ETロボコン2019

デベロッパー部門 アドバンストクラス

チームNo. 030　　チーム名：AT車限定～Gold License～

地区：南関東

所属：富士ゼロックスアドバンストテクノロジー株式会社

このサンプルに使っているモデル図は2018年のものです。（AT車限定～Gold License～の協力で流用しています）2019年とは競技内容や審査規約が異なります。どんな内容を記載するのかについては、必ず2019年の規約を確認しましょう。

チーム紹介、目標、意気込み

チームのことや、今回のチャレンジへの意気込みなどをアプールしましょう

○チーム紹介
　AT車限定～Gold Li
トテクノロジー株式会
のみで構成されており、今年は2年目の社員が多くフレッシュな気持ちを大切にして取り組んできました。
　モデル図・ソースコード共に手探りで開発に取り組んできましたが、頼りになる先輩の指導のもと、一致団結して活動に取り組んできました。
○目標：もちろん全国大会優勝です!!地区大会での経験を活かし、全難所を突破して見せます!!
○意気込み：モデル、走行ともに1位を目指して、頑張って来たので悔いの

モデル図に書いてあることをコンパクトにまとめましょう（補足じゃないですよ）

モデルの概要

ここを読めば、どのような分析から、目標タイム、得点方法、技術的対策などといった自チーム固有の要求が得られたのかわかるようにしましょう

○要求：チームとしての要求
いことをかみ砕くことで、仕

○分析：重要課題であるブロ
を攻略する上で、ゲームの構成要素を図式化し、何をもってゲームを制すかを分析した。

○設計（構造）　　：クラス
計した。分析結果を反映し、
と値のやり取りを明確にし

○設計（振る舞い）：クラス
として見える形で設計した。
全体動作を、シーケンス図で詳細な動作を設計した。

○制御：高い信頼性を確保し　また、ゲーム攻略に必要な要素技術と制御方法を記載した。
サーで必要となる攻略方法

「～という方式を使い、～方式に比べてXXくらい短時間で数字を認識できるようになった」など具体的な方式、結果、他のチームとの効果や確度の違いなどを書くとよいでしょう

構造や振舞い、制御面の設計などについて、図を見なくても結果、効果、価値がわかるように書きましょう

モデルの構成

どんなことについて書いた図や説明が登場し、互いにどのように関係しあっているのかわかるようにまとめましょう

1. 要求
- 目標設定：チーム
- 機能要求：ユースケースを用いた必要機能の分析
- 非機能要求・要素技術：非機能要求と要素技術を洗い出した
- 仕様：洗い

2. 分析
- ゲームの要
- 解決方針：抽出した課題の解決方針の検討
- 解法：分析クラス図（解法）とシーケンス図を作成

3. 設計（構造）
- パッケージ構造：クラスの値の流れをパッケージ図で表現
- 区間制御：ゲーム攻略で重要となるパッケージの解説
- ：パッケージを反映したクラス図の設計

舞い）
- シン設計：制御状態をステートマシン図で設計
- 設計：攻略区間の動作をシーケンス図で設計

5. 制御
- 自己位置推定：走行距離と旋回角度を用いた自己位置の特定
- 制御：カ
- クストラ
- の特定方
- 体の回転
- プレ走行によ
- 化フィル
- AP制御：避けることができないブロックの避難
- ック位置の導出方法：初期位置コードの変換方法

何をやったのかを書いても内容は伝わりません
何を決めた、洗い出す等して何が得られた、落とし込んだ仕様はどんなものになったかを書きます

図を書いていることは書かなくてもわかります
その図で何（どこ）が見てほしい点か、他のチームと比べてどう異なっているのか、方法はどんな効果や結果をもたらしたのかわかるように書きます

1.要求　AT車限定 ～Gold License～

要求モデル（１ページ）

1.1目標設定

全国大会優勝

- 走行競技No1!!
- モデル審査A判定!!

■ 走行競技No1!!

L+Rコースの合計リザルトタイム **-26秒**を目指す。
走行タイムは、L+Rコース共に16秒を目指す。地区大会では確実にゴールするため、バッテリー残量を減らし、速度を落とすことで安全な走行をした結果、20秒となった。
図1.1の実測値から速度を上げれば16秒を達成することを確認済み。
ボーナスタイムについて、Rコースは-28秒を目指す。Lコースは-30秒を目指す。
(28秒の理由は[2.2 解決方針]④に示す)

■ モデル審査A判定!!

A判定が取れるモデル図を書くことで、精度高く安全な実装につながると考えた。
A判定になるには…

［要求］：目標を達成するための要求を段階的に分解して機能要求と非機能要求を明確に示す。
［分析］：ゲームを解くために必要な情報と解き方を十分に分析し、明確に示す。
［設計］：設計意図を明記して、要求と分析結果からソフトウェアの構造と振る舞いを設計する。
［制御］：難所を攻略するための要素技術を的確に選定し、その必要性や効果を明確に示す。

表1.1 目標リザルトタイム内訳

	Lコース	Rコース
走行タイム	16秒	16秒
ボーナスタイム	-30秒	-28秒
リザルトタイム	-14秒	-12秒

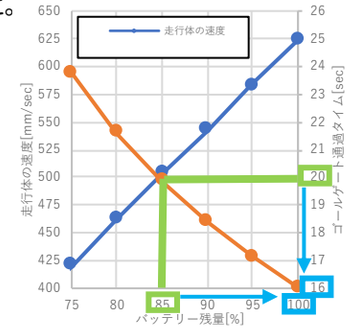


図1.1 目標走行タイム設定

1.2機能要求

我々のチームでは、全国大会優勝にはより速い走行タイムを取得する事が必須であると考えた。さらに、高いボーナスタイムを獲得し、最速のリザルトタイムを残すことが

このサンプルに使っているモデル図は2018年のものです。（AT車限定～Gold License～の協力で流用しています）2019年とは競技内容や審査規約が異なります。どんな内容を記載するのかについては、必ず2019年の規約を確認しましょう。

図1.2 ユースケース図

事前条件	走行体のキャリブレーションが完了していること
基本フロー	走行体が中間ゲートを通過する
事後条件	走行体がゴールゲートを通過していること

事前条件	走行体がゴールゲートを通過していること
基本フロー	走行体がブロック並べゲームを攻略する。 走行体が直角駐車場で停止する
事後条件	走行体が時間内にボーナスタイム獲得条件を満たしていること

