

# ETロボコン2023

デベロッパー部門 アドバンストクラス

チームNo. 63 チーム名: KatLab

地区: 九州南

地域: 宮崎県宮崎市



所属: 宮崎大学 工学部

## チーム紹介、目標、意気込み

KatLabは、ソフトウェア工学の研究室に所属する学生のチームです。ETロボコンには毎年、研究室の有志メンバーで参加しています。今年は学部生4人、修士課程の学生6人で参加しています。主にモデル作成を行う設計チーム、実装や要素技術の検証を行う実装チーム、開発の効率化を目的とした生産性向上チームの3チームで開発を行っています。

今年の目標は、**CS大会の総合で3位以内に入賞**することです。KatLabは、2018年から6年連続CS大会に出場しており、年々成績は良くなっているものの目標とする総合入賞を果たせたことはありません。今年こそは、競技とモデルの両方で良い結果を出し、総合での入賞を目指します！！

## モデルの概要

※主にロボコンスナップについて記載

### 信頼性と保守性を重視したシステム

- 攻略システムとカメラシステムの2つのシステムを実装し、計算コストの高い処理はカメラシステムで実現することで、**走行体の動作に与える悪影響を抑制できる**ようになった。
- 走行体の競技動作を実現するためにCommandパターンを採用することで、**システムの拡張性が上がり、効率的な開発ができる**ようになった。
- 今年の競技固有の機能と、汎用的な機能を別のパッケージに分離し、レイヤー構造を採用することで、**来年以降も再利用しやすい**構造にした。
- ロボコンスナップ攻略ではベストショットを達成するために、**機械学習を用いて、ベストショットに最適な画像を選択**した。
- 制御パラメータの調整を支援するGUIツールを開発することで、**動作確認にかかる時間を約72%削減**した。

## モデルの構成

※主にロボコンスナップについて記載

### 1. 要求分析

目標リザルトポイント **106pt** を達成するために、主に以下の要求を導出した。

- 14秒でLAPゲートを通過し、ボーナスポイントは90pt獲得する。
- 走行体の計算コストを減らしたい。
- 来年以降も再利用しやすい構造にしたい。

### 2. 分析モデル

- 走行に関する処理を主に行う攻略システムと、画像処理に関する処理を行なうカメラシステムで、システムの範囲を定義する。
- ロボコンスナップでは、ミニフィグAは最大4か所から撮影を行い、ミニフィグBは事前準備時に指定した1か所から撮影を行う。

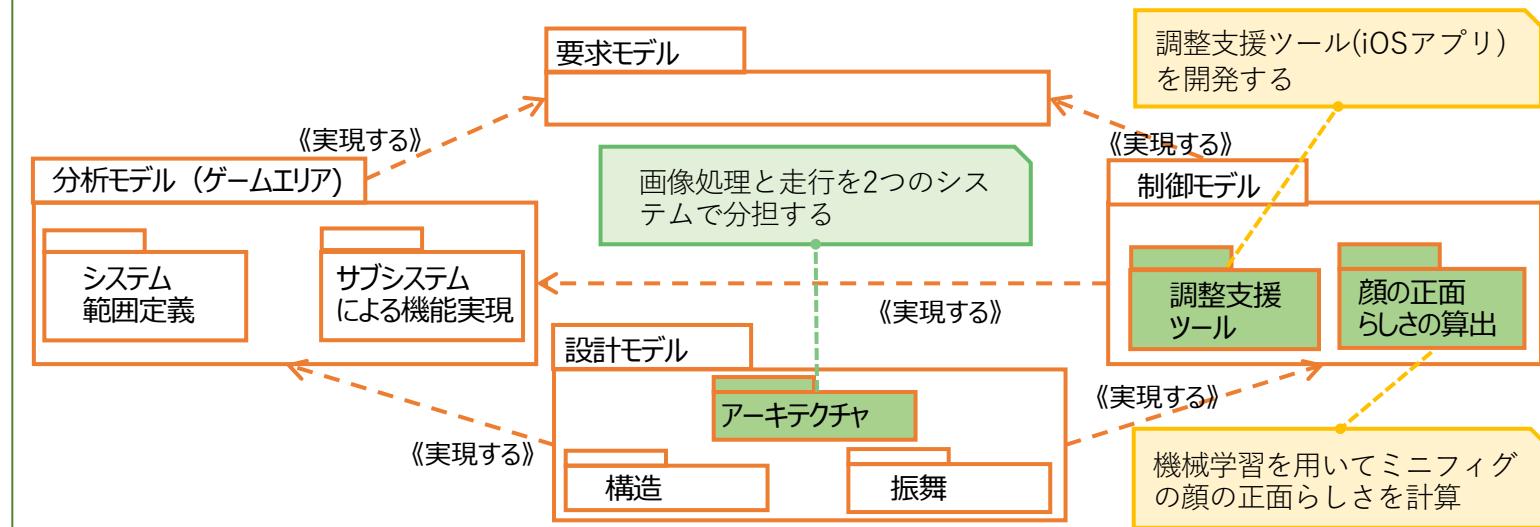
### 3. 設計モデル

- 攻略システムの走行体とカメラシステムの無線通信デバイスで処理を分担し、計算コストの高いミニフィグ画像に対する処理は、無線通信デバイスで行う。
- 今年の競技固有の機能と、汎用的な機能を別のパッケージに分離し、複雑なパッケージでは、Facadeパターンを採用することによって来年以降も再利用しやすい構造にする。
- Commandパターンを採用し、拡張しやすい構造にする。

### 4. 制御モデル

以下の事項を実現するために、制御戦略を立てた。

- ベストショット条件を満たしている画像を選択する。
- 走行体における動作調整の効率化を図る。



## 1.1 開発の目標

### 開発目標：CS大会で総合3位以内に入賞する

今年は、昨年以上の成績を残すために、CS大会で総合3位以内に入賞することを目標とする。目標を達成する上で十分な目標走行ポイント、目標ボーナスポイント、目標リザルトポイントを検討する。

### 目標走行ポイント：16pt

今年の競技ではボーナスポイントの重要性が高いと判断する。よって、LAPゲート通過を確実に行なうことを優先し、走行タイムは地区大会時の結果を向上させたタイムとして、14sを目標とする。そのため、目標走行ポイントは16ptに設定する。

### 目標ボーナスポイント：90pt

確実にCS大会で総合3位以内をとるために、すべての難所を最も高い点数で攻略することを目標とする。そのため、目標ボーナスポイントを90ptに設定する。

### 目標リザルトポイント：106pt

目標走行ポイントと目標ボーナスポイントから、目標リザルトポイントを106pt (=16pt + 90pt)に設定する。

## 1.2 ユースケース分析

開発目標を実現するためのユースケースを、図1.2-1に示す。また、各ユースケースの詳細を、表1.2-1と表1.2-2にそれぞれ示す。なお、ユースケース「LAPゲートまで走行する」については、紙面の都合上省略する。

