

チーム紹介、目標、意気込み

私たち ぱんぱかぱん! は北海道登別市にある日本工学院北海道専門学校の情報処理科 大学編入理系コースの5名で構成された学生チームです!

授業や勉強、バイトなどで時間がない中で合間の時間を使い、このチームで活動しています。

チーム全員の努力が無駄になることなく悔いの残らない走行、設計で優勝を目指します!

モデルの概要

- 各エリアをシナリオとして設定し、それらを随時利用することで、コースを攻略する
- シナリオは走法、区間、判定で構成され、様々なコース形状や課題に対応しやすい構成にした
- ブロック並べエリアについて、論理的な一部を分離し、PCを用いたシミュレーションにも活用可能な構成にした
- Rコース、Lコースに必要な制御を共通に開発し、開発効率を高める設計とした

モデルの構成

1. 要求モデル

- (a) ルール規定やリスクを想定しRコースの目標タイムを設定した
- (b) ユースケース図から機能を抽出した
- (c) 要求図を作りブロック並べではブロック置き場間の移動が必要で、ブロック置き場の入退出、ライトレース、非ライトレースという3つの要素が必要だと導き出された

2. 分析モデル

- (a) 要求モデル(c)で得られた走行体に必要な動作を定義した
- (b) 指針を全てのカラーブロックを確実にパワースポットに有効移動させるとした
- (c) 指針を実現させるためブロックの移動順を全てシミュレーションして最適解を求める
- (d) 解法としてダイクストラ法を採用し、攻略するために要素の定義と手順を決定した

3. 設計モデル

- (a) 分析モデルで得た走行体の動作定義に走法、区間、判定の3つのクラスを定義した
- (b) 走法とは走行体を動かす方法を表すクラスである
- (c) 区間とは走法を活用し、始まりと終わりがある区間を表すクラスである
- (d) 判定は区間の終わりを検出するためのクラスである
- (e) 上記の3つの要素をシーケンス図やクラス図で表した

4. 制御モデル

- (a) ブロック並べ、AIアンサー、どちらでも必要な共通部の3つで制御モデルを記載した
- (b) 共通の制御としてブロック置き場進入の際、走行体の向きがずれてしまうのを回避するために戦略を決定した
- (c) コース全体を走行させるための要素技術について記載した

P2 要求モデル

ぱんぱかばん！

目標 Rコースリザルトタイムー8.0秒！

(I) 目標の検討

以下、区間ごとの小目標を設定し、それらすべて達成することで、目標の達成となる

- ① Rコースで走行タイム20.0秒以内で走破。
- ② Rコースでゴールゲート通過後走行タイム7.0秒以内にブロック並べエリアに到着。
- ③ ブロック並び替えエリアにて、パワーブロックの初期位置の状態での最高得点獲得。

○ Rコースのスピードコース、ゴールゲート通過後の走行を確実に走破するための制約

- ① 黒線に沿って走行する
- ② 他の走行体と接触しない

○ ブロック並べエリアでパワーブロックの初期位置の状態での最高得点獲得するための条件

- ① パワースポットの消失を回避

スピードコース走行タイム 20.0秒

- ② 全ブロックの有効移動成立

ブロック並べエリアボーナスタイム -23.0秒

- ③ 重複するパワースポットに確実に移動させる

駐車ボーナス -5.0秒

-8.0秒

○ 目標設定の根拠

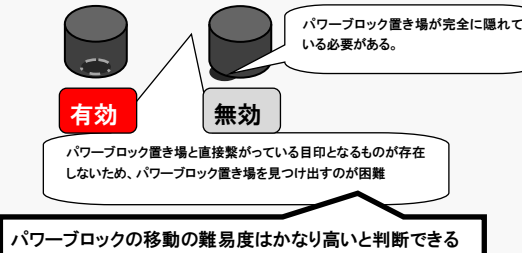
スピードコース走行タイムの設定はRコースのスピードコースの距離と走行体の直進時の速さを求め、次のように設定した。

$$\frac{\text{Rコースのスピードコースの距離(mm)}}{\text{走行体の速さ(mm/s)}} = \frac{12500 \text{ (mm)}}{630 \text{ (mm/s)}} = 19.84126\cdots \text{ (s)} \approx \underline{20 \text{ (s)}}$$

駐車タイムは上記と同じ手段で計測したが、紙面上では割愛する。

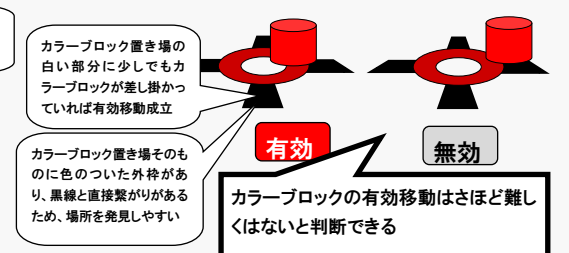
ブロック並べエリアのボーナスタイムを獲得するため、以下のように検討した

○ 有効パワーブロックの条件と移動のリスクについて検討



カラーブロックの有効移動は4つすべてできると判断し、この場合の獲得できる最大のボーナスタイムである23秒に設定。

○ カラーブロックの有効移動成立の条件と移動のリスクについて検討



○ コース区分

Rコースをそれぞれの区間に分け、区間名を設けた。

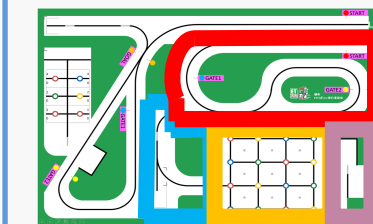


図2-1-1

□ : スピードコース

□ : ゴールゲート通過後エリア

□ : ブロック並べエリア

□ : 直角駐車エリア

(Ⅲ) 要求分析

(I) 目標の検討と、(II) ユースケース分析を基にRコースの各エリアごとと要求分析を行った。また、非機能要求において、品質特性(信頼性、保守性、効率性)について表2-3-1に示した。各要求の凡例を色分けし、可視性の向上を図った。その凡例を左下に示す。

