

チーム紹介、目標、意気込み

チーム紹介

RAT記は仙台高等専門学校の研究室メンバーを中心に構成されたチームです。今年はメンバーを大幅に増やし、4、5年生を中心に活動してきました。

目標

試走会のタイム「22.5秒」を更新する

意気込み

過度な無理はせずに、チームで協力しあって活動したい。

モデルの概要

- 要求：チームの目標を細分化し、目標を実現するための要求を導いた。
- 分析：現在の前提条件を確認するために、コースの状態を分析した。（ゲーム部分のみ）
- 設計：要求と分析から、プログラムを設計した。
- 制御：走行体のセンサやモータを制御している技術について記載した。
- （文字サイズは16ポイント以上、全角で300文字程度）

モデルの構成

1. 要求分析

- ダミーテキストダミーテキストダミーテキストダミーテキストダミーテキストダミーテキストダミーテキストダミーテキスト
- ダミーテキストダミーテキスト
- ダミーテキストダミーテキスト

2. 分析モデル

- ダミーテキストダミーテキストダミーテキストダミーテキストダミーテキストダミーテキストダミーテキストダミーテキスト
- ダミーテキストダミーテキスト
- ダミーテキストダミーテキスト
- （文字サイズは16ポイント以上）

ここに書いた説明で、どのように分析設計が進められ、分析に何が書いてあるか、設計の何が書いてあるか、制御として何に取り組んでいるか、それらがどのようにつながっているか、といったことが把握できる

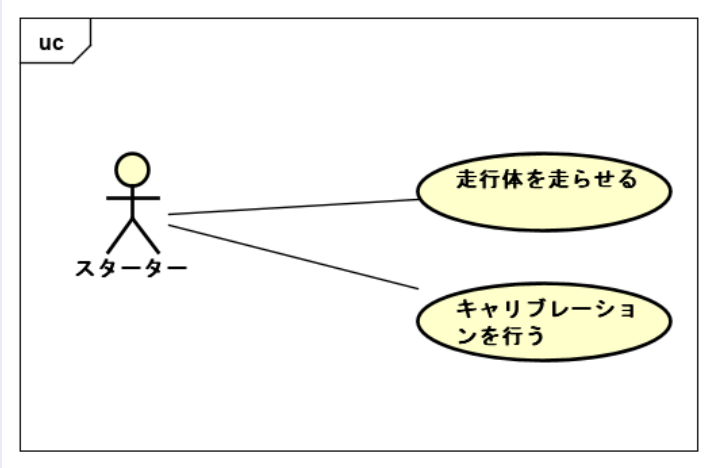
1. 要求モデル

目標

試走会で安全に走行した際の走行タイム「22.5秒」を更新し、最大のボーナスタイムを獲得する。

1.2 ユースケース

アクターに提供する価値に注目してユースケースを抽出した。



ユースケース記述

走行体を走らせるユースケースについて記述した。

| 項目 | 内容 |
|----------|------------------------------------------------------------|
| ユースケース | 走行体を走らせる |
| 概要 | スターターがプログラムを実行することで走行体が起動する。走行体はコースの走行、ゲームの回答、ガレージへの停止を行う。 |
| アクター | スターター |
| 事前条件 | 走行日、もしくは試走会日である。 |
| 事後条件 | 走行が終了する 競技が終了する |
| 基本系列 | タッチセンサを押す コースを走行する ゲームを解く ガレージに止まる |
| 代替系列 | NULL |
| 例外系列 | フライングスタートをする 走行体が完全停止する 失格になる リタイアする |
| サブユースケース | |
| 備考 | |