競技を攻略するうえで、コースを4つのエリアに分ける。4つのエリアに分けたコースを、図1.2-1に示す。走行ポイントに関わる赤枠のエリアを「ラインレースエリア」、ダブルループとロボコンスナップに関わる青枠のエリアを「ロボコンスナップエリア」、IoT列車に関わる紫枠のエリアを「IoT列車エリア」、ブロックdeトレジャーに関わる緑枠のエリアを「ブロックエリア」と、それぞれ呼称する。

競技規約では、チェックポイント通過は難所とされていないが、本モデルでは難所の1つとして扱う。

ETOPCON2023攻撃システム

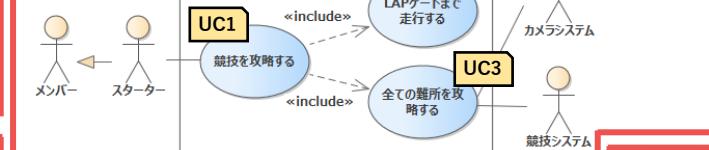


表1.2-1:「競技を攻略する」の詳細

ユースケース名	競技を攻略する
アクター	スター
目的	リザルトポイント106ptを獲得する
事前条件	システムの初期設定が終了している
トリガー	スターが走行体のスタート操作を行う
基本フロー	BF1. 走行体はLAPゲートまで走行する BF2. システムはすべての難所を攻略する BF3. システムは走行体を停止する
事後条件	LAPゲートの通過とすべての難所の攻略が完了し、走行体が停止している
例外フロー	競技開始から120秒(最大計測時間)経過する、または、走行体が走行不能になった場合、システムは走行体を停止する

表1.2-2:「全ての難所を攻略する」の詳細

ユースケース名	全ての難所を攻略する
アクター	カメラシステム
目的	ボーナスポイント90ptを獲得する
事前条件	走行体がLAPゲートを通過し、ロボコンスナップエリアに侵入している
基本フロー	BF1. ロボコンスナップエリアの難所を攻略する BF2. IoT列車エリアの難所を攻略する BF3. ブロックエリアの難所を攻略する
事後条件	全てのエリアで難所を攻略し、スタート/ゴールエリアで停止している
例外フロー	競技開始から120秒(最大計測時間)経過する、または、走行体が走行不能になった場合、システムは走行体を停止する



図1.2-1:エリア分割したコース

## 1.3 要求分析

1.2節のユースケース分析を基に導出した、開発目標を満たすための要求を、図1.3-1に示す。さらに、図1.3-1の各要求ごとに、下位要求を導出した。以降、紙面の都合から、ロボコンスナップの攻略に関する要求を中心に記載する。開発目標に関する要求図を図1.3-2に、LAPゲートまでの走行に関する要求図を図1.3-3に、ロボコンスナップに関する要求図を図1.3-4に、それぞれ示す。なお、ロボコンスナップにおける配置エリアAのミニフィグをミニフィグA、配置エリアBのミニフィグをミニフィグBと呼ぶ。

非機能要求は、ISO/IEC25010が定義する製品品質モデルの品質特性に基づいて分類した。品質特性を表1.3-1に示す。なお、セキュリティと使用性、互換性の品質特性については、今回の競技では使用しないため省略する。