1.3非機能要求・要素技術

［1.2機能要求］で示したユースケース図に関して非機能要求の分析を行い、必要な要素技術を抽出した。

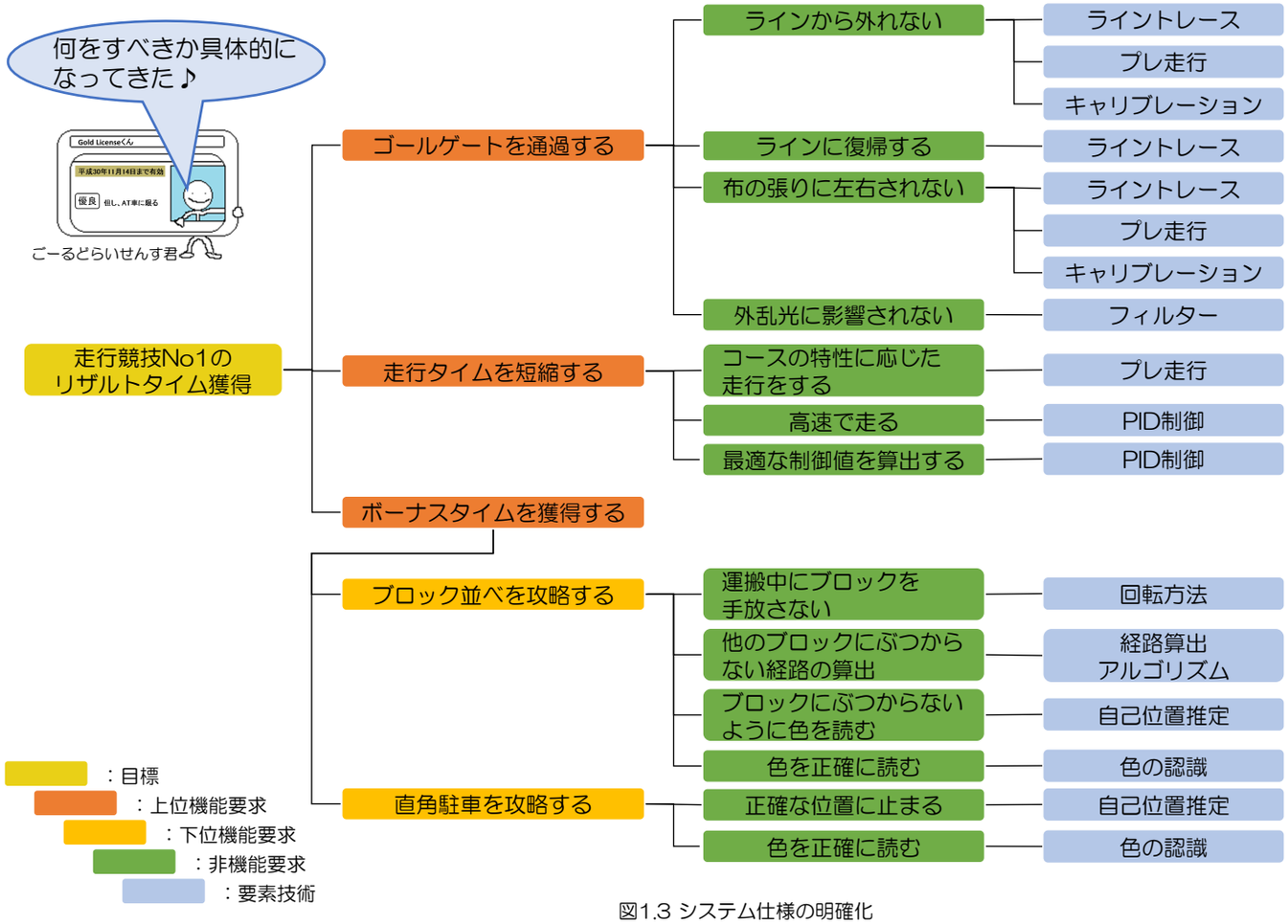


図1.3 システム仕様の明確化

1.4仕様

［1.3非機能要求・要素技術］で洗い出した要素技術を仕様に落とし込んだ結果を表1.4に示す。目標を達成するために必要な作業項目が明確になった。

表1.4 仕様抽出

ゴールゲートを通過する	
ライントレース	仕様：黒のラインに沿って走行する。
プレ走行［5.6 プレ走行によるパラメータ決定］	仕様：コースを測量する。
	仕様：コースの色の値を取得する。
キャリブレーション	仕様：スタート位置調整
	仕様：アーム位置調整
フィルター［5.8 平均化フィルター］	仕様：突発的な色の変化による影響を防ぐ
	仕様：カラーセンサから読み取る精度を上げる
走行タイムを短縮する	
プレ走行	仕様：「ゴールゲートを通過する」の「プレ走行」と同様
PID制御	仕様：現在値と目標値の差から最適な制御値を算出
ボーナスタイムを獲得する	
ブロック並べを攻略する	
回転方法	仕様：ブロックを走行体に押し付けて回転する
経路算出アルゴリズム	仕様：ブロック位置を把握する
	仕様：目的のブロック置き場を決める
	仕様：目的のブロック置き場までの経路を算出する
自己位置推定	仕様：走行距離と回転角度の算出
色の認識	仕様：色情報を取得し、正確に色を区別する
直角駐車を攻略する	
自己位置推定	仕様：「ボーナスタイムを獲得する」の「自己位置推定」と同様
色の認識	仕様：「ボーナスタイムを獲得する」の「色の認識方法」と同様

2. 分析

AT車限定 ～Gold License～

分析モデル

（1ページもしくは次ページへ続く）

2.1ゲームの要素定義

ブロック並べの要素定義を行い、ゲーム攻略に必要な課題を列挙した。

- **ゲームの構成** … 図2.1の中のブロック並べパッケージに示す。
- **ゲームの前提** … 走行体初期位置は8番

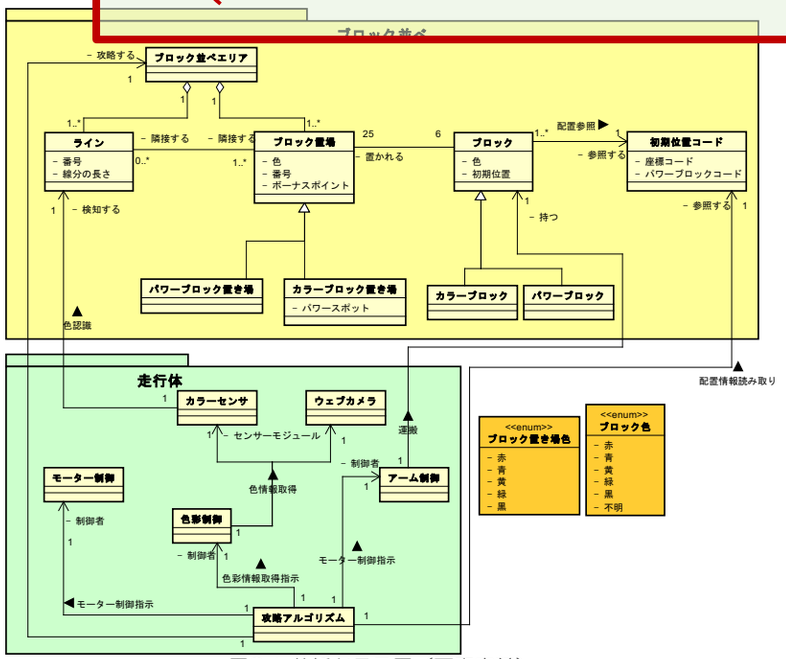


図2.1 分析クラス図（要素定義）

- **課題** … [1.4 仕様]で抽出した仕様と、競技規約から作成した分析クラス図（図2.1）から、課題は以下のようになった。

表2.1 課題の抽出

仕様	課題	解決方法
ブロックを走行体に押し付けて回転する	押し付けて回転する具体的方法	[5.4 回転方法]
ブロック位置を把握する	初期位置コードの取得方法	[2.2 解決方針] ①
	初期位置コードからブロック位置の特定方法	[5.9 ブロック位置の導出方法]
	ブロック位置の更新方法	[2.2 解決方針] ②
目的のブロック置き場を決める	目的の運搬先にブロックがあった場合の対処法	[2.2 解決方針] ③
	ブロックの運搬先の決定方法	[2.2 解決方針] ④
	運ぶブロックの決定方法	[2.2 解決方針] ⑤
目的のブロック置き場までの経路を算出する	最短経路アルゴリズムの選択	[2.2 解決方針] ⑥
	全体経路の決定（走行体がどこを走るか）	[2.2 解決方針] ⑦
走行距離と回転角度の算出	走行距離と回転角度の算出方法	[5.1 自己位置推定]
色情報を取得し、正確に色を区別する	ブロックの色情報の取得方法	[2.2 解決方針] ⑧
	ブロックの色の判別精度向上	[2.2 解決方針] ⑨

2.2解決方針

[2.1 ゲームの要素定義]から抽出した課題の解決方針を検討した。解決方針を表2.2に示す。

表2.2 解決方針

課題	解決方針
①初期位置コードの取得方法	キャリブレーショ めるために、PCて 通信をする方式を
②ブロック位置の更新方法	ブロック置き場情 行体がブロックの ロックを運び終え
③目的の運搬先にブロックがあった場合の対処法	手数が増えるパタ ブロック置き場の方をどかす。 （[5.8 SWAP制御]）
④ブロックの運搬先の決定方法	有効パワーブロックの判定基準が厳しいこと、移動成功時の加点が低いことから、パワーブロックは移動せず、カラーブロックを周りに並べる方法をとる（最高ボーナスタイム-28秒）。走行中の計算処理削減のため、図2.2のように、パワーブロックの初期位置によって、予めブロックの目標エリアを決定しておく。 (表4.3 目的のブロック置き場の決定方法)
⑤運ぶブロックの決定方法	SWAP制御の実施を極力避け、制限時間内にゲーム攻略を終了するために、目標エリアにあるブロックを第一優先にする。 (表4.3 目的のブロック置き場の決定方法)
⑥最短経路アルゴリズムの選択	状況によって計算時間が変わらないこと、アルゴリズムが簡素なためダイクストラ法を採用する。（[5.3 ダイクストラ法]）
⑦全体経路の決定（走行体がどこを走るか）	全体経路は図2.2に示すカラーブロック間の黒ラインである。確実な走行と計算処理削減のため、黒ラインがないところは経路とせず、ライントレースを用いて走行する。
⑧ブロックの色情報の取得方法	ウェブカメラは光の外乱が多く、色を誤検知しやすいため、カラーセンサを採用する。
⑨ブロックの色の判別精度向上	過去の大会の結果から効果があることを確認しているためHSVを採用する。 （[5.2 色彩制御]）

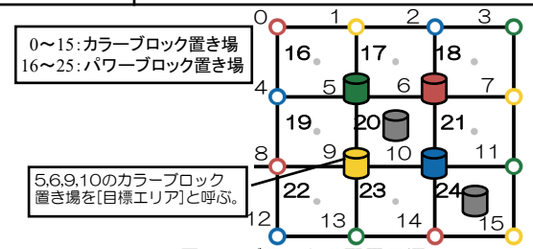


図2.2 ブロックの配置目標

2.3解法

[2.2解決方針]に基づいて、ゲーム攻略に必要な要素をクラス図（図2.3）に示す。また、クラス間のシーケンスを図2.4に示す。

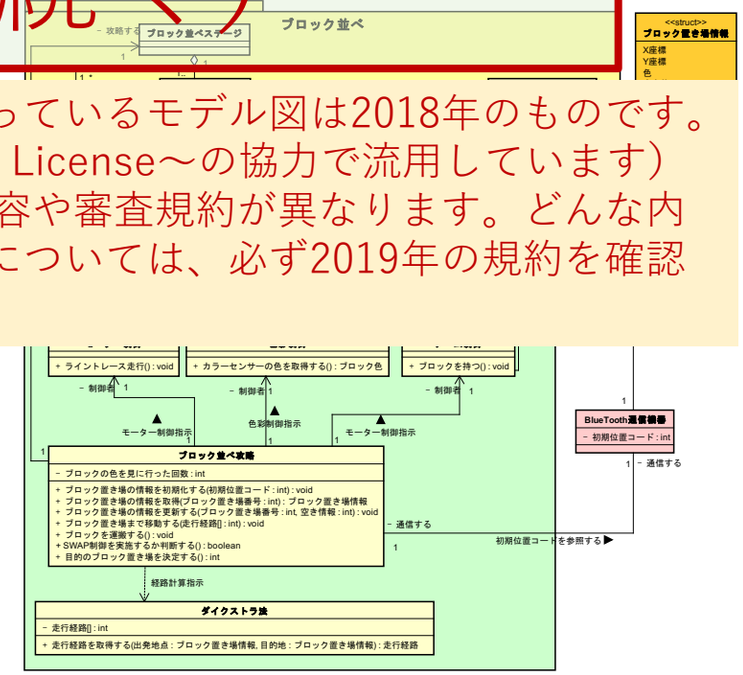


図2.3 分析クラス図（解法）

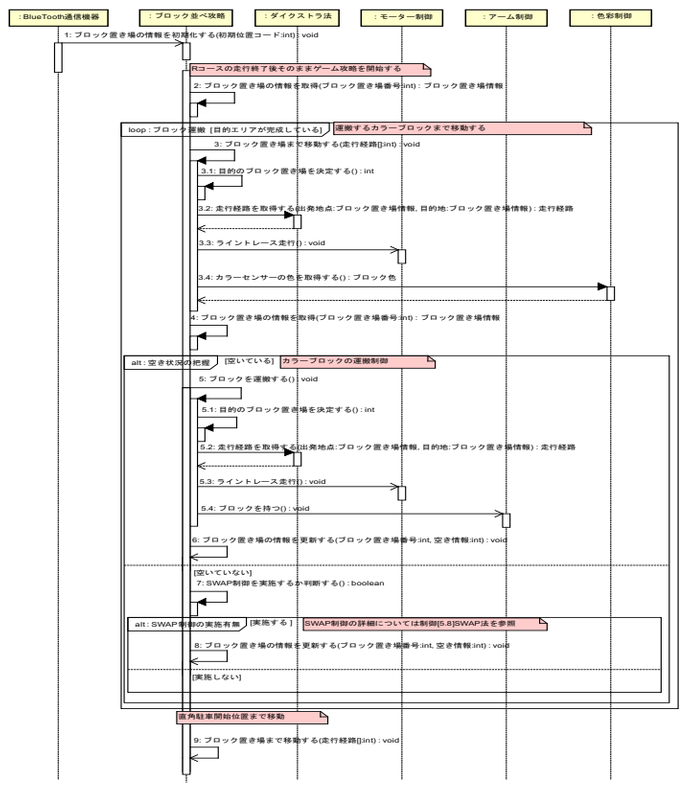


図2.4 分析シーケンス図（解法）

このサンプルに使っているモデル図は2018年のものです。（AT車限定～Gold License～の協力で流用しています）2019年とは競技内容や審査規約が異なります。どんな内容を記載するのかについては、必ず2019年の規約を確認しましょう。

3.設計（構造）

AT車限定 ～Gold License～

分析モデルまたは設計モデル（どちらか、あるいは両方に使える）

3.1パッケージ構造

クラス設計を分かりやすくする為、図3.1のパッケージ図の流れでクラス図を表現する。また、今回の設計においての設計思想について下に記す。

設計思想

クラス全体を通して、クラス間での値のやり取りを減らし、制御に必要なパラメータと役割を明確にした。

また、競技規約から、記載する範囲は、ブロック並べの攻略を主な役割とした範囲を記載し、ほかの制御部分に関しては省略している。

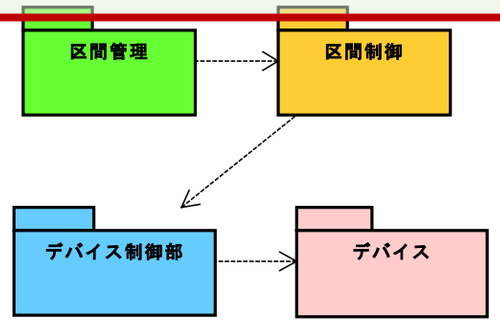


図3.1 パッケージ構造

3.2区間制御

パッケージの中でも、ブロック並べ攻略の主となる区間制御パッケージの解説を行う。

上位の区間管理パッケージからの命令を基に、一連の動作を生成し、実行する。

[3.2解決方針]で記載したブロック置き場に関連するブロックの配置や色に関連する情報と、走行体の位置情報を管理する。

区間制御

ブロック並べ攻略区間

ブロック並べ攻略情報

値の受け渡
[ブロック並べ攻略区間
[ブロック並べ攻略情報
ブロック置き場情報と
情報を取得し、
[動作パラメータ算出]
の位置・方向、ブロック置き場の情報を受け渡す。

報により走行経路をたどることができる動作パラメータを算出する。

このサンプルに使っているモデル図は2018年のものです。（AT車限定～Gold License～の協力で流用しています）2019年とは競技内容や審査規約が異なります。どんな内容を記載するのかについては、必ず2019年の規約を確認しましょう。

3.3クラス設計

クラス設計を右に記載する。クラス設計はブロック並べの攻略を主としており、他区間の攻略については省略している。

また、パッケージ群には記載はしていないが、構造体やenum型配列を用いた値の定義を行っている。

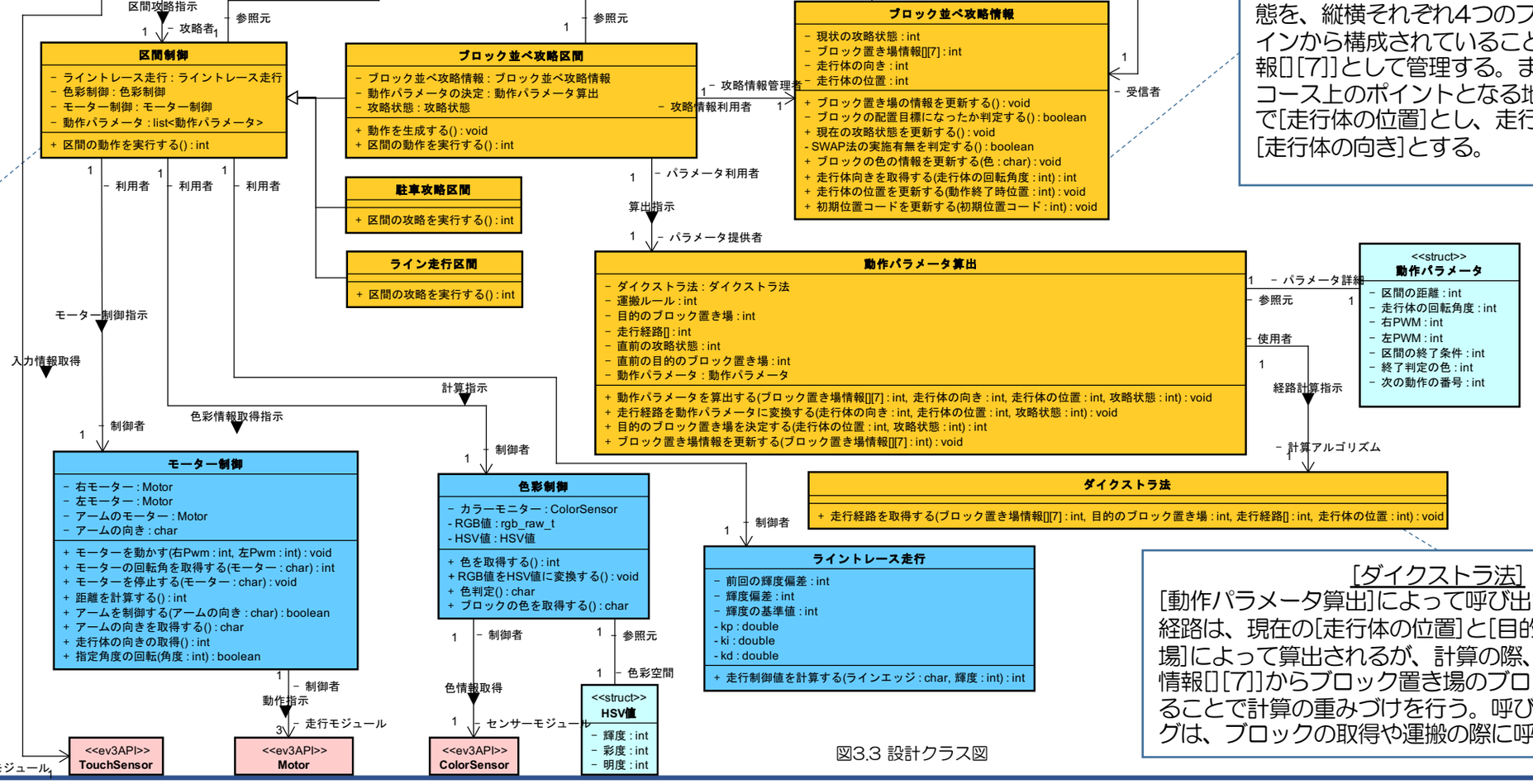


図3.3 設計クラス図

[区間制御]
[区間制御]はそれをスーパークラスとした複数の子クラスで構成される。それぞれ共通する動作として操作:[区間の攻略を実行する]が存在するが、[ブロック並べ攻略区間]では、柔軟なコースへの対応が必要であり、ブロック並べを複数の詳細な「動作」で攻略し、現在の「動作」が終了し次第、次の「動作」を生成する。

[ブロック並べ攻略情報]
ブロック並べ攻略情報では、[ブロック置き場]の状態を、縦横それぞれ4つのブロック置き場、3つのラインから構成されていることから[ブロック置き場情報[][7]]として管理する。また、走行体の位置をコース上のポイントとなる地点を数値に変換した値で[走行体の位置]とし、走行体の向いている方向を[走行体の向き]とする。

[ダイクストラ法]
[動作パラメータ算出]によって呼び出される。算出する経路は、現在の[走行体の位置]と[目的のブロック置き場]によって算出されるが、計算の際、[ブロック置き場情報[][7]]からブロック置き場のブロックの有無を調べることで計算の重みづけを行う。呼び出されるタイミングは、ブロックの取得や運搬の際に呼び出される。

4. 設計（振る舞い）

AT車限定 ～Gold License～

設計モデル
（1ページもしくは前ページからの続き）

4.1 ステートマシン設計

区間管理

区間管理内部の攻略状態をステートマシン図で表した。ブロック並べ攻略区間の前後の状態遷移を図4.1に、生成した動作の終了条件を表4.1に示す。

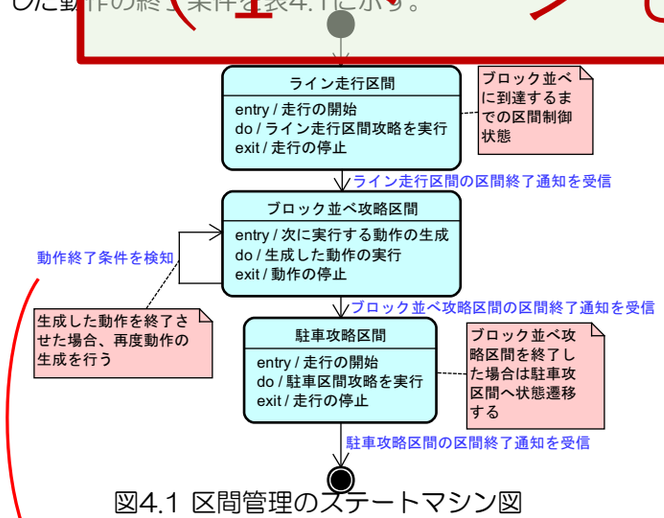


図4.1 区間管理のステートマシン図

表4.1　動作の終了条件

動作名	動作終了条件
前進動作/後退動作	指定距離を移動したこと
方向転回動作	指定角度回転したこと
アーム昇降動作	アームが指定角度回転したこと

攻略状態

攻略状態により、ブロック並べ攻略における制御状態を管理する。[正常動作中]の状態を繰り返すことでブロック並べを攻略する。図4.2中の終了条件について表4.2に示す。

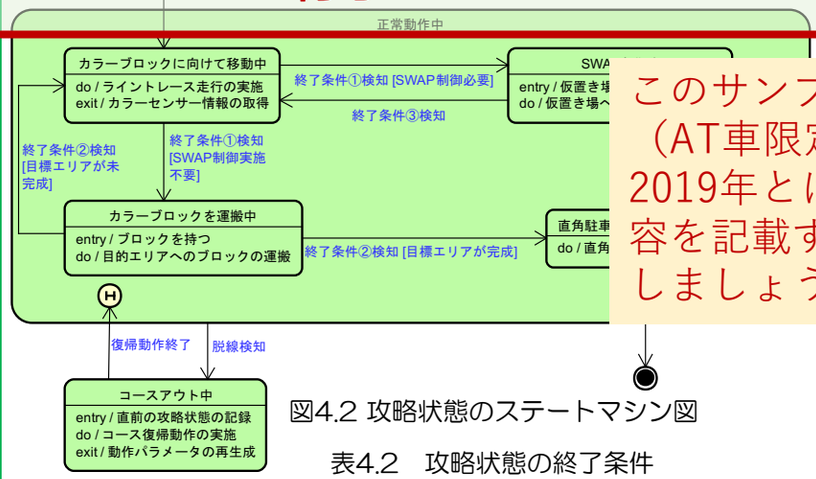


図4.2 攻略状態のステートマシン図

表4.2　攻略状態の終了条件

条件名	条件の内容
終了条件①	カラーブロックのあるブロック置き場に到着したこと
終了条件②	目標エリアにカラーブロックを運搬したこと
終了条件③	仮置き位置にカラーブロックを運搬したこと
終了条件④	最終ポジションとなるブロック置き場に到着したこと

4.2 シーケンス設計

■目的のブロック置き場の決定方法

目的のブロック置き場は攻略状態及びブロック置き場の状況によって変わり、動作生成時に毎回更新を実施する。表4.3に規則を示す。

表4.3 目的のブロック置き場の決定方法

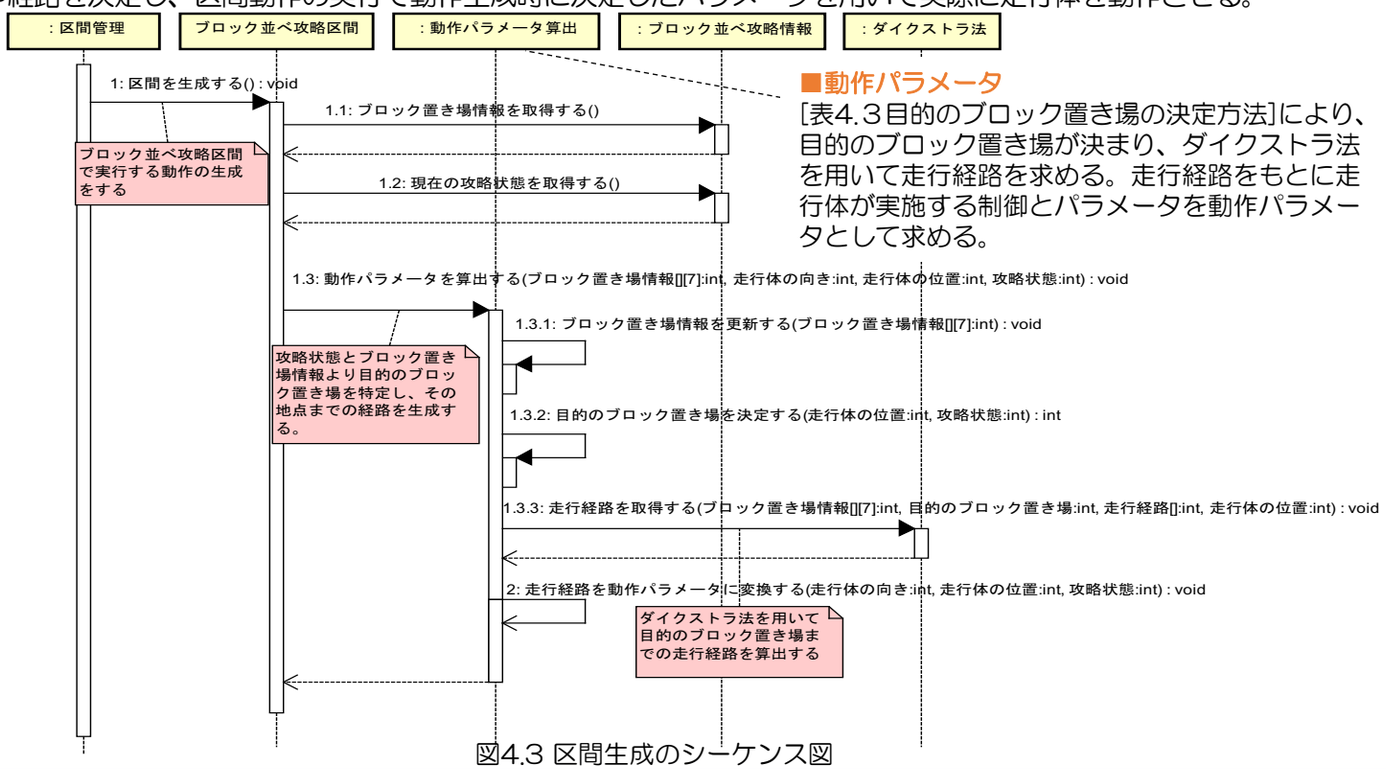
駐車スタート位置に移動する	-	最終ポジションとなるカラーブロック置き場

※目標エリア：[2.2解決方針]で定めた目標エリア

このサンプルに使っているモデル図は2018年のものです。（AT車限定～Gold License～の協力で流用しています）2019年とは競技内容や審査規約が異なります。どんな内容を記載するのかについては、必ず2019年の規約を確認しましょう。

区間生成の振る舞い

ブロック並べにおける区間生成及び動作の一連の流れを図4.3に示す。主に動作生成により、走行体を制御する際のパラメータや経路を決定し、区間動作の実行で動作生成時に決定したパラメータを用いて実際に走行体を動作させる。

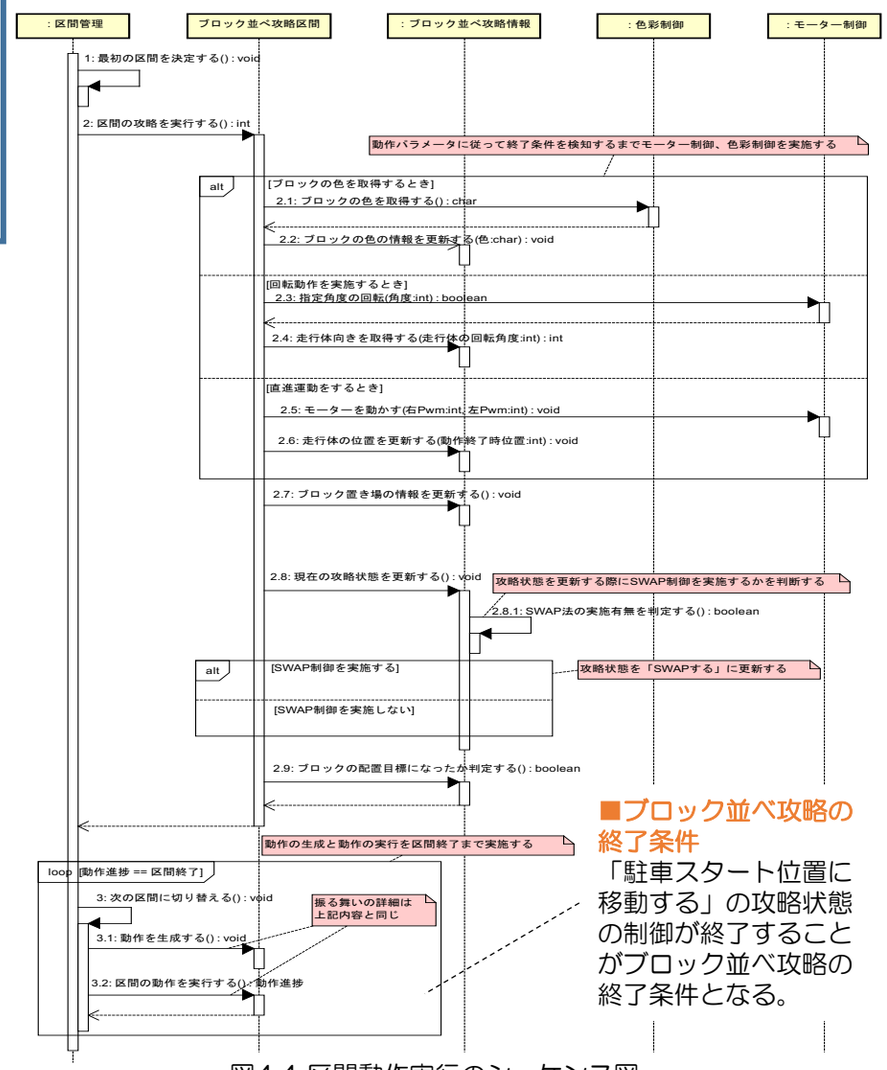


■動作パラメータ

[表4.3目的のブロック置き場の決定方法]により、目的のブロック置き場が決まり、ダイクストラ法を用いて走行経路を求める。走行経路をもとに走行体が実施する制御とパラメータを動作パラメータとして求める。

図4.3 区間生成のシーケンス図

区間動作の実行の振る舞い



■ブロック並べ攻略の終了条件

「駐車スタート位置に移動する」の攻略状態の制御が終了することがブロック並べ攻略の終了条件となる。

図4.4 区間動作実行のシーケンス図

5. 制御

制御モデル（1ページ）

AT車限定 ～Gold License～

ブロック：並べるゲーム攻略に使用する制御：ブロック　AIアンサーのゲーム攻略に使用する制御：AI　走行制御：走行

5.1 自己位置推定　ブロック　AI　走行

走行体がコース上のどこを走っているのか認識する。総走行距離$D_C[mm]$、走行体の向き（回転角度: ω）を算出することで実現する。
式(5.1)により左右のタイヤの走行距離Dを求める。

$D = 2\pi D_W \times \frac{rad}{360}$…式(5.1)

D_W：タイヤの直径[mm]
rad：モーターの回転角

走行体の走行距離D_Cは式(5.2)となる。

$D_C = \frac{(D_R+D_L)}{2}$…式(5.2)

D_R：右タイヤの走行距離
D_L：左タイヤの走行距離

走行体の向き（回転角度ω）は式(5.3)となる。

$\omega = \frac{(D_R-D_L)}{2w} \times \frac{180}{\pi}$…式(5.3)

w：車軸中心からタイヤまでの長さ

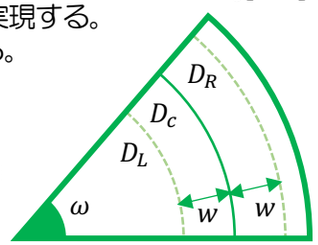


図5.1 自己位置推定

5.2 色彩制御　ブロック　AI

カラーセンサからRGB値で取得し、黒を除くカラーブロックの色を決定する。RGB色空間から、HSV色空間に変換を行い、色相・彩度・明度に分解し色識別を行う。検証で得られたデータの結果、色相のデータに基づき色を決定する。距離が異なっていても得られる色相の値が大きな差は無いため、色相のみ用いる。
パワーブロック(黒)の配置は固定のため、黒色の識別は必要ない。

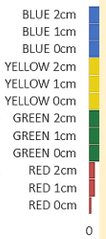


図5.2 色・距離ごとの色相

5.3 ダイクストラ法

・ブロック置場間の経路に重みをかける
・ブロックが配置された場所から伸びる経路は重みをかける
・現在地から目標地点までのルートごとに経路の重みを合計
・合計の重みが最軽量のルートが他のブロックに邪魔されない最短ルート

このサンプルに使っているモデル図は2018年のものです。（AT車限定～Gold License～の協力で流用しています）2019年とは競技内容や審査規約が異なります。どんな内容を記載するのかについては、必ず2019年の規約を確認しましょう。

5.4 数字の特定方法　AI

1. 右出題数字(デジタル数字)の場合

デジタル数字の読み込みを１本の経路で行い、図5.4の通りに区画分けをする。

:スタート位置への経路
:読みとりスタート位置
:読みとり経路
①～⑤:経路ごとの区間分け
:左出題数字(丸数字)準備経路

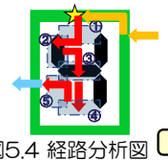


図5.4 経路分析図

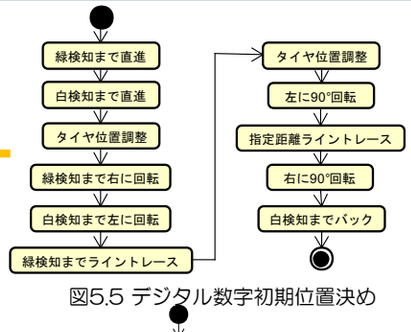


図5.5 デジタル数字初期位置決め

図5.5のアクティビティ図の流れで、デジタル数字読み取りのスタート位置へ移動。図5.6のアクティビティ図の流れで黒線の有無を検知判断し、表5.1のパターンより数字を一意に特定する。

表5.1 各数字の区間ごとの黒線の検知パターン

デジタル数字		0	1	2	3	4	5	6	7
経路1	区画①	○	×	○	○	×	○	○	○
	区画②	○	×	×	×	○	○	○	×
	区画③	×	×	○	○	○	○	○	×
	区画④	○	×	○	○	×	○	○	×
	区画⑤	○	×	○	×	×	×	○	×

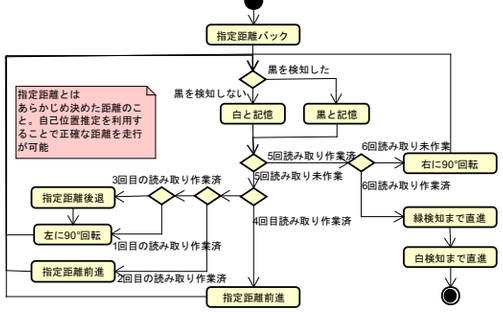


図5.6 デジタル数字読みとり

2. 左出題数字(丸数字)の場合

左出題数字は、図5.7のアクティビティ図の動作を行い読みとり開始位置まで移動する。そして、読みとりラインに沿って走行し区間を記録する。区間とは、白を読んでから黒を読み取るまでの区間と、黒を読みとってから白を読み取るまでの区間をさす(図5.8参照)。区間数と区間の長さを事前に取得したデータと比較することで数字を特定する。表5.2が数字を特定する条件である。

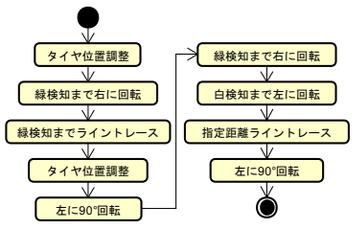


図5.7
丸数字読み取り開始位置移動

表5.2 数字特定条件

区間数	長さ	丸数字
1	-	1
3	区間①が23cm以上	4
	区間②が20cm以上	0
	区間③が18cm以上	7
	区間①②が10cm以上	6
5	区間⑤が20cm以上	5
	区間⑤が10cm以下かつ	2
	区間③が3cm以下	3

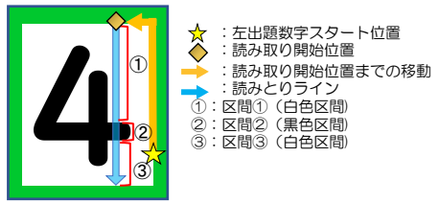


図5.8距離測定パターン例

5.5走行体の回転方法　ブロック

1. ブロック所持時の回転

図5.9のようにタイヤを軸に回転することで、ブロックの力の向きが図5.10のようにアームに押し付ける形になる。このためブロックが回転時に飛び出る可能性を減少させている。

2. 走行体反転方法

カラーブロックを確保した際、次の走行ルートにより走行体を反転させなければならない事態が起こり得る。その場合、その場での回転は使用不可なため、上記の[1.ブロック所持時の回転]方法を組み合わせ反転を行う。（図5.11）

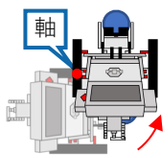


図5.9 90度回転例

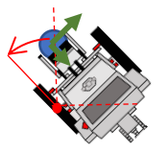


図5.10 回転時の力のモーメント

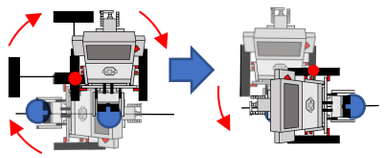


図5.11 走行体反転図

5.6 プレ走行によるパラメータ決定　走行

実際の走行により得られたデータからコースに応じた制御切り替えタイミングを解析し走行のパラメータを決定する。

走行データの傾きが変化した時、制御を切り替えることで、コースの張りによる距離の誤差を最小にする。

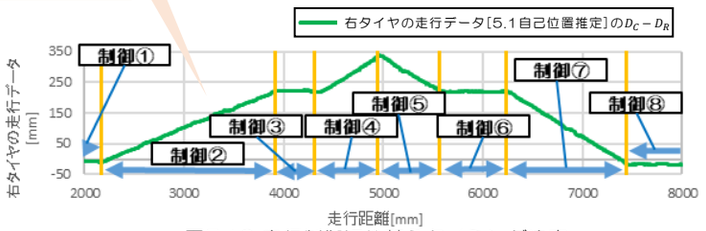


図5.12 走行制御切り替えタイミング決定

5.7 平均化フィルター　AI

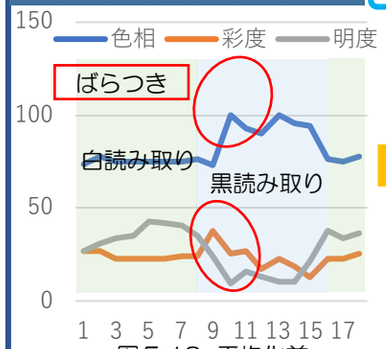


図5.13 平均化前

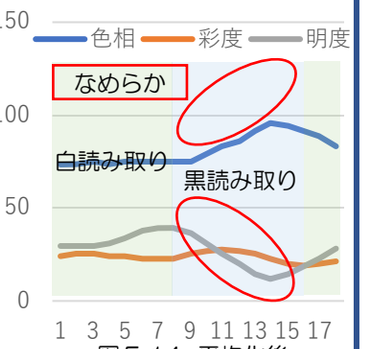


図5.14 平均化後

カラーセンサから色情報を取得する際、走行環境による車体の傾きや、速度の変化による車体のぐらつき、地面の色と色の境界により、取得できる色情報にばらつきが発生し誤検知の要因となる。そこで、取得した値を過去数回の値で平均化し、誤検知を防ぎ、正確な色情報の取得を可能にした。

5.8 SWAP制御　ブロック

SWAP制御のアクティビティ図を図5.15に示す。

SWAP制御実施決定例（図5.16）
走行体はブロックBの色を取得し、目的のブロック置き場を決定する。その場所にブロックがあるとわかった時点では、ブロックBの仮置きを始めない※。ブロックAの色を取得し、入れ替える必要がわかった時点で、アクティビティ図の動作を行う。
※ブロックの入れ替える必要がない場合、ブロックBの仮置きが無駄な動作となるため、それを防ぐ。

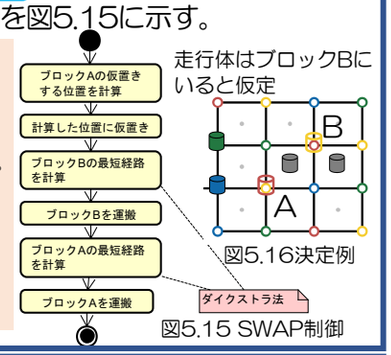


図5.15 SWAP制御

図5.16決定例

5.9ブロック位置の導出方法　ブロック

初期位置コード $XXXXX_{(10)} = B_1 \times 16^4 + B_2 \times 16^3 + B_3 \times 16^2 + B_4 \times 16 + B_P$

16進数に変換
（XXXXXが16未満になるまで16で割る）

【初期位置コード定義】
B_x：ブロック位置番号 (X:1~4)
B_P：パワーブロックパターン

変換後のコード $YYYYY_{(16)} = B_1\ B_2\ B_3\ B_4\ B_P$
この数値は一番高い位から順に B_1, B_2, B_3, B_4, B_P に該当するので、ブロックの配置が特定できる。