要求図とユースケース記述の対応

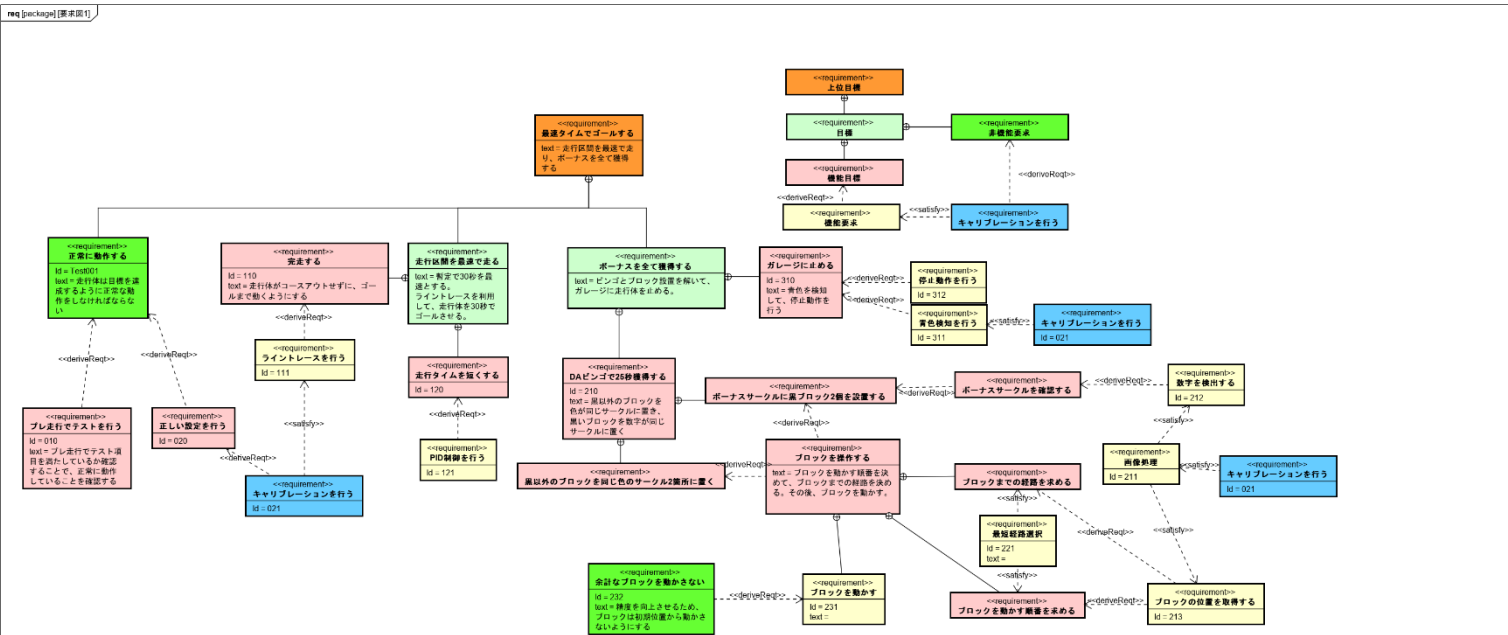
要求のIDとユースケース記述の基本系列は対応している。

| ID | 記述 |
|-----|----------|
| 0xx | 非機能要件 |
| 1xx | コースを走行する |
| 2xx | ゲームを解く |
| 3xx | ガレージに止まる |

0xxは信頼性などの非機能要件に関するIDとした。
Xには任意の自然数がひとつ当てはまる。

機能要求の検討

走行区間とゲーム区間に分け、目標を細分化し、ユースケース記述と関連付けながら、機能要求を抽出した。※動作はIDが小さいほうから順に実行される



テスト項目の検討

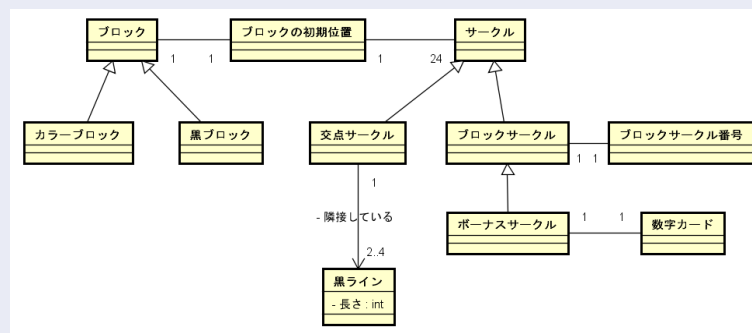
目標達成に必要なテスト項目を実際に動作を確認する場合の
手順から抽出した。

ブロックの初期位置が取得できている
数字が取得できている
コースを外れていない
2分以内にゴールしている
ゲームを完全回答できている
ガレージに停車できる

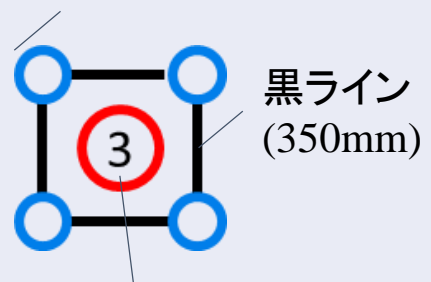
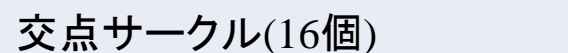
2. 分析モデル