また、要求図の凡例を図1.3-5に、本モデル内の関連を表す凡例を図1.3-6に、それぞれ示す。

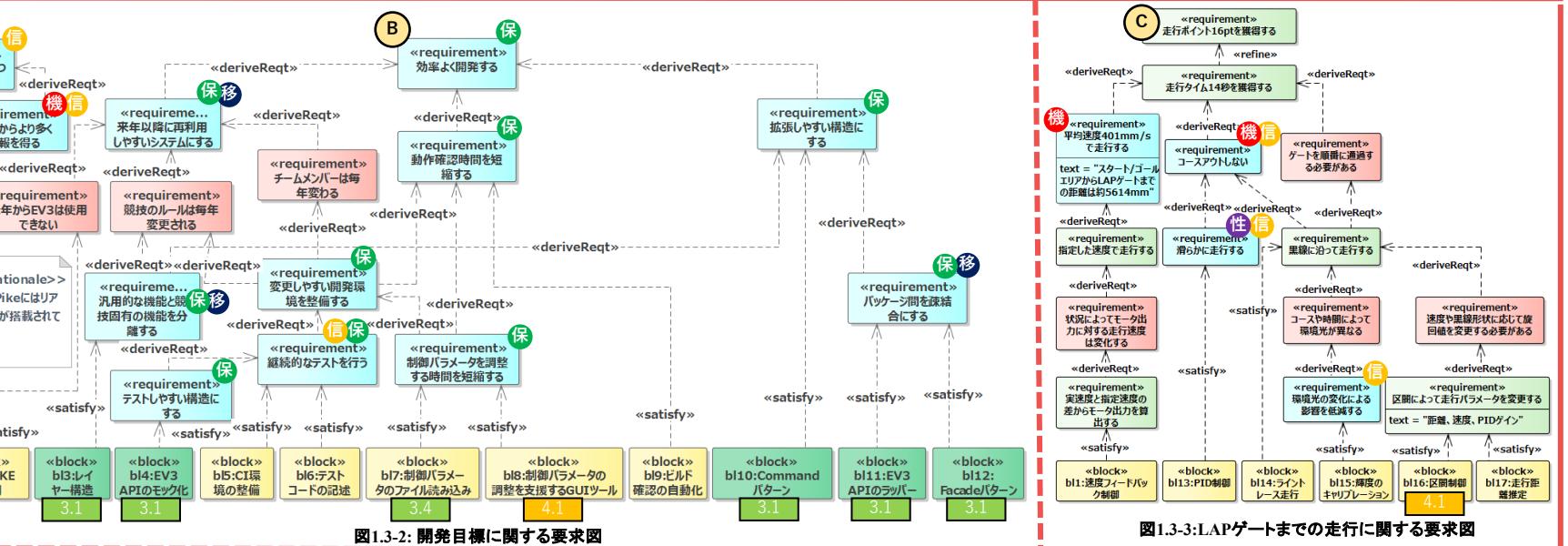


図1.3-2:開発目標に関する要求図

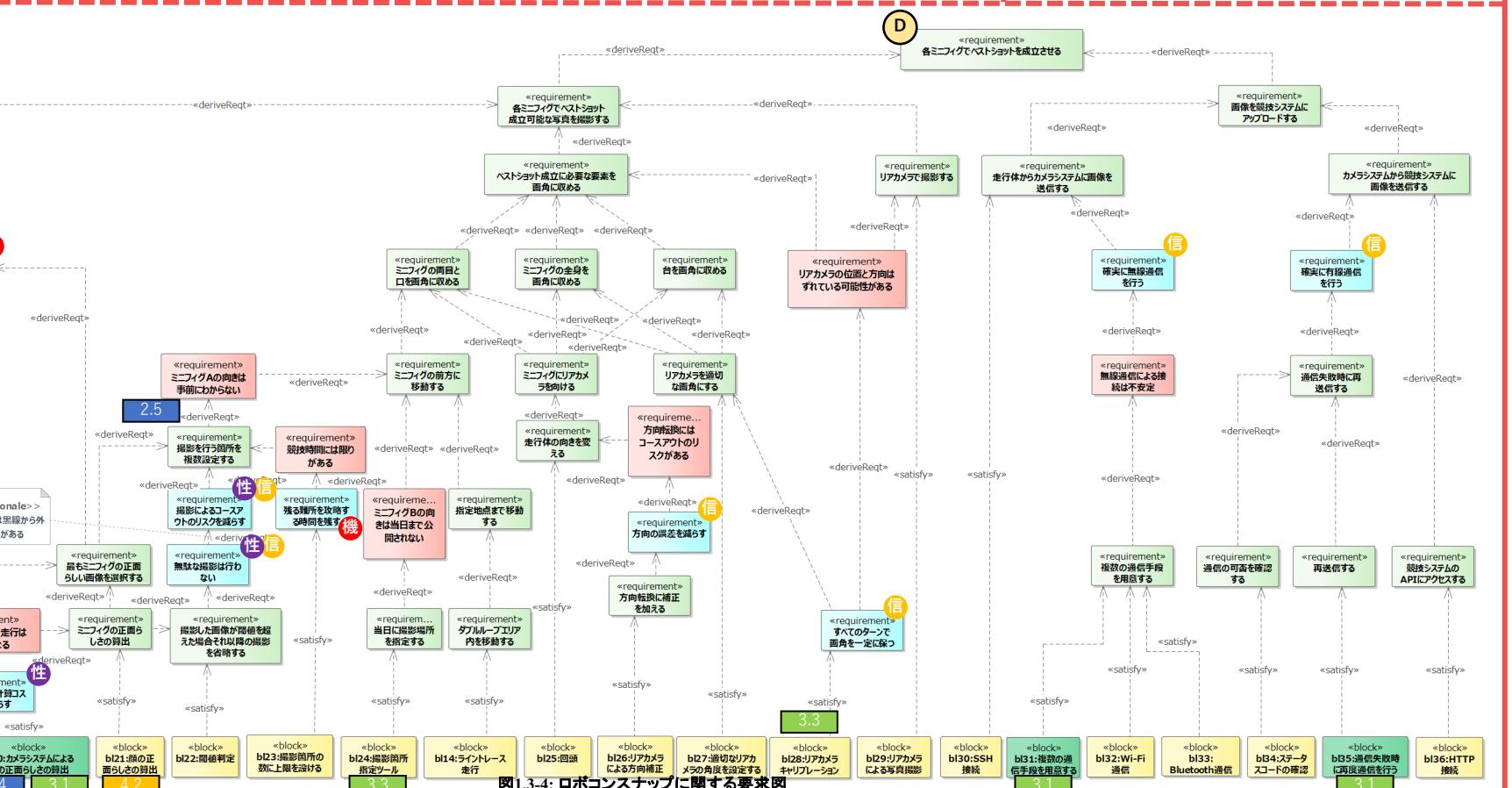


図1.3-3:LAPゲートまでの走行に関する要求図

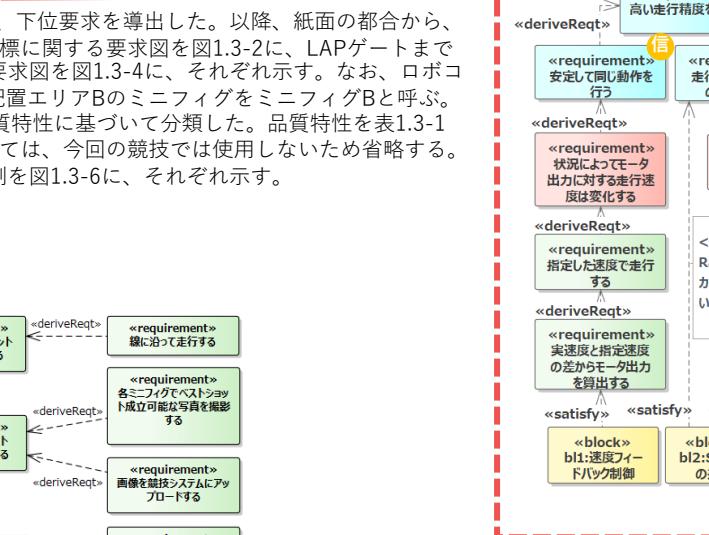


図1.3-1:ユースケースを基に導出した要求

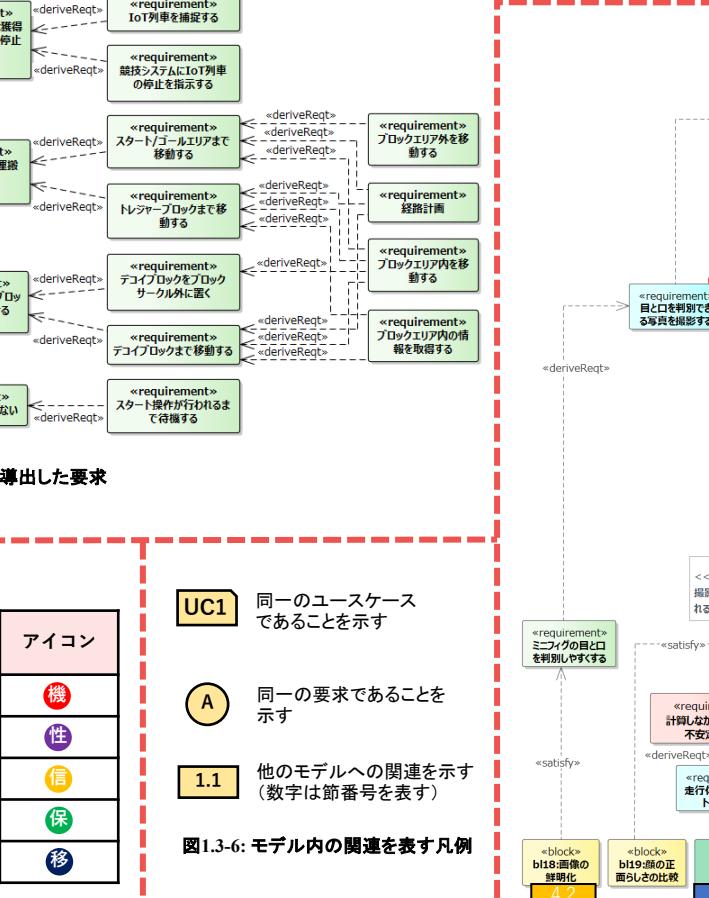


図1.3-4:ロボコンスナップに関する要求図

図1.3-6:モデル内の関連を表す凡例

図1.3-5:要求図の凡例

表1.3-1:品質特性

品質特性	概要	アイコン
機能適合性	目的に対する機能の適切さ	機
性能効率性	システムの処理能力やリソース活用の適切さ	性
信頼性	正しく機能できる度合、異常発生時の回復しやすさ	信
保守性	保守・修正のしやすさ	保
移植性	異なる環境にシステムを移す時の効率化	移