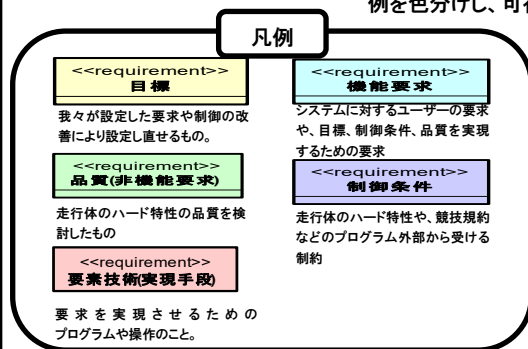
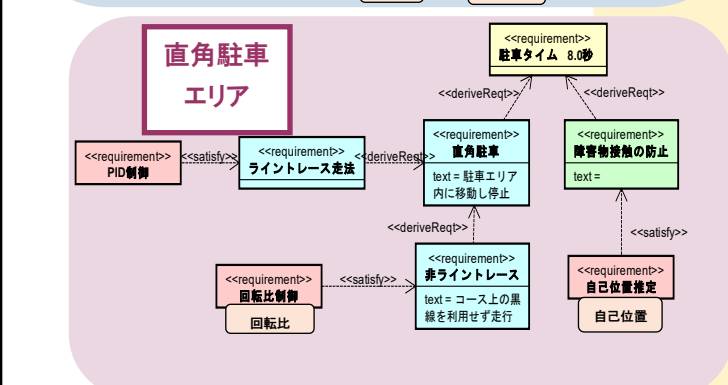
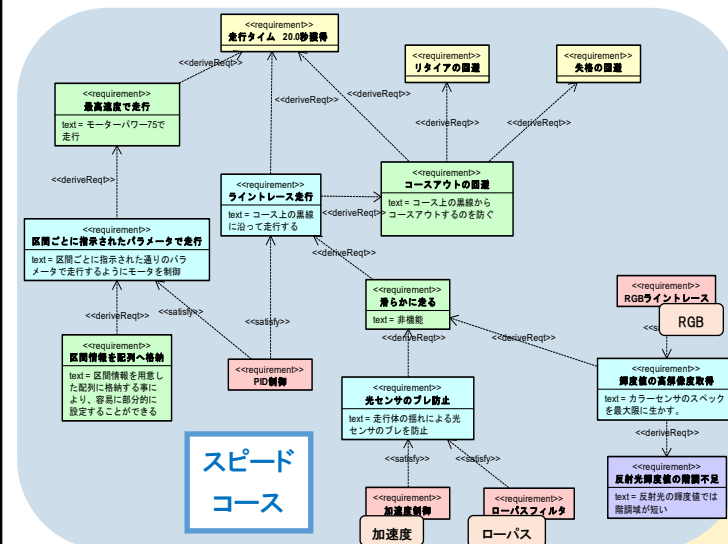
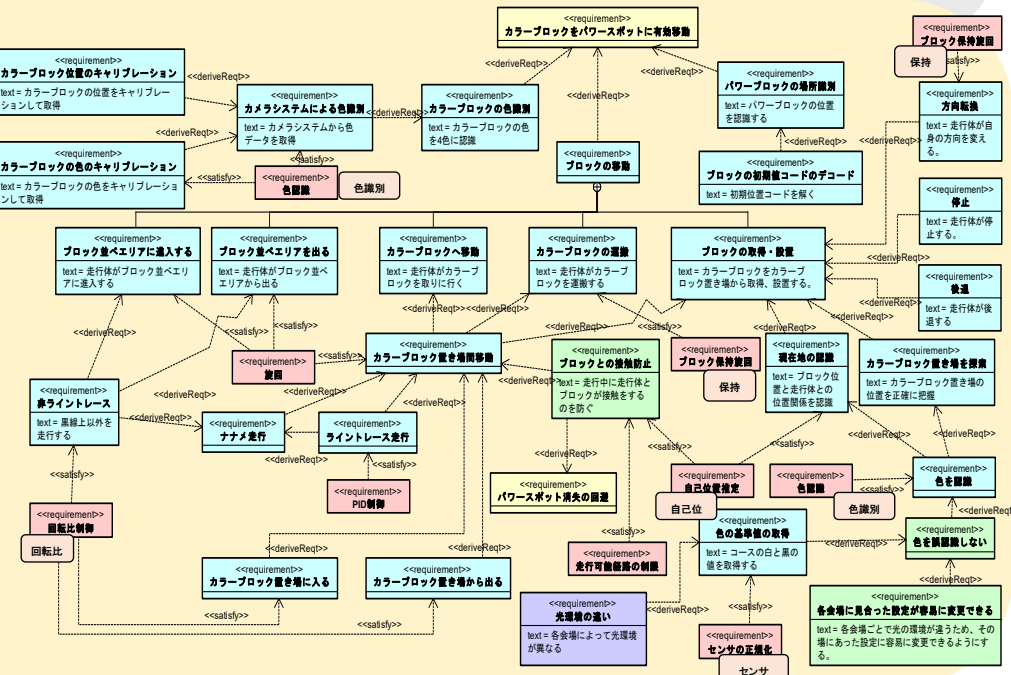


表2-3-1

非機能要求名	信頼性	保守性	効率性
最高速度で走行			○
区間情報を配列に格納		○	
コースアウトの回避	○		
滑らかに走る	○		○
ブロックとの接触防止	○		
色を誤認識しない	○		
色の情報を容易に変更できる		○	
障害物接触の防止	○		



ブロック並べエリア



(II) ユースケース分析

図2-2-1のユースケース図を用いて、アクターを中心として必要な機能の抽出を行う。表2-2-1に示されていない他のユースケース部分は省略している。

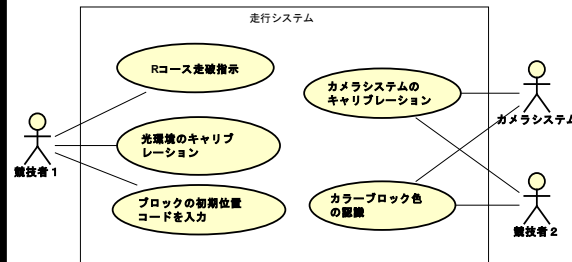


図2-2-1

表2-2-1

ユースケース名	Rコース走破指示	ユースケース名	カメラシステムのキャリブレーション
アクター	競走者1	アクター	カメラシステム、競走者2
概要	走行体の初期化が完了している	概要	ブロック並べに必要なブロックの色を取得
事前条件	競走者がRコースを選択している	事前条件	走行体の初期が終了している
事後条件	システムは終了する	事後条件	システムは終了する
基本系列	1. 競走者はタッチセンサを押してRコースの走破を指示 2. システムはスピードコースを攻略する 3. システムはブロック並べを攻略する 4. システムは直角駐車を攻略する 5. システムは停止する	基本系列	1. ブロック位置の座標を決定する 2. ブロックの色を設定する 3. 初期位置コードを入力する 4. 走行体とのBluetooth接続を開始する

(Ⅳ) 要素技術まとめ

(Ⅲ) 要求分析に示した要素技術を次の表2-4-1にまとめた。

(Ⅲ) 要求分析において、要素技術が示すものを右の表2-4-1に示す。(Ⅲ) 要求分析の、表2-4-1のアイコンに当たる部分と対応付けてあることにより、トレーサビリティを確保している。また、このアイコンは P3 分析モデル 走行体の動作定義、P6 制御モデルにて同一のアイコンを使用している。制御モデルにて一部詳細を記載している。

表2-4-1

要素技術	概要	アイコン
自己位置推定	走行体が自身の座標と方向を推定する	自己位置
回転比制御	直進や曲線走行を制御する	回転比
色識別	4色の色を識別する	色識別
RGBライントレース	解像度の高い輝度値を取得する	RGB
ローパスフィルタ	光センサの雑音を消化する	ローパス
ブロック保持旋回	ブロックが走行体のアームから離れないようにする	保持
センサの正規化	環境に影響されない安定したライントレースができる	センサ
PID制御	黒線からのずれを修正する	
加速度制御	加速度を制御し、走行体のブレを修正する	加速度
走行可能経路の制限	走行可能な経路を制限し、ブロック並べを攻略する	
旋回	走行体の向きを変える	旋回