2.1 ゲームの要件定義

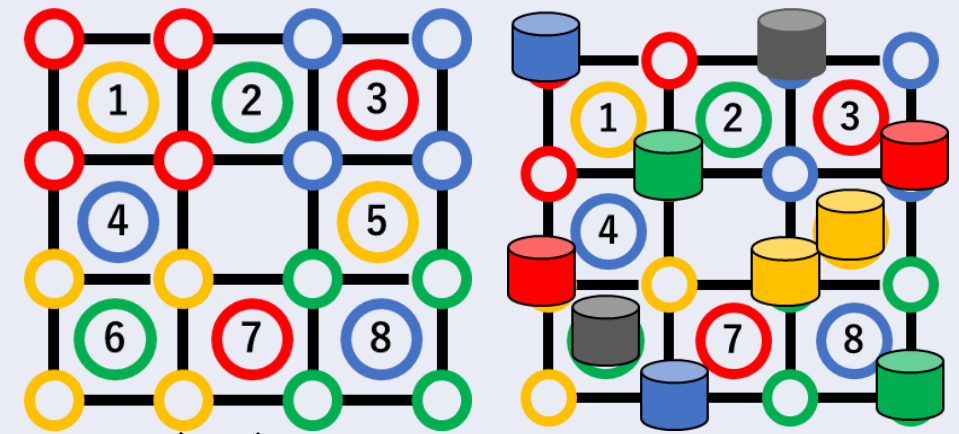
ゲームには赤・青・緑・黄色の計8個のカラーブロックと2個の黒ブロックがあり、黒ブロックを数字カードによって決まるボーナスサークルの中に運ぶことでボーナスタイムを得られる。またそれぞれのカラーブロックを色に対応するブロックサークルの中に運ぶことでボーナスタイムを得られる。初期のブロックの配置は8個のブロック（カラーブロック7個＋黒ブロック1個）が交点サークルの決まった場所に置かれる。残りの黒ブロック1個はボーナスサークル以外のところに置かれる。そして1個のカラーブロックはどちらかの同色のブロックサークルに置かれる。



ゲームの構成



ブロックサークル(1~8)



ビンゴコース

初期配置例

2.2 走行体の動作定義

ゲームを解くうえで前提となる走行体の動作を表に示す。

| 動作 | 課題 | 解決方法 |
|------------------|------------------------------------|--------|
| ブロックを保持して回転 | ブロックをこぼさずに回転する方法 | 5.2 |
| ブロックサークルにブロックを置く | ブロックサークルの中に入れる方法 | 3.1 |
| ブロックの位置と色を把握 | ブロックの初期位置と各色の取得 | カメラで取得 |
| 走行体が移動する経路を算出 | 最短経路の算出 | 2.3 |
| | 移動する途中に通るブロックサークルにブロックがあった場合走行できない | |
| 数字判定 | 具体的な数字の判定方法 | |