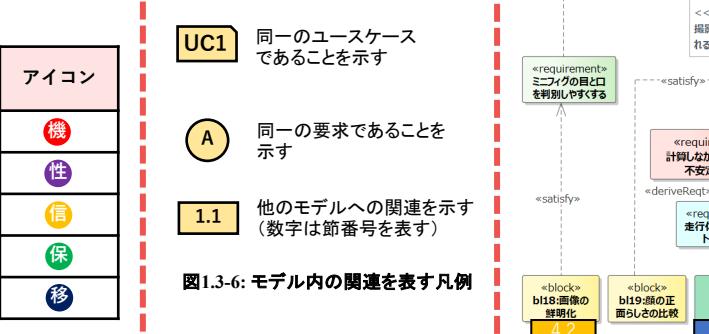


図1.3-4:ロボコンスナップに関する要求図



# 3 設計モデル（1）

KatLab

## 3.1 アーキテクチャ

1章の要求モデルで抽出したソフトウェア設計に関する要求を基に、決定した設計方針を表3.1-1に示す。また、2章の分析モデルからロボコンスナップを攻略する上で検討すべき事項を抽出し、その具体策について考えた結果を、表3.1-2に示す。なお、検討すべき事項の抽出は、図2.6-3のアクティビティ図におけるアクションを対象としている。

Commandパターンにおけるコマンドと競技に必要な動作の対応表を、表3.1-3に示す。

表3.1-1:システムの設計方針

根拠となる要求	設計方針
拡張しやすい構造にする	b10: Commandパターン ①競技動作をコマンドで表現するCommandパターンを採用する
来年以降に再利用したい	b11: レイヤー構造 ②レイヤー構造を採用する
パッケージ間を疎結合にする	b12: Facadeパターン ③Facadeパターンを採用する
ETロボコン2023攻略システム	b11: EV3APIのラッパー ④EV3APIのラッパーを作成する
走行体の計算コストを減らす	b10: カメラシステムによる顔の正面らしさの判別 ⑤計算コストの高い処理はカメラシステムで行う
継続的なテストを行いたい	b14: EV3APIのモック化 ⑥テストしやすい構造にする
確実に無線通信を行う	b13: 複数の通信手段を用意する ⑦複数の通信手段を用意する
確実に有線通信を行う	b13: 通信失敗時に再度通信を行う ⑧通信失敗時に再度通信を行う

具体策①  
動作をコマンドで定義している

表3.1-2: 検討すべき事項と具体策

対象アクション	検討すべき事項	具体策
走行体が撮影場所まで走行する	走行体の動作をどのように実現するか？	①設計方針①より、動作をコマンドで定義する ②設計方針①より、各エリアについて、それぞれコマンドファイルを読み込み、動作を実行する
フィグ撮影を行う	フィグ撮影をどのように実現するか？	③設計方針①②より、基本動作を含む、応用動作を定義する
配置エリアBでの撮影場所を指定する	撮影場所をどのように指定するか？	④事前に6つの撮影場所に対するコマンドファイルを定義し、キャリブレーション時にスターが選択する
カメラシステムにミニフィグ画像を送信する	送信失敗時の挙動はどうするか？	⑤設計方針⑧より、送信が成功するまで再送信する
画像を送信する（カメラシステム→競技システム）	送信失敗時の挙動はどうするか？	⑥設計方針⑧より、送信成功のレスポンステータスを受信できるまで再送信する
顔の正面らしさを算出する	どういった手法で顔を認識するか？	⑦機械学習を用いて物体検出する (詳細は4.2節参照)

表3.1-3: コマンドと動作の対応表

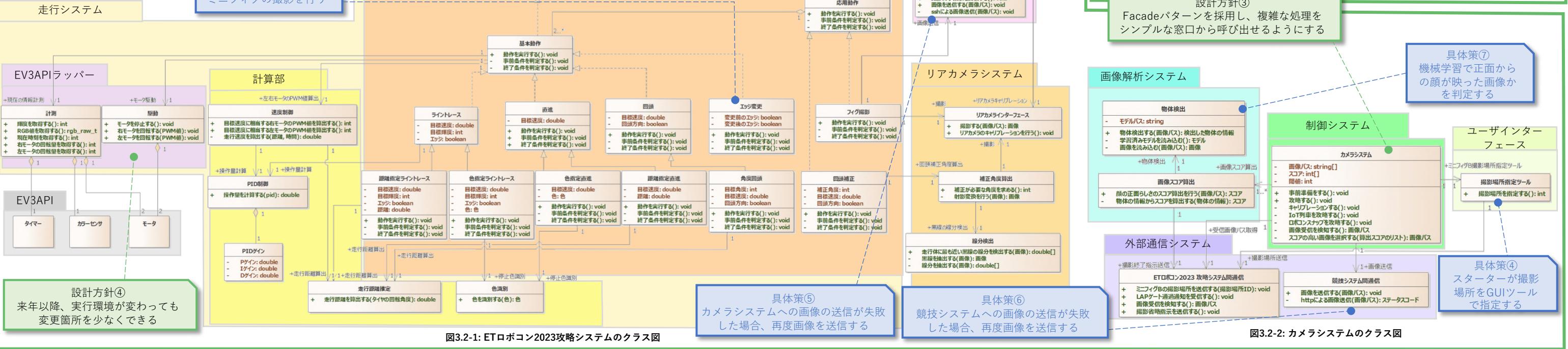
コマンド	基本動作	説明
DL	距離 ライントレース	指定距離走行するまでライントレースする
CL	色まで ライントレース	指定色を認識するまでライントレースする
DS	距離 直進	指定距離走行するまで直進する
CS	色まで 直進	指定色を認識するまで直進する
AR	回頭	指定角度回転するまで回頭する
EC	エッジ変更	指定されたエッジに切り替える
コマンド	応用動作	説明
CA	フィグ撮影	回頭と撮影を行うことでミニフィグを撮影する
XR	角度補正回頭	リアカメラ画像から回頭補正を行う

## 3.2 構造設計

3.1節で述べたアーキテクチャを踏まえて、構造を設計する。ETロボコン2023攻略システムの構造を図3.2-1に、カメラシステムの構造を図3.2-2に、それぞれ示す。なお、システムの構造は、以下のルールの下で記述する。

- setter及びgetterの操作については省略する。
- EV3APIモックパッケージは、EV3APIパッケージと同様の構造のため省略する。

具体策③  
応用動作のフィグ撮影でミニフィグの撮影を行う



### 3 設計モデル (2)

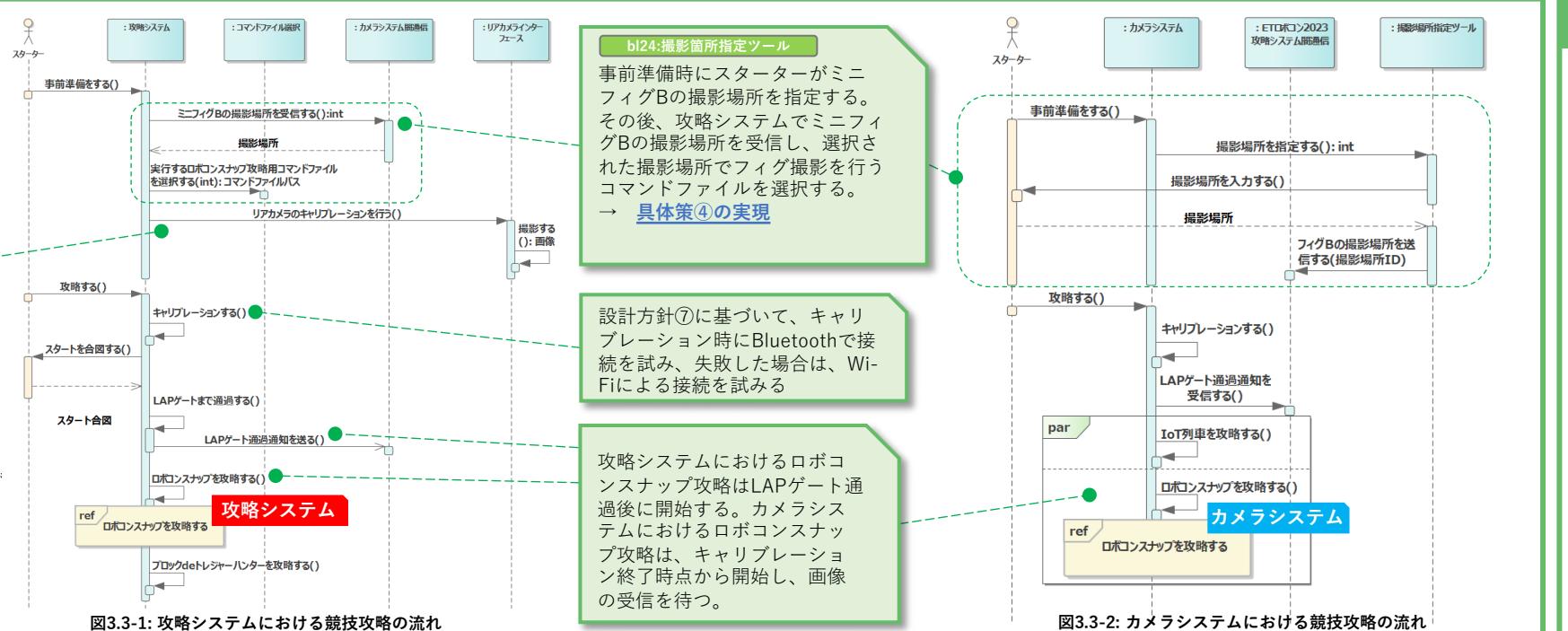
KatLab

#### 3.3 振舞設計 (全体)

3.2節に基づき、競技攻略において、攻略システムとカメラシステムの2つのシステムで作業を分担する。攻略システムにおける競技攻略の流れを図3.3-1に、カメラシステムにおける競技攻略の流れを図3.3-2に、それぞれ示す。

**bl28:リアカメラキャリブレーション**  
リアカメラ用いた回頭補正の誤差を小さくするために、ArUcoマーカ等を用いたキャリブレーション用の治具を使って、リアカメラキャリブレーションのための撮影を行う。この際、射影変換に用いるパラメータファイルを作成する。

図3.3-1と図3.3-2から、ロボコンスナップ攻略は2つのシステムで分担することが分かる。また、ブロックdeトレジャーハンターゴロコンはETロボコン2023攻略システムのみが行い、IoT列車攻略はカメラシステムのみで行う。上記の分担で、万が一、走行体が走行不能になった場合でも、IoT列車攻略に影響が出ない設計とした。



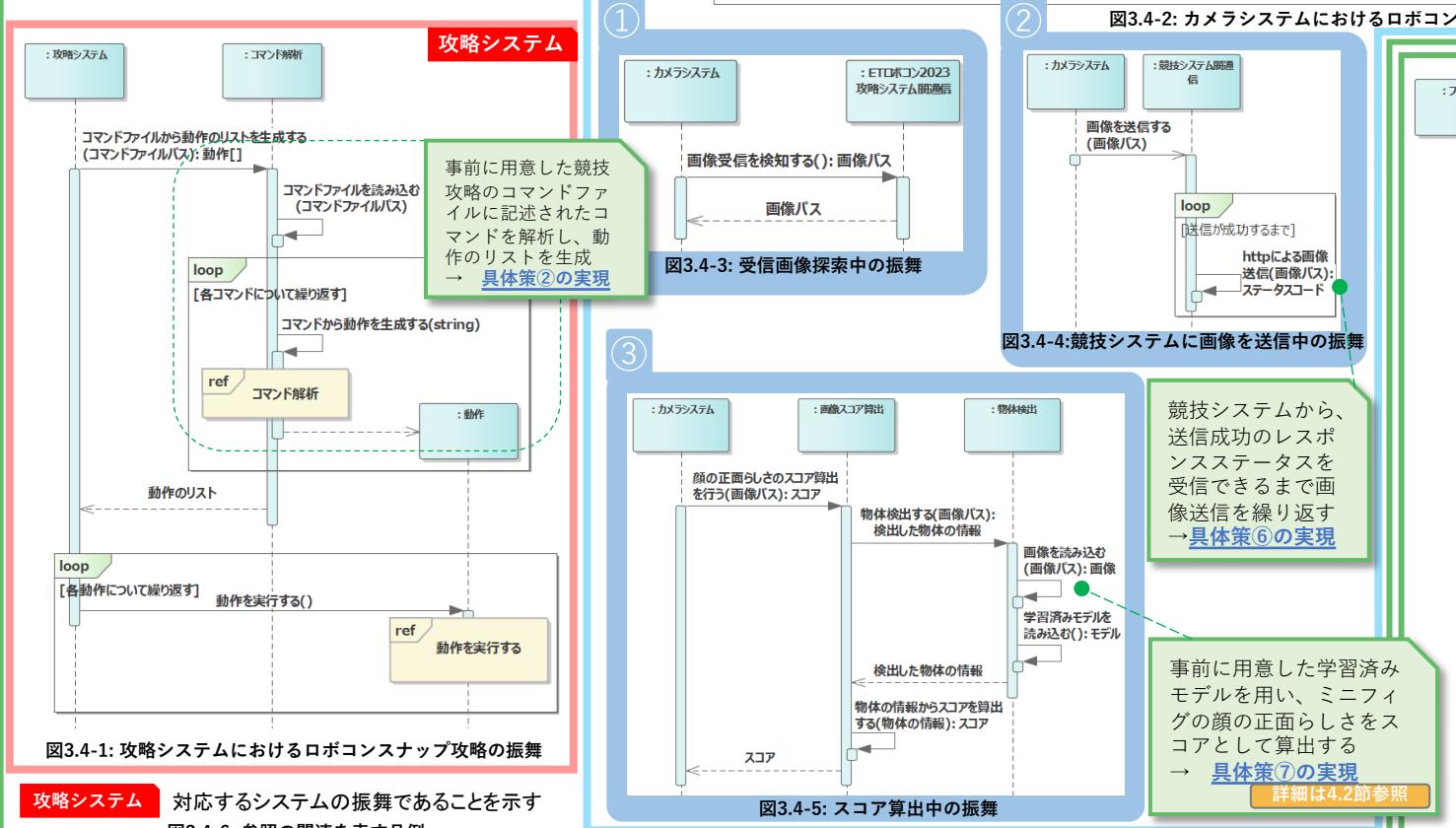
#### 3.4 振舞設計 (ロボコンスナップ)

3.3節に基づき、攻略システムとカメラシステムによる、ロボコンスナップ攻略の振舞を図3.4-1と図3.4-2にそれぞれ示す。また、図3.4-2に示す各状態における振舞を図3.4-3～3.4-7に示す。

攻略システムは、コマンドファイル読み込み(制御パラメータ)を実行する。これにより、制御パラメータの調整をする際に、コンパイルを省略することができ、効率よく開発することができる。攻略システムが読み込むコマンドファイルは、全部で3つあり、いずれも事前にコマンドと制御パラメータを調整したものを用意する。攻略システムが読み込む3つのコマンドファイルを、表3.4-1に示す。

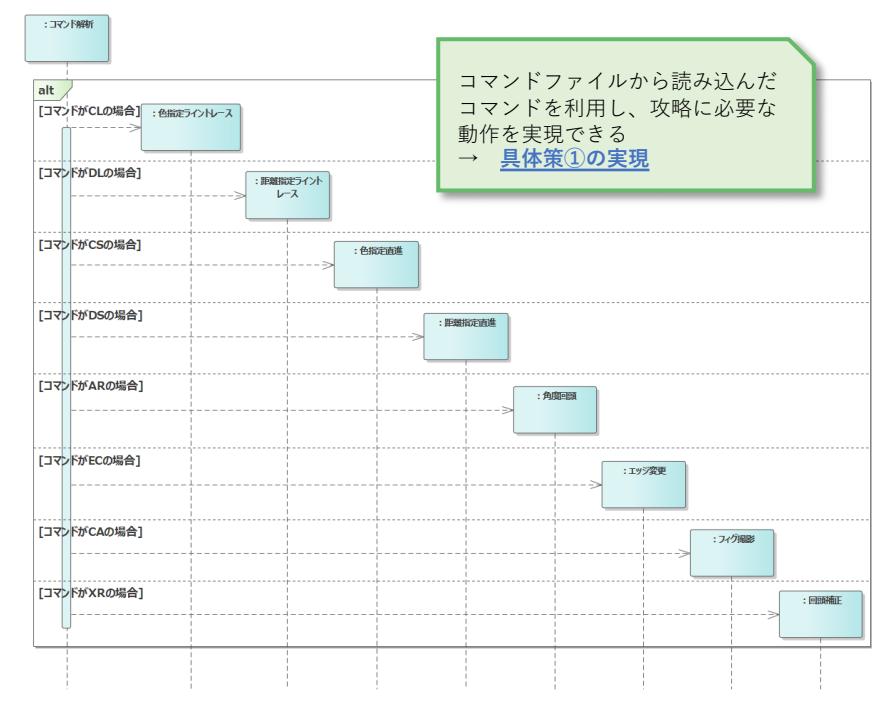
##### コマンドファイルの変更で他の競技を攻略できる振舞

図3.4-1の振舞は、競技によらない。これは、他の競技を攻略するに当たって、コマンドファイルの変更のみで攻略システムが競技を攻略できることを示している。



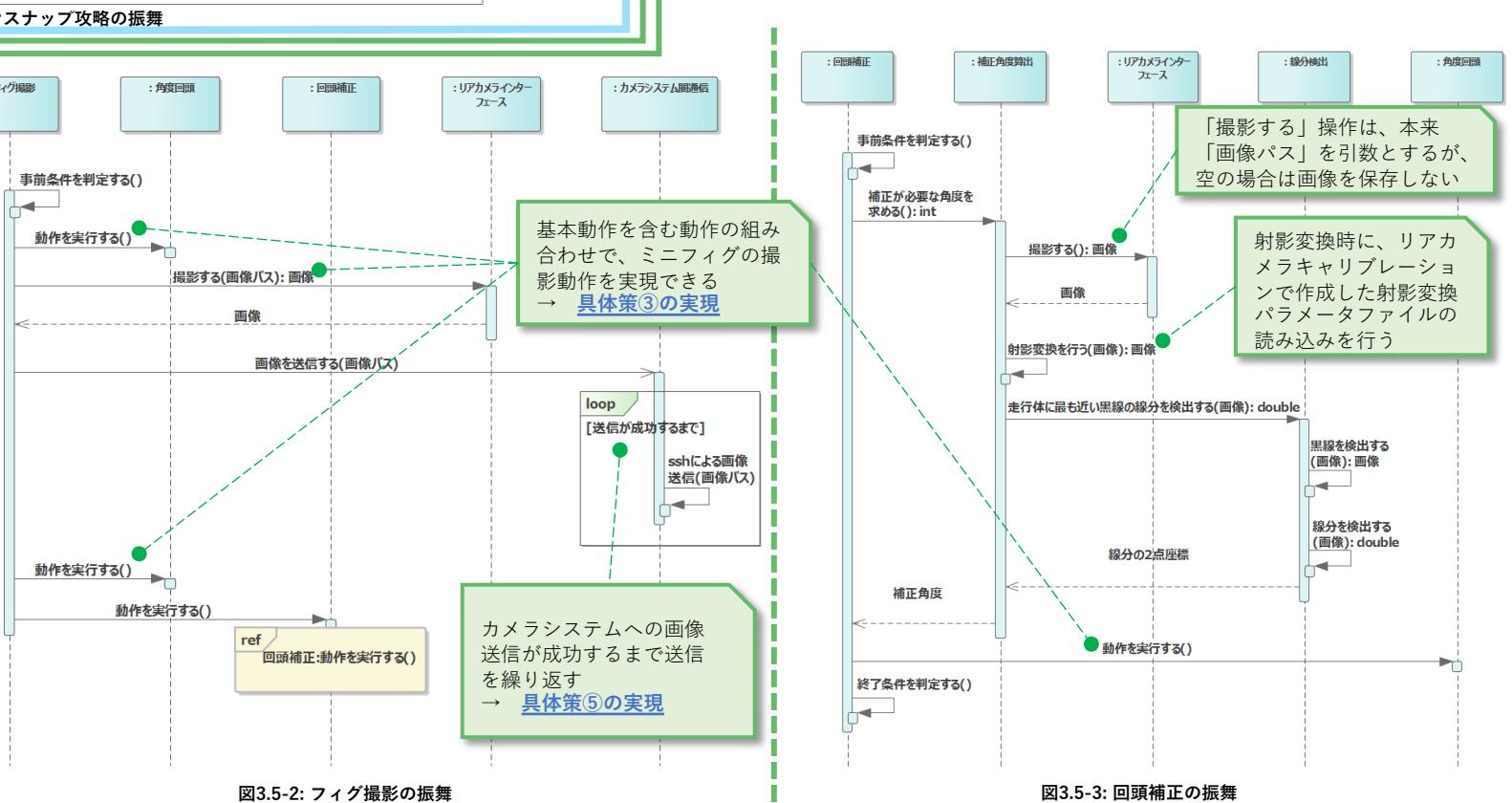
#### 3.5 振舞設計 (動作)