P3 分析モデル

ぱんぱかぱん！

(Ⅰ)ゲームの要素定義

ブロック並べエリア(図3-1-1全体のエリアを指す)において、図3-1-1のような問題領域(オブジェクト図として図3-1-3)についての分析を行う。例として図3-1-2のようなブロック配置の完成図にすると、一部のオブジェクト図を図3-1-4で表している。これらのゲームの構成要素を、図3-1-3のクラス図で定義した。(図3-1-6は、グラフ全体の線分の長さを表す)

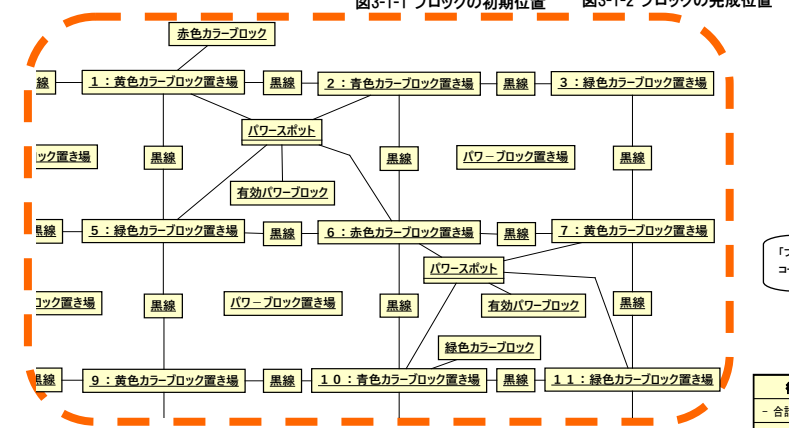
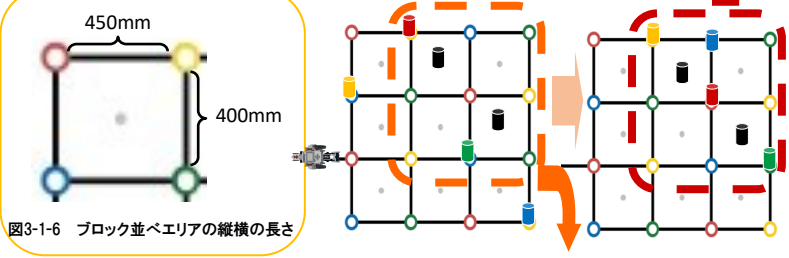
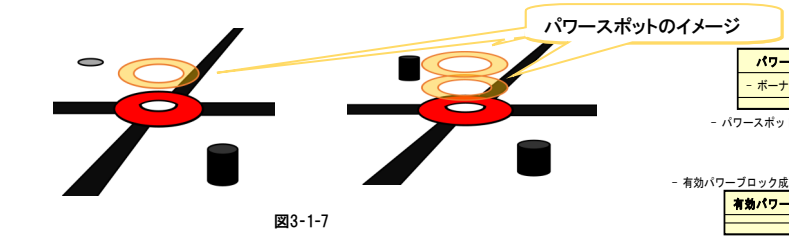


図3-1-3 並べ替え前初期位置のオブジェクト関係の一部(オブジェクト図)

パワースポットが複数重なる、その分パワースポットにおいて獲得できる点数も重複する。例えば、図3-1-7では左方が2×1点、右方が2×2点である。



※1 「カラーブロック置き場」と「パワースポット」でボーナスタイムが変化する。

(Ⅱ)走行体の動作定義

走法

(1)旋回

走行体はその場で旋回をする。左右どちら回りでも可能

複合区間

(8)ナナメ走行

④を用いて45° 旋回→②、⑥を用いて黒線まで直進→④を用いて45° 旋回 の組み合わせで成り立つ。

区間

(4)45° 90° 180° 旋回

①を実行中、45° 90° 180° 角度分旋回した時、

(2)非ライトレース

黒線上を走行しない。前進、後退を行う。

(3)黒線ライトレース

黒線に従って走行する。

(5)色条件で停止

③を実行中、HSV値によって色を判別し、指定の色を判断した時、停止する。

(6)輝度条件で停止

②または③を実行中、輝度値によって白と黒の判別をし、指定の輝度値を判断した時、停止する。

(7)距離条件で停止

②または③を実行中、指定された距離を走行した時停止する。

ブロック配置を完成させるために、まずコース形状や距離、ルールの制約に基づいて走行体の動作を定義する。また、走法と区間を定義し、すべてこれらの組み合わせで走行体は競技する。複合区間はその組み合わせを1つにまとめて思考したものである。

(Ⅲ)指針

・各カラーブロックを1回の運搬で最大得点を獲得！

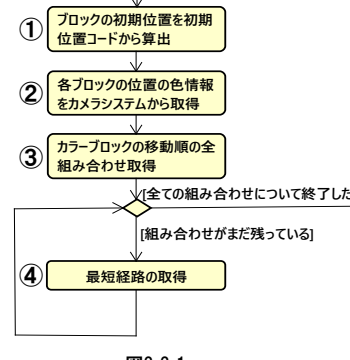
ブロック並べ最短経路取得

指針設定

我々は目標達成のためには、すべてのカラーブロックを確実にパワースポットに有効移動させたい。そのために有効移動不成立のリスクを避け、カラーブロックを一度で移動先位置に移動させる。パワーブロックも同様に移動のリスクを考慮し、パワーブロックを初期位置の状態での最高得点を獲得する。制限時間もあるため上記の指針も考慮しながら最短経路で並び替えを行う。以上の指針を以下のように検討する。

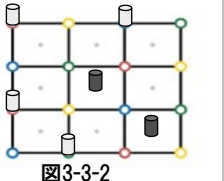
④取得したカラーブロックの移動順を基に最短経路を取得

パワースポットへカラーブロックを移動させる最短経路を取得。最短経路の取得では走行体の向いている方向も考慮する。詳細をP4 (Ⅳ)解法、方向ノードにて記載している



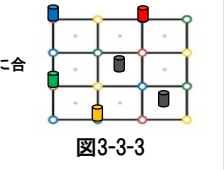
①ブロックの初期位置を初期位置コードから算出

(算出方法に関しては競技規定に基づくため割愛)
算出した各ブロックの位置のイメージを図3-3-2に示す。



②カメラシステムから色情報を取得

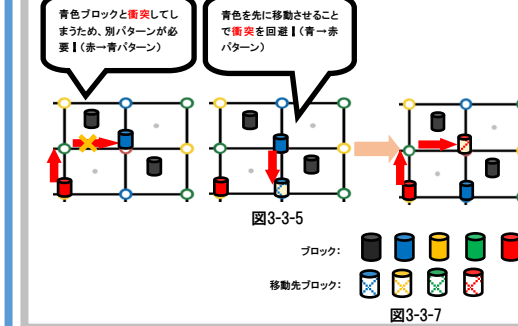
算出した各カラーブロックの位置とその色情報のイメージを図3-3-3に示すように合致させる。ただし、パワーブロックの色情報は取得しない。