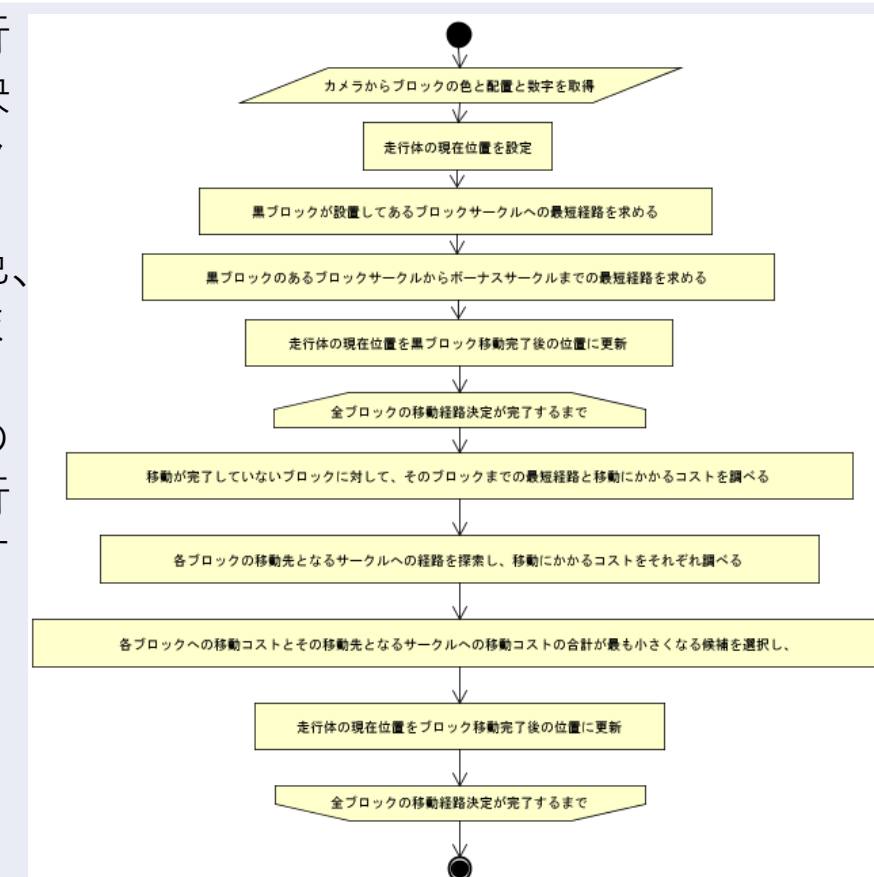
2.3 指針

ビンゴにおける走行体の走行ルートの決定方法をフローチャートに表す。

ブロックの配置や色、数字はレースが始まる直前にわかる。

よって走行ルートの決定は走行中に走行体で計算され決定するものとする。

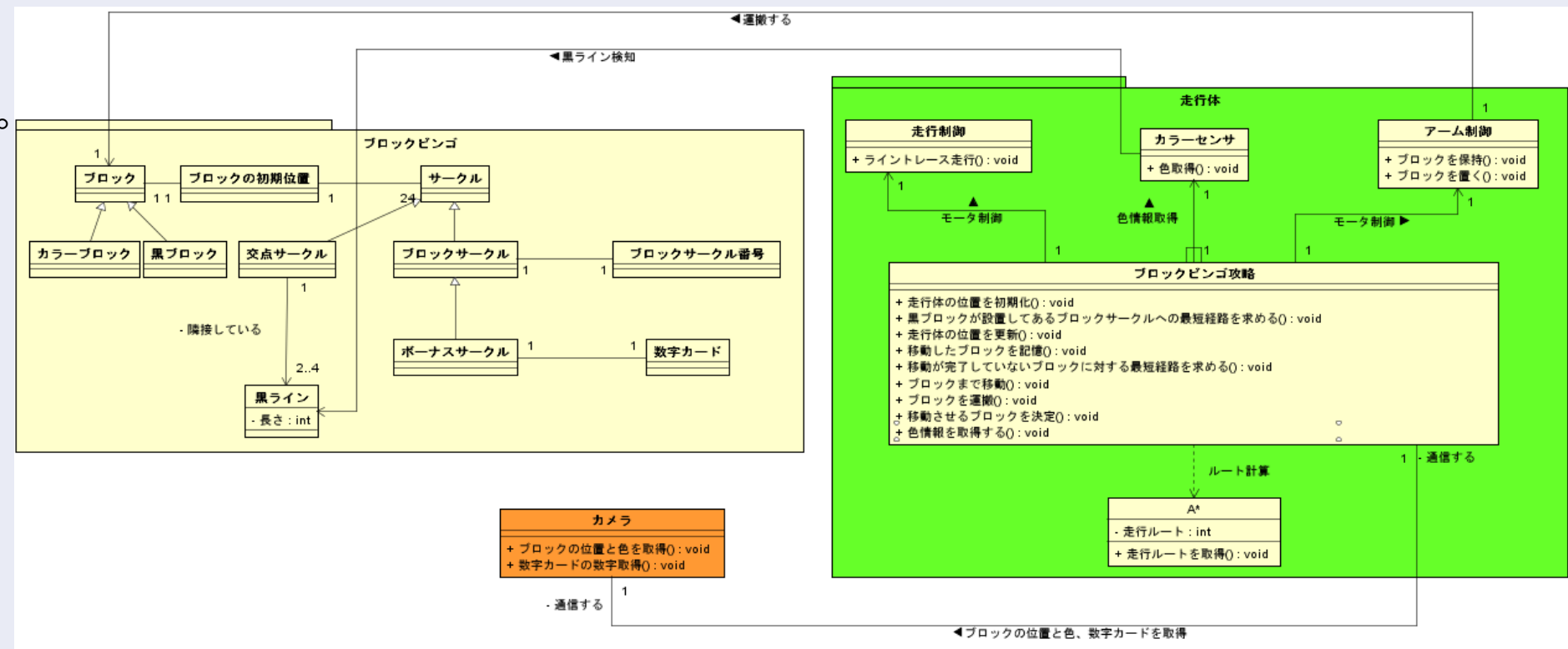
黒ブロックを先にとる理由の記述



3.分析モデル、設計モデル

3.1 解法

指針をもとにブロックビンゴを攻略するクラス図を作成した。最短経路を求めるためにA*法を用いる。A*法により各ブロックの最短経路を求め、効率的にブロックビンゴを攻略する。