図3.4-1の攻略システムによるロボコンスナップ攻略の振舞では、「コマンドから動作オブジェクトを生成する」操作によってコマンドから動作オブジェクト生成し、動作リストを取得している。ここでは、コマンド一つ一つから各動作オブジェクトを生成するコマンド解析の振舞を図3.5-1に示す。



#### ロボコンスナップ攻略に関連した動作の振舞

図3.4-1の「動作を実行する」操作は、図3.5-1の振舞によって各コマンドから生成された動作オブジェクトをもとに動作を実行するため、動作によって振舞が異なる。ここでは、ロボコンスナップ攻略において、今回用いた動作であるフィグ撮影の振舞を図3.5-2に、回頭補正の振舞を図3.5-3にそれぞれ示す。紙面の都合上、他の動作の振舞については、省略する。フィグ撮影は回頭動作とリアカメラによる撮影動作、回頭補正を組み合わせて実現する。回頭補正是、リアカメラを利用しダブルループから曲線を検出することで、期待した回頭角度と実際の回頭角度の誤差を修正する回頭角度を算出し、角度指定回頭を行うことで実現する。



## 4.1 制御パラメータ調整支援ツール

### 4.1.1. 走行に対する制御戦略

bl16:区間制御

走行経路や黒線の形状に応じて、走行区間を細かく分割し、走行体の動作や制御パラメータを変更することで、走行区間に適した動作を設定する区間制御を行う。動作をコマンドとして表現し、コマンドファイルとして管理することで、各エリアにおける走行を実現する。走行体の動作に対応するコマンドを、3.1節のアーキテクチャに示す。例として、Lコースのライントレースエリアを走行する際における動作の組み合わせであるコマンドファイルを、図4.1-1に示す。

```
DL,2800,510,0,0.11,0.04,0.02
DL,530,180,0,0.9,0.5,0.12
DL,1650,510,0,0.11,0.04,0.02
DL,550,200,0,0.84,0.4,0.12
DL,100,200,0,0.22,0.08,0.04
CL,BLUE,-15,0.13,0.04,0.02
```

図4.1-1: Lコースのライントレースエリアを走行する際のコマンドファイル

### 4.1.4. 制御パラメータ調整支援ツール

bl8:制御パラメータの調整を支援するGUIツール

- 走行体からのコマンドファイル受信
- コースマップ表示
- コマンドファイルに対応する区間の強調
- コマンドの追加、削除、並び替え
- パラメータの編集**
- LコースとRコースの切り替え
- 走行体へのコマンドファイル送信
- PWM値と輝度値のログデータのグラフ描画

### パラメータの編集

Lコースのライントレースエリアを走行する際のコマンドファイルをツールで編集する様子を図4.1-2に示す。

走行体からコマンドファイルを受信すると、選択されたコマンドのリストに、自動でコマンドのセルを追加する。セルにはそれぞれのコマンドのパラメータを入力する欄があり、コマンドファイルのパラメータが初期値として入力される。

調整メンバーは**パラメータの入力欄の数値や選択肢を変更して、パラメータを編集する**。入力欄は、各パラメータの名前や単位を示し、ETロボコン2023攻略システムで扱う型に対応している。例えば、「輝度調整」は攻略システムでint型として扱うため、整数值の入力のみを受け付ける。「目標距離」はシステム内でdouble型として扱うため、実数值の入力のみを受け付ける。対応していない値「ア」や「0.0.1」などを入力すると、入力欄は変更を取り消す。また、指定色までライントレースを行うCLコマンドの色パラメータは、選択肢から1色を選ぶ形式にし、色のサンプルを表示している。

これらの機能により、**調整したいパラメータを見つけやすく、タイプミスをなくすことができる**。

### PWM値と輝度値のログデータのグラフ描画

Lコースのライントレースエリアを走行した際のログデータをグラフで可視化した様子を図4.1-3に示す。グラフは、左モーターのPWM値を青、右モーターのPWM値を赤、輝度値を灰色で表す。y軸はPWM値や輝度値を表し、x軸はログを取得した時間(0.1秒刻み)を表す。それぞれのグラフの表示をトグルボタンで非表示に切り替えることができる。また、ログデータには外れ値が含まれるため、y軸の上限を50から400、下限を0から-400まで変更することができる。ログデータの可視化は解決すべき課題に含まれていないが、輝度値の可視化により、調整メンバーはカーブの際に輝度値が高い状態が続いているためPゲインの値をもっと高くする必要があるとわかった。また、可視化したPWM値は、速度フィードバック制御の精度を測る指標となっている。