③カラーブロックの移動順を全組み合わせを取得。

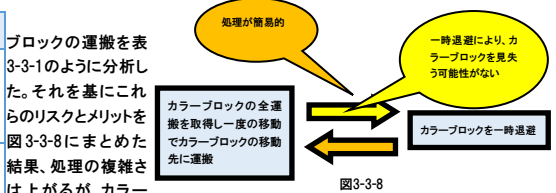
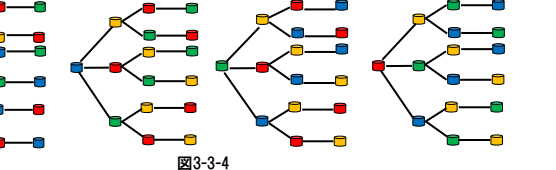
カラーブロックの移動順の全 4! (=24) 通りの組み合わせを取得。以下の図3-3-3にそのイメージを示す。全通り取得することにより、カラーブロックを一時退避させることなくカラーブロックを移動させることができる。(退避をした場合、有効移動したブロックも動かしてしまうというリスクがあるため)

例えば、図3-3-5のように赤色ブロックを移動する先に青色のブロックがあるため移動できない。(赤→青)赤色ブロックと青色ブロックの移動順を変えることで移動することが可能になる(青→赤)。このような場合を見つけるために、全パターンを見つける必要がある。(ブロックの表示は、図3-3-7で表す。)



○運搬方法の分析

表3-3-1		
運搬方法	リスク	メリット
A カラーブロックを一時退避	一時退避したカラーブロックを見失う可能性がある。	処理が簡単。
B カラーブロックの全運搬を取得し一度の移動でカラーブロックの移動先に運搬	カラーブロックの全移動順の中でも置ける組み合わせが無い可能性がある。	一時退避により、カラーブロックを見失う可能性がない。



○カラーブロックの全移動順の中でも置ける組み合わせが無い可能性の回避

表3-3-1のBの運搬方法ではカラーブロックのどの運搬順でも並び替えが出来ないカラーブロックの初期位置のパターンが存在する可能性がある。このリスクを解決すべく、図3-3-9のように進行方向上にカラーブロックが存在していた時、黒線外を走行するナナメ走行を駆使する。ナナメ走行を駆使しない場合とする場合の経路数を以下の図3-3-10のように分析した。

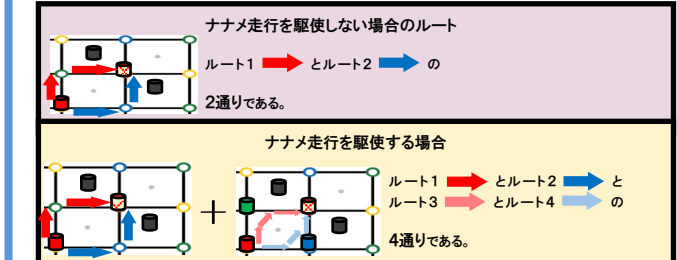


図3-3-10のように、ナナメ走行を駆使した方が経路数が多いため、並び替え不可能なカラーブロックの運搬順になる可能性が低くなる。よって我々はナナメ走行を駆使することにした。

ブロック並べで高得点の獲得

ブロック並べで高得点を取るために、すべてのカラーブロックを有効移動かつパワースポットに移動する。そのために各カラーブロック、パワースポットに点数をつけ一番多い点数のところにブロックを配置することについて表3-3-2のように検討した。

表3-3-2			
パワースポット重複数	メリット	デメリット	取り得る最高得点
A パワースポットの重複1ヶ所(初期位置)	パワースポットを移動させる必要がない	最高得点を取ることが出来ない。	・カラーブロックの有効移動4個→13点 ・パワースポット設置2点×5個→10点 計→23点
B パワースポットの重複2ヶ所	重複するポイントが増え、最高得点が狙える	パワーブロックを移動させる必要があるため、パワースポットが消費する可能性がある。	・カラーブロックの有効移動4個→13点 ・パワースポット設置2点×6個→12点 計→25点

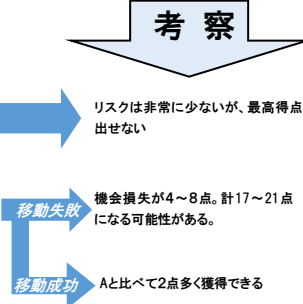


表3-3-2を考察してみるとBは移動したときの機械損失がAと比べて4～8点と大きく、成功した際の獲得点は2点とハイリスクな割にリターンが少ないである。また、移動成功の難易度を図3-3-11のように分析したところ、黒線ライトレースが得意でない上に、有効パワーブロック不成立のリスクが非常に高いと分析。よって我々はブロック並べで最高得点獲得することを妥協し、パワーブロックを初期位置の状態で競技することとした。

図3-3-11

P4 分析モデル

(Ⅳ)解法

ブロックの移動順をすべてシミュレーションして最適解を求める

○方向ノード

方向ノードは、走行体がカーブブロック置き場に侵入してくる方向を考えたものである。そのため図3-4-2の1～4のように、各カーブブロック置き場に4個ずつ存在することになる。そのため、方向ノードの数はカーブブロック置き場の数×4個存在する。

図3-4-2

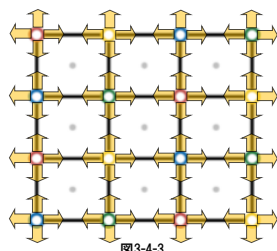


図3-4-3

② 取得した図3-4-6の運搬順序を基に通過可能なものの中で最短ものを取得

例として、図3-4-7のようなカーブブロックの図3-4-7に示す、ナメ走行を用いた経路が最短と考えられる。

説明:

図3-4-7

③ブロックの運搬決定

カラーブロックの移動を行う際に、“コスト計算”を行う。コスト計算を行い、総コストが最も低いものにカラーブロックの移動先を決定。同じカラーブロック置き場に移動する場合でも方向ノードによりコストが変わる。ここで、すべての方向ノードへのコストを一度の計算で算出することができるダイクストラ法を利用する。

○コスト計算

我々はそれぞれの動作に次の表3-4-1のようにコストを設定。コストの値は所要時間を基に設定してある。

動作	コスト値(×10 ² 秒)
横方向直進	20
縦方向直進	23
90° 旋回	30
180° 旋回	45
ナメ走行	60

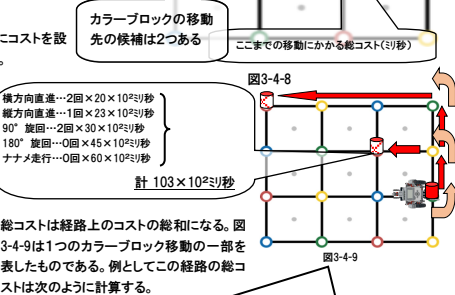
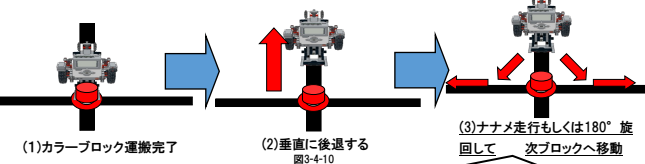


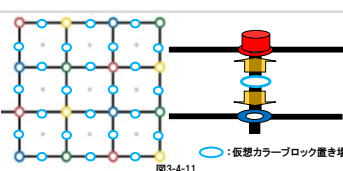
図3-4-8

図3-4-9

○カラーブロックの運搬後、次のカ



○仮想カラーブロック置き場



上記の(2)垂直に後退するの、後、進路を決定する際、“仮想カラーブロック置き場”を利用する。これは我々が独自に設定した仮想のカラーブロック置き場であり、図3-4-11のように各カラーブロック置き場の間に存在し、それぞれ方向ノードを2つ持つ。

○後方へ進行
ナメに移動した先にカラーブロックが存在する場合または迂回するルートよりも後方へ移動した方がコストが低い時、図3-4-13のように180° 旋回を用いて後方へ移動する。

○前方へ進行

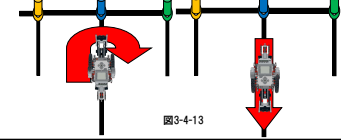
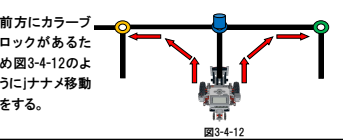


図3-4-12

図3-4-13

※ ○ は設計モデルにも記載しており、関連があることを表す。

指針にしたがって問題を解く。指針を基にブロック並べを解くクラス図を図3-4-1を作成した。

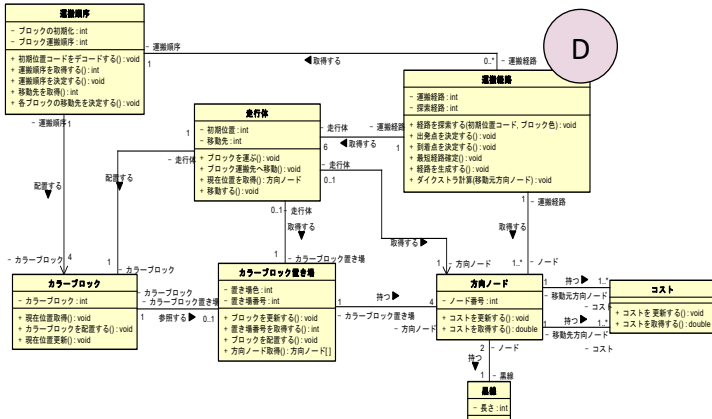


図3-4-1

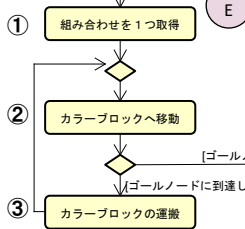


図3-4-4

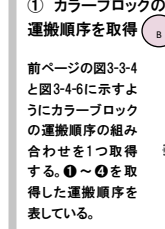


図3-4-6

○走行体の進行経路

上記のナメ走行、ブロック並べエリアの進入、退出の経路を基に走行体を通りうる経路を導出した。それを図3-4-14に示す。ここで、進入地点、退出地点に仮想カラーブロック置き場を設置し、それぞれスタート仮想カラーブロック置き場、ゴール仮想カラーブロック置き場とする。

○スタート仮想カラーブロック置き場



図3-4-15

○ゴール仮想カラーブロック置き場

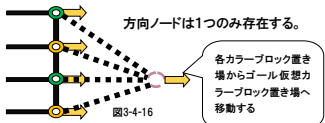


図3-4-16

○ナメ走行の移動経路

ナメ走行では次の図3-4-17のような経路を通る。

ナメ走行は黒線ラインレスを行わずに進行する。

..... : 走行体が行進し得る黒線外の進行経路

図3-4-17

以上の内容を踏まえた結果のコミュニケーション図を次の図3-4-18に示す。初期位置から運搬経路を取

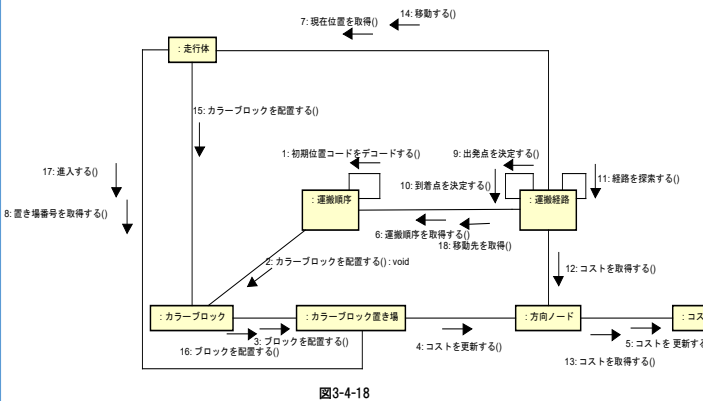


図3-4-18

設計モデル

ぱんぱかぱん！

(Ⅰ)設計意図・方針

- 走法、区間、判定の3つのクラスを定義し、変更や追加を容易にする
- Lコース、Rコースに必要な基本的な制御を共通に準備し、開発効率を上げる
- 様々な判定をパッケージ化、共通化し、各クラスから容易に利用可能にする
- ブロック並べの論理的な機能は独立して開発可能とし、PCを用いたシミュレーションからも利用可能にする(図4-1-1に示す)

開発言語：Visual C#

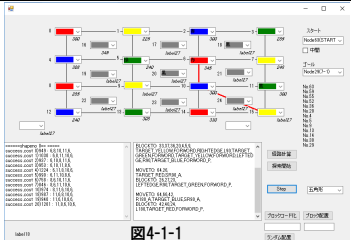


図4-1-1

(Ⅱ)振る舞い

○メイン処理

Rコースのメイン処理のシーケンス図を図4-2-1に表す。多くのシナリオには事前に設定されている。ただし、ブロック並べエリアの初期位置は、キャリブレーション時にユーザーが初期位置コードを入力することでブロック位置を取得している。今回は、カメラシステムでブロック色を取得するため、タッチセンサー押下までブロック色はわからない。これらのことから、ブロック並べ進入の時点で行うことにする。また、図4-2-1をもとに走行体と

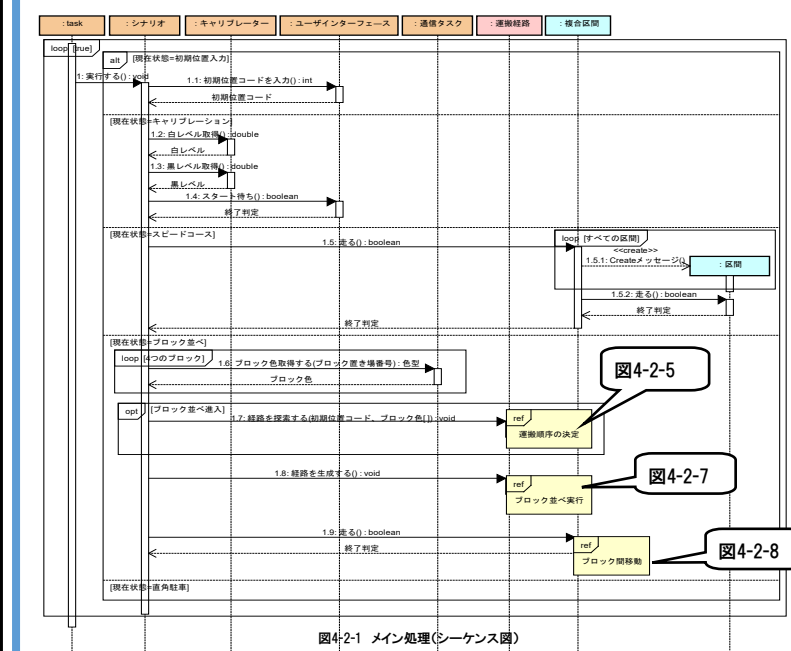


図4-2-1 メイン処理(シーケンス図)

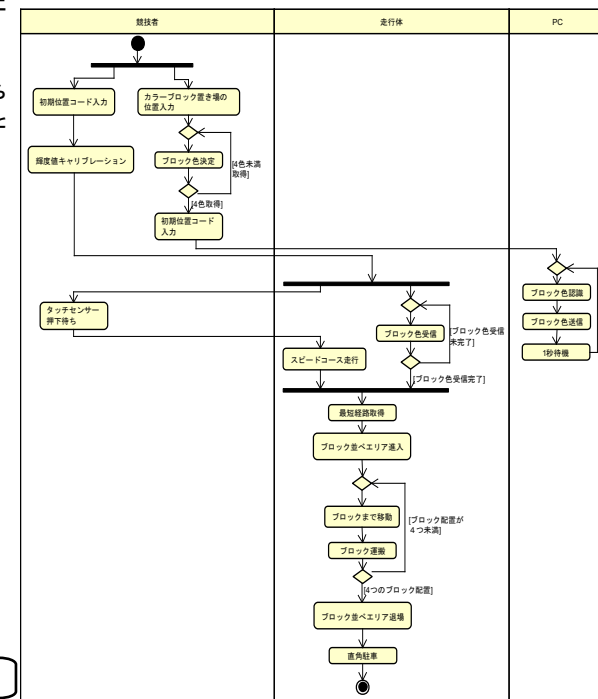


図4-2-2 コース走破全体(アクティビティ図)

○カメラシステムとの連携

ブロック色を取得するためのカメラシステムからの画像を解析するアクティビティ図を図4-2-3で示す。キャリブレーションによりブロックの位置を取り込み色の認識を継続して行う。

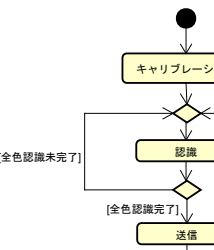


図4-2-3 カメラシステム

○通信タスク

ブロック色取得のための、通信タスクについてのステートマシン図を図4-2-4に示す。ブロック色取得のためにカメラシステムから受信したデータの解析の様子を表した。

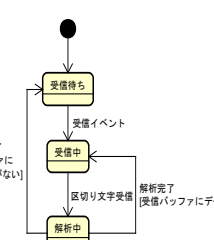


図4-2-4 通信タスク

○運搬順序決定

P3(Ⅲ)指針 運搬順序を決定する際の詳細を次の図4-2-6に示す。図3-4-4より実装方法を検討した。

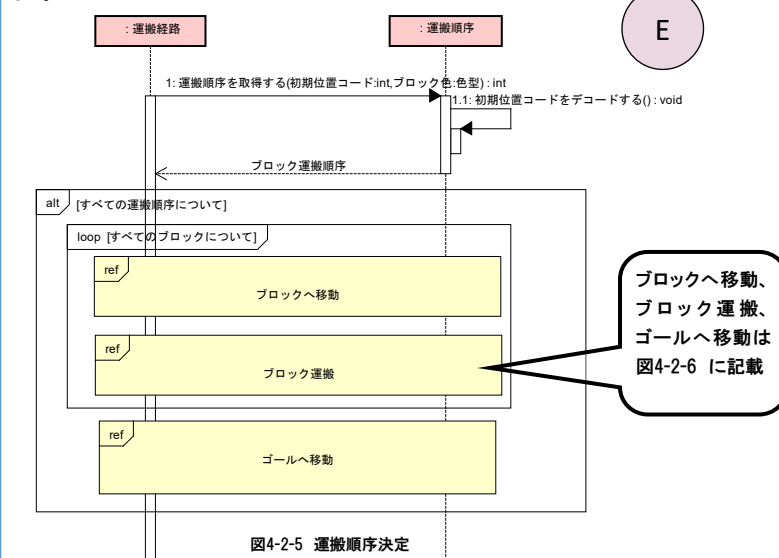


図4-2-6 運搬順序決定

P5 設計モデル

ぱんぱかぱん！

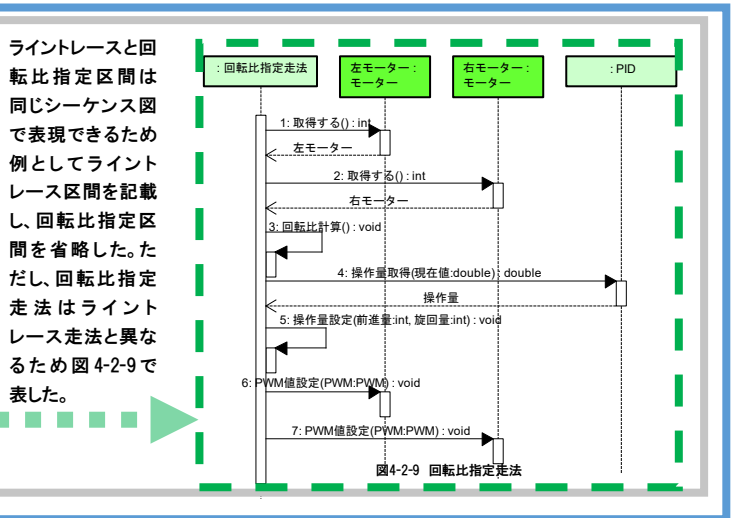
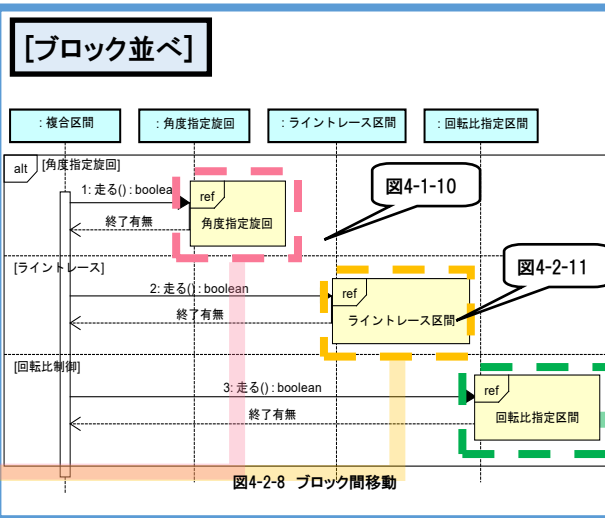
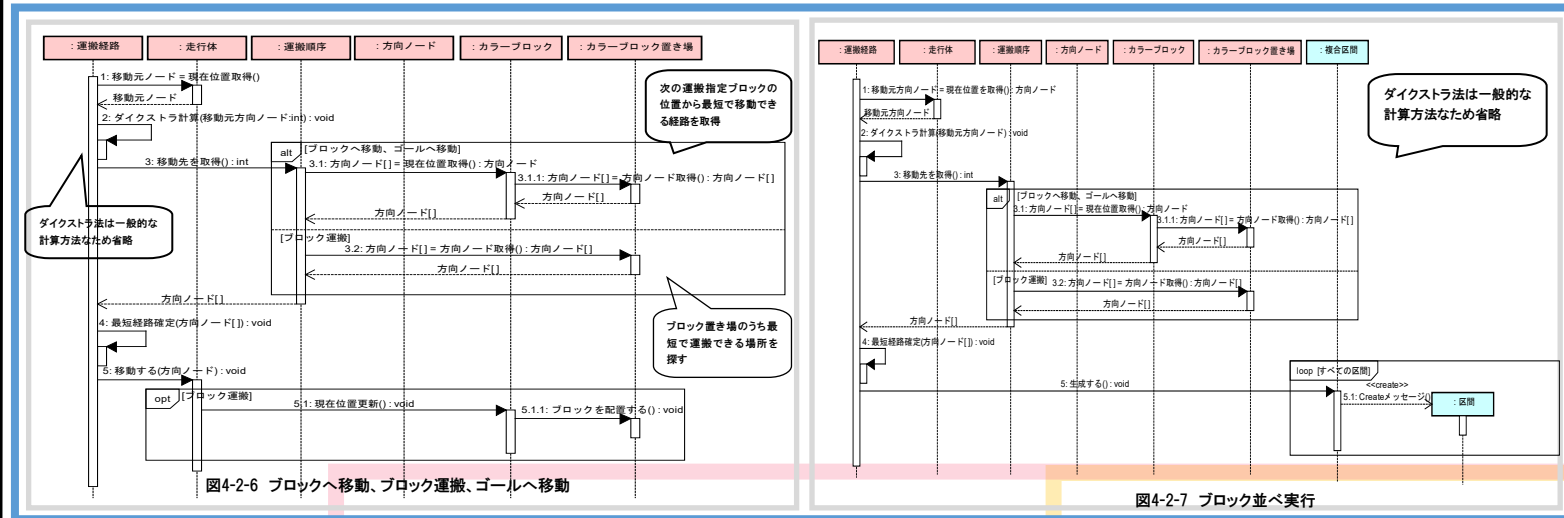
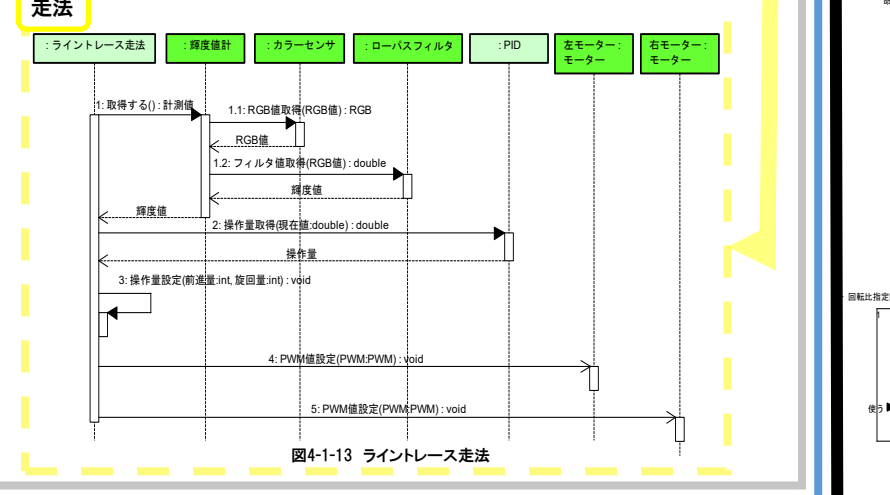
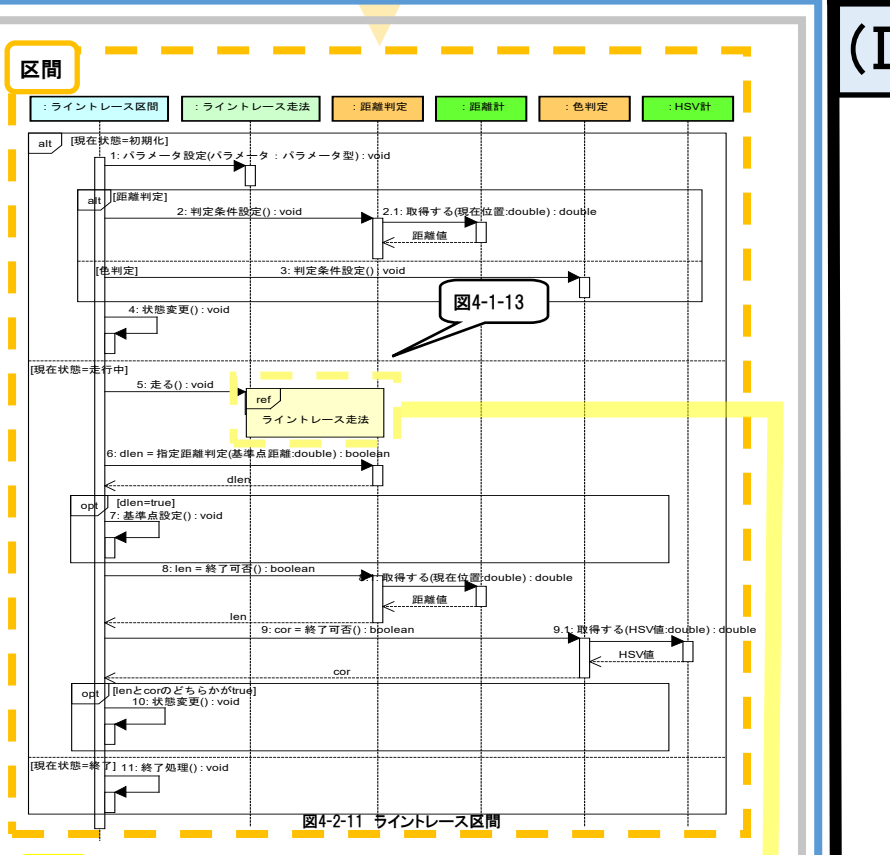
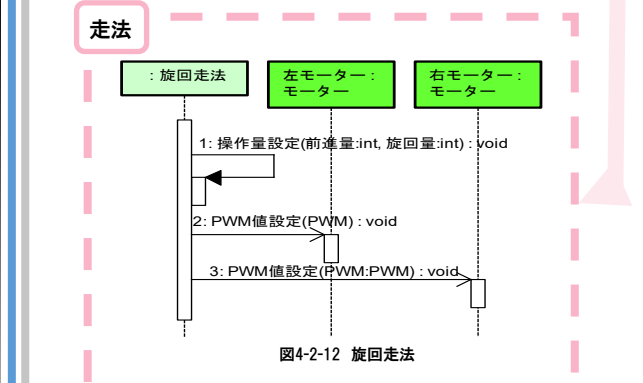
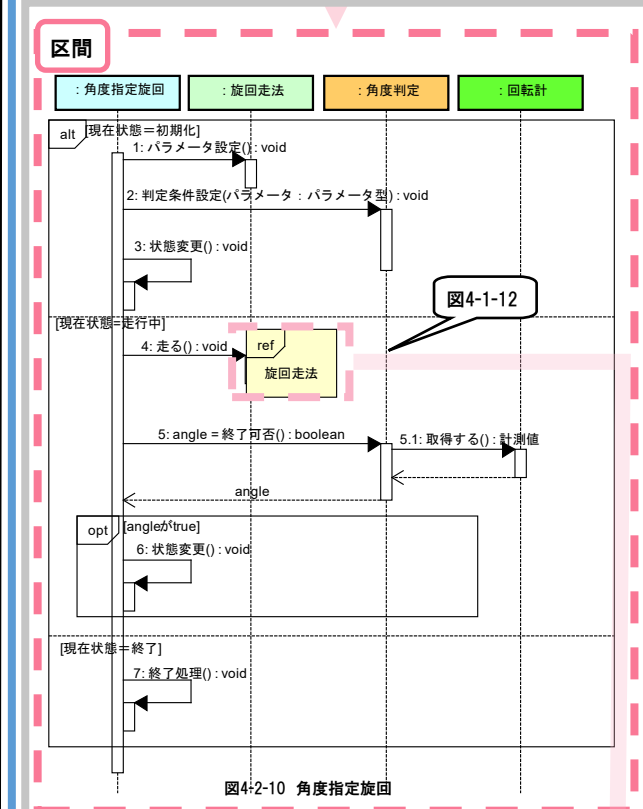
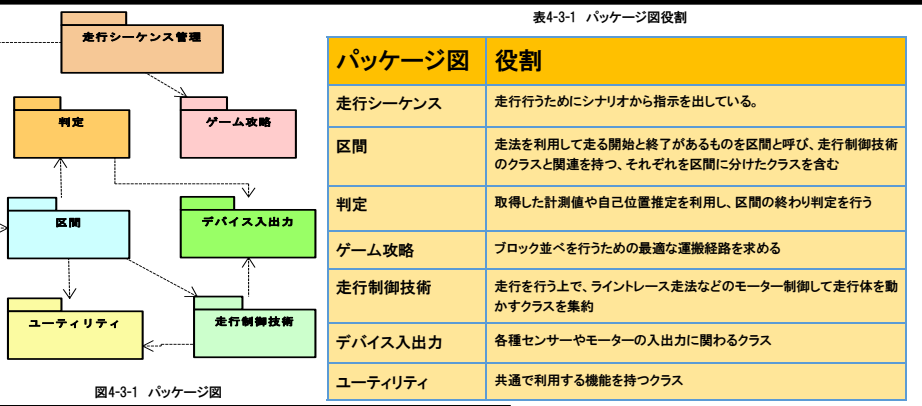


図4-2-8の角度指定旋回は、図4-2-10、図4-2-12で表現した。また、ライトレース区間は、図4-2-11、図4-2-13で表現した。回転比指定区間では、上で記述した通りである

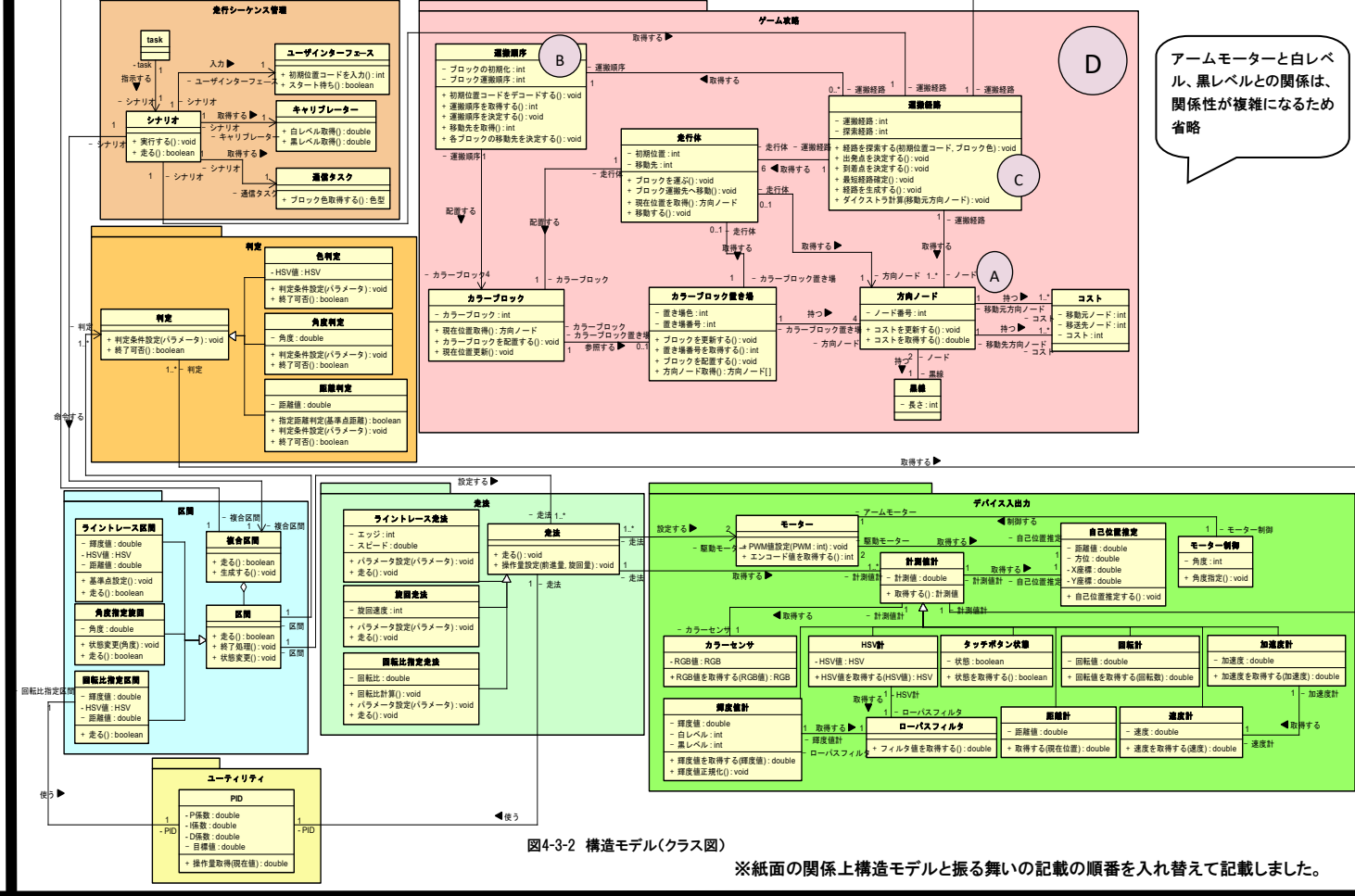


(Ⅲ)構造

分析もとに図4-3-1のようなパッケージ図を作成した。その役割として表4-3-1の表で表した。それらをもとに図4-3-2のクラス図を作成した。図4-2-2では、ユーザーインターフェイスとタッチボタンの繋がりが持つが煩雑になるため省略した。



パッケージ図	役割
走行シーケンス	走行を行うためにシナリオから指示を出している。
区間	走法を利用して開始と終了があるものを区間と呼び、走行制御技術のクラスと関連を持つ、それぞれを区間に分けたクラスを含む
判定	取得した計測値や自己位置推定を利用し、区間の終わり判定を行う
ゲーム攻略	ブロック並べを行うための最適な運搬経路を求める
走行制御技術	走行を行う上で、ライトレース走法などのモーター制御して走行体を動かすクラスを集約
デバイス入出力	各種センサーやモーターの入出力に関わるクラス
ユーティリティ	共通で利用する機能を持つクラス



※紙面の関係上構造モデルと振る舞いの記載の順番を入れ替えて記載しました。