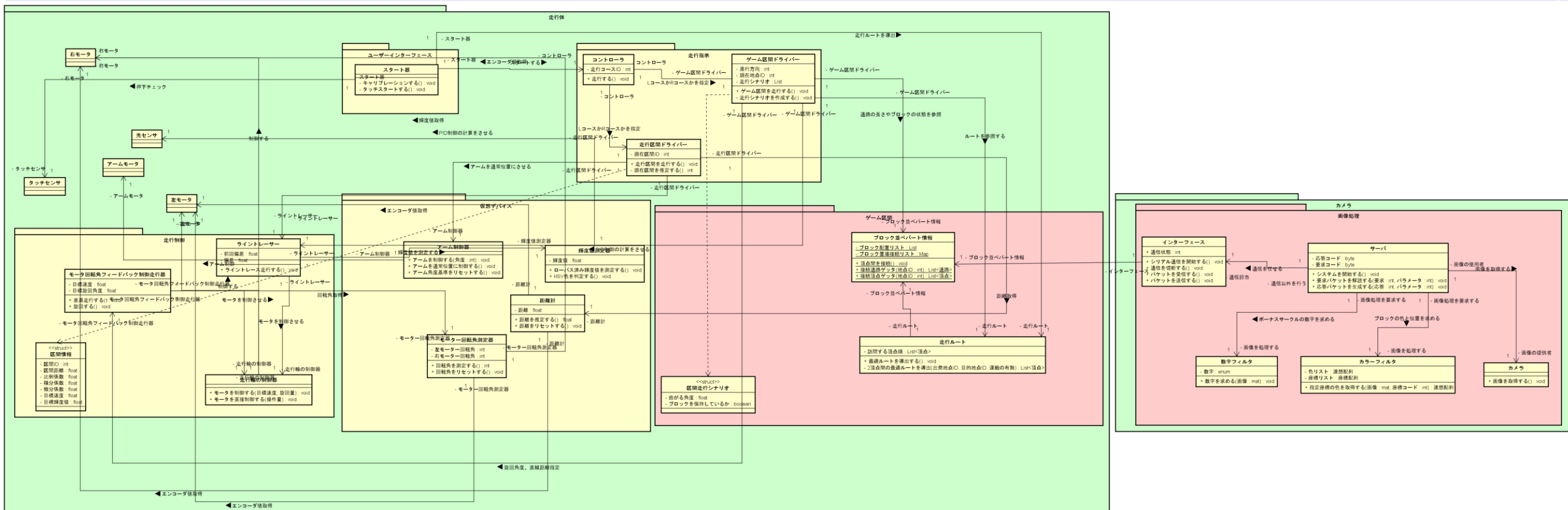
-----ここまで分析モデル、ここから設計モデル-----

3.2 設計意図、方針

設計の意図、方針をパッケージ構成等で示す。

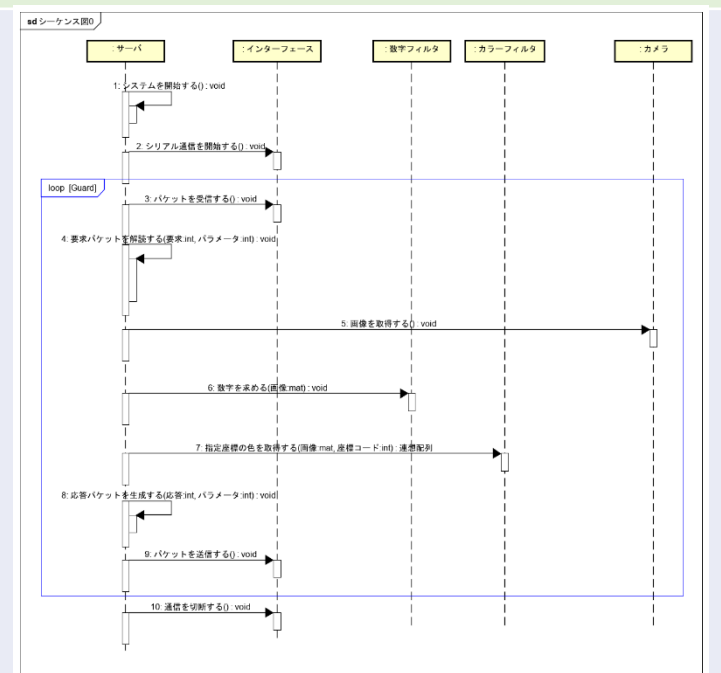
設計モデル

構造



振る舞い

シーケンス図



5.制御モデル

5.1 輝度値補正

環境ごとの差異を少なくするために、サンプリングした値をもとにして輝度値正規化を行う。

正規化後輝度値センサ値=(輝度値センサの値－黒の基準値)/(白の基準値－黒の基準値)

また、輝度値センサの値にはノイズが含まれているため、差分方程式より以下のローパスフィルタを用い、ノイズの影響を低減する。

$$y_t = a * y_{(t-1)} + (1 - a) * x$$

y_t = t における出力値 a = 係数 (0.8) x = 入力値

結果、輝度値は0~1の間に収まり、ローパスフィルタを用いたほうがノイズが少なくなった



5.2 ブロック保持

ブロックを保持している状態で、どのように回転すればブロックを保持して回転できるか検証した。

ブロックを持った状態で一回転した時、ブロックを保持し続けられるかを、左右のモータの回転速度を変えて検証した。検証は回転速度ごとに3回行った。検証結果を図に示す。

検証結果から、片方のモータを速度64rpm,逆のモータを速度-32rpmにした時、最も早く安全に回転できることがわかる。

| 回転速度(rpm) | | 結果 |