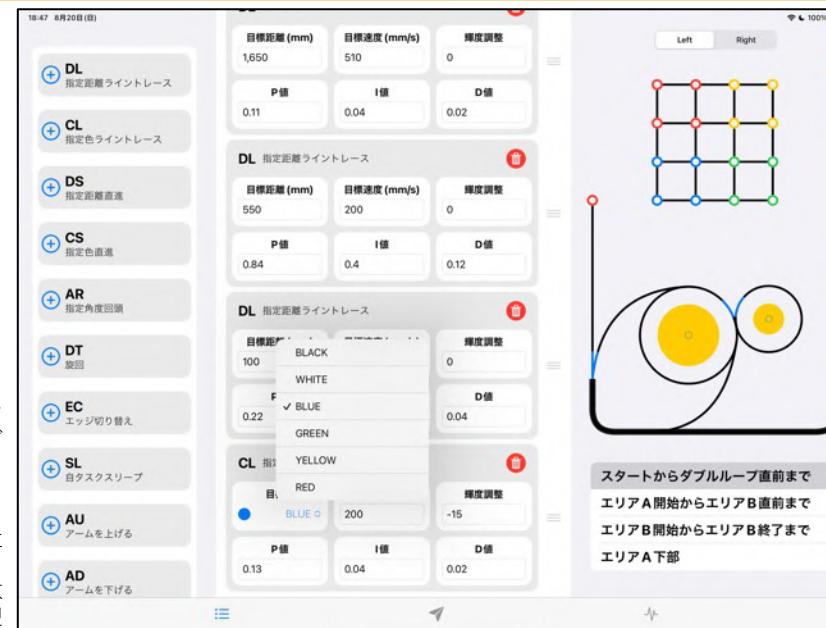


図4.1-2: Lコースのライントレースエリアを走行する際のコマンドファイルをツールで編集する様子

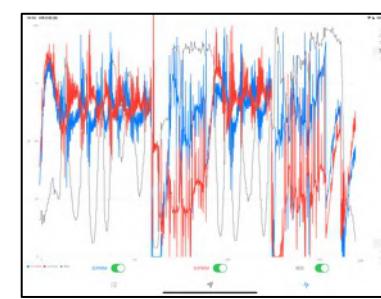


図4.1-3: Lコースのライントレースエリアを走行した際のPWM値(左が青、右が赤)と輝度値(灰)のグラフを描画した様子

### 4.1.5. ツールの検証

#### 実験

ツールを用いることで課題を解決できたかどうか検証するため、実験を行った。実験において、調整メンバーは従来の手法とツールを用いる手法それぞれでコマンドファイルの変更を10回ずつ行った。奇数回目では、従来の手法で調整をした後、同じ内容の変更をツールを用いて行った。偶数回目では、ツールを用いて調整をした後、同じ内容の変更を従来の手法で行った。変更点は、1回ごとにランダムに決定した。

#### 考察

実験結果より、走行体へのデバイスの接続から攻略システムの実行までにかかる時間を、約72%減少することに成功した。このことから、ツールを用いることで、競技を攻略するために必要な調整にかかる作業時間を減らすことができる。また、従来の手法では、PCと走行体を接続する際にSSH接続を行うが、この接続に失敗することがある。対して、ツールの接続ではSSH接続を用いないため、接続が安定する。さらに、コマンドファイルを直接編集する従来の手法では存在するタイプミスが、ツールを用いた際には起きなかった。よって、ツールを用いることで課題を解決できた。

### 4.1.2. 課題

図4.1-1のように、各動作コマンドには複数の制御パラメータが存在する。そのため、競技の攻略には、各エリアの動作コマンドごとに制御パラメータを調整する必要があり、その調整には時間がかかる。

コマンドファイルはcsv形式のファイルであり、コマンドとその制御パラメータが列挙されている。例えば、1行目のDLコマンドの引数は、左から順に「距離(mm)」「速度(mm/s)」「基準輝度値に対する補正値」「Pゲイン」「Iゲイン」「Dゲイン」である。しかし、これでは調整時に、どの値が調整したい制御パラメータかわかりにくい。また、タイプミスをする可能性もある。

### 4.1.3. 課題に対する解決策

制御パラメータを調整する手間を削減するために、**制御パラメータ調整支援ツールを開発する**。支援ツールの開発は、研究室が所有しているiPadがあること、調整時の走行体との接続やパラメータ変更をGUIで行えることから、iOSアプリとして開発する。

## 4.2 ベストショットに適した画像の選択

### 4.2.1. 制御戦略

ベストショットを成立させるために必要な要素(両目、口)を含んでいる画像を選択する。

- 課題を満たす手法を選ぶ際に以下を考慮する。
- ①ミニフィグや環境の変化に対応できる。
  - ②開発時間は限られており、総合的に開発コストが高くなりすぎないようにする。
  - ③大会当日の作業量を少なくしたい。

### 4.2.2. 課題に対する解決策

#### 案1 機械学習による物体検出

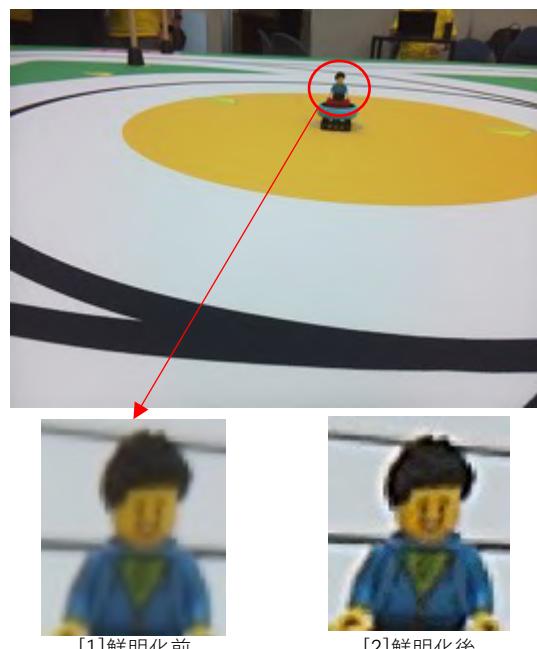
- ・ミニフィグの姿勢や大きさなどの変動に対応可能
- ・複数のパターンに対応可能

#### 案2 パターンマッチング

- ・開発コストが機械学習より低い
- ・大会当日の作業量を少なくしたい。

### 「案1 機械学習による物体検出」を以下の理由により採用

- 1.以下の理由によりパターンマッチングはテンプレート画像を大会当日に用意しなければならない③
  - ・使用されるミニフィグは大会当日まで分からない
  - ・ミニフィグのポーズは試走後まで分からない
- 2.撮影距離によって画像上のフィグの大きさが変わるが、機械学習はこのような変化に強い③
- 3.様々なパターンを学習させることで、環境の変化に対応できる
  - ※試走会1,2,開発環境の3ヵ所の画像を混合させて学習①



[1]鮮明化前

[2]鮮明化後

図4.2-1: リアカメラによる撮影した画像

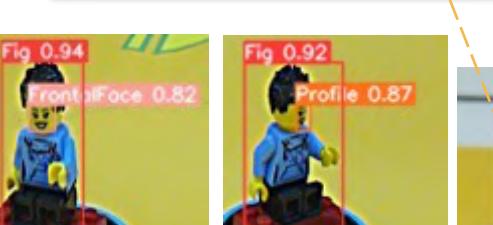
表4.2-1: 物体検出から得られる情報

得られる情報	解説
物体の位置情報	物体の位置を囲むバウンディングボックスの左上と右下の座標を検出する。
物体のクラス情報	識別した物体が設定したクラスのどれに属するか同定する。
物体の信頼度スコア	信頼度スコアは物体の検出が正確である確率を表す。

表4.2-2: 検出するクラス

クラス	項目の詳細
Fig	ミニフィグ全体
Frontal Face	ミニフィグの顔(両目口が認識できる)
Profile	一部が隠れたミニフィグの顔

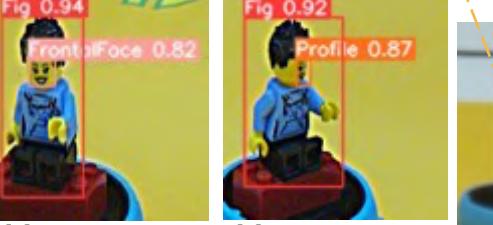
Frontal Faceを二重で検知してしまっている



[1]正面を向いたミニフィグ画像

[2]正面を向いていないミニフィグ画像

図4.2-2: 顔を2種類のクラスに分ける効果



[1]鮮明化なし画像での検出

[2]鮮明化あり画像での検出

図4.2-3: 鮮明化の効果