシュマップ制作
下田 アプリ内のカレンダー機能の開発
野中 アプリ内のホーム画面設定および機能の結合
柳川 アプリ内のイラスト作成およびグループ週報の作成

後期

- 石崎 シミュレーターの制作
佐藤 シミュレーターの制作
野中 ゴミ箱ロボットのLEDを用いた呼び出し機能
柳川 ゴミ箱ロボットのLEDを用いた呼び出し機能

(※文責：柳川大輝)

6.3. 今後の課題

6.3.1. アプリケーション

Android Studio上で開発したLEDライトを制御するアプリケーションをUnityに実装できるaarファイルへエクスポートを行い、Unityへの実装作業へ取りかかったが、作成したアプリケーションを実装しようとした際に実装が行えないという問題が発生した問題点としては、

Challenge to Complex Intelligent Robot

Unityと作ったアプリケーションで動作可能なバージョンが異なる、aarファイルを上手くエクスポート出来ていない、Unityのコードに問題があり動かないの3点が可能性として上げられるので問題点を発見し、アプリケーション内でのLEDライトの制御を可能にしたい。

さらに、呼び出し機能においては将来的に燃えるゴミ、燃えないゴミ、ビン、カン、プラスチック、ペットボトルの様に複数のゴミ箱ロボットの使用を目標としているのでアプリケーション内での呼び出しボタンを増やすために複数の色のLEDライトの点灯を可能にしたいため、スマートフォンだとLEDライトの色が一色しかないのでプロジェクト内でゴミ箱ロボット対応の専用デバイスを開発し燃えるゴミの場合の点灯方法と、燃えないゴミの点灯方法を変化させることにより呼び出すゴミ箱ロボットの種類を識別できるようにしたいと思う。

アプリ内の機能面で上げるとゴミ箱内のゴミの蓄積量の表示とゴミの蓄積量を時間や日にちななどで区別し、グラフ化を自動的に行える機能を開発したい。

ゴミ箱内の蓄積量を反映させるためにメカ班と協力し、データの送受信方法を確立させていきたい。ゴミ箱の蓄積量をグラフ化させるためにデータを自動的に計算しグラフを作成するツールの開発も行っていきたい。

(※文責：柳川大輝)

6.3.2. シミュレーター

現時点では時間との兼ね合いで二輪駆動のロボットまでしか制作に着手できていなかった。しかし、本プロジェクトで用意していたロボットの本体は三輪駆動であったため、よりシミュレータの再現度を上げ、実物に近づけるためにタイヤを一つ増やし三輪駆動でのシミュレータを制作したい。単純な二輪の構造と違い直進だけでもより複雑な制御が要求されると思われるので、三輪駆動のロボットへの理解を深めなければならない。また、PID制御に関しても現段階では比例を扱うP制御の部分しかできていないため、ずれを減らして機体をまっすぐ走らせるというところが実装できていない。そのため、求めたずれの微分値や積分値を求め、微分ゲインや積分ゲインを調整したい。そしてその結果を本体に反映して行きたいと思う。今回のシミュレータで用いた部屋は単純な構造だが、より実践的な場面でのシミュレートができるようより複雑な部屋の構造や細かい障害物の配置、人を配置して完成度を高めていきたいと思う。

(※文責：石崎智哉)

6.3.3. 発表に関する反省点

成果発表のアンケートを集計した結果ソフト班には以下のような意見があった。

発表技術について

良い点

- ・スライドは見やすかった
- ・アニメーションや動画を適切に使用していた

悪い点

- ・声が小さかった
- ・専門用語が分からなかった
- ・誤字があった

発表内容について

良い点

- ・身近な問題に取り組んでいて分かりやすい

悪い点

- ・チームとして何をしていたのかが分かりにくい
- ・システムの詳細よりどう作ってそれで何ができてどんな課題が出来たのかを説明して欲しかった
- ・問題、結果、課題が分かりにくかった
- ・呼び出し方法としてなぜスマホを使うのか？（動作でもよかったのでは）

まず発表技術についてはスライドの作成にあたっての問題はあまりなかったように感じられた。しかし誤字が残っており最終確認不足であったように感じられた。また、発表者の声が小さかったという意見が挙げられていた。これは単純な声量不足だけではなくスライドを見ながら説明を行っていたために声を出している方向によって聞き取りにくくなったためとも考えられる。また、広い空間であったことや他の発表者の声にかき消されて聞こえなかったという可能性も挙げられる。今後は声量をより大きくし、目配りを行うことで声を出す方向に気を付け、周囲の環境に応じた発表を行いたい。また、何も知らない人々にも専門用語の説明などを加えることで多くの人に理解ができるような発表を行っていきたいと考える。

次に発表の内容については着眼点は身近な問題で他の人にも分かりやすいものでよかったのではないかと考えていただきよかった半面全体的なチームとしての流れが分かりにくいという指摘があった。この指摘を受けて各個人でやったことについてだけを説明していたということを感じたのでそれだけでなく全体としてどうだったのかという総括が必要であった。また、問題提起やそれぞれの課題、そして結果について明確に区切った説明がなされていなかったためにメリハリがなかったとも感じた。内容について書いていく場合にはよりメリハリをもってここでは何を言いたいのかを明確に提示し、内容を理解しやすくしていきたいと思う。また誘導の方法についてもスマホを使わなければという固定観念があったために動作を用いるという

Challenge to Complex Intelligent Robot

方法を見落としてしまっていたので特に構想段階では考えを柔軟に働かせて今後の活動ではより良い考えを出せるようにしていきたいと思う。

(※文責：野中和馬)

参考文献

掌田津耶乃著/見てわかる unity2017c# スクリプト入門編/2018.1/

藤森将昭著/Unityではじめるおもしろプログラミング入門/2017/1

横山直隆著/やさしい[マイコンロボットの制作]/1997