|-----------|-----|-----|
| 左輪 | 右輪 | |
| 50 | -25 | ○○○ |
| 60 | -30 | ○○○ |
| 64 | -32 | ○○○ |
| 66 | -33 | ○○× |
| 68 | -34 | ○×○ |
| 70 | -35 | ○×× |
| 72 | -36 | ××× |
| 76 | -38 | ××× |
| 80 | -40 | ××× |

5.3 ライントレース

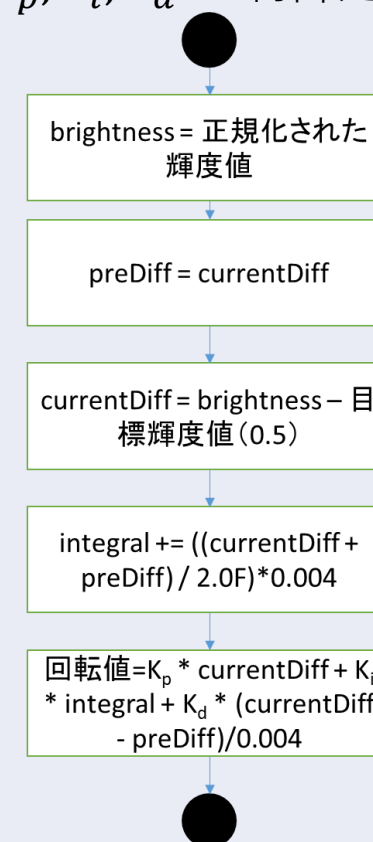
正確なライントレースを行うために、現在の輝度値と目標値の差を用いて、PID制御を行う。

$$f = x_t * Kp + K_i \int_0^t x_t dt + (x_t - x_{t-1}) * Kd$$

x_t = 時刻 t での輝度値と目標値の差

K_p, K_i, K_d = 制御定数

実装方法



また、走行する区間によって、制御定数を変化させることによって、区間に合わせた適切な制御を実現する。制御定数は走行させた結果によって値を変える。例えば、カーブを曲がり切れない場合は K_p を増やし、カクツキが大きい場合は K_p を減らす。

数字認識

Keras と MNIST を利用して、深層学習を行い、数字認識を行うモデルを作成する。走行体から要求があったときに作成したモデルを利用して数字の判定を行い、ボーナスサークルの番号として数字を走行体に返信する。以下に詳細を示す。

モデルの作成

モデルの作成にはkeras（ライブラリ）とMNIST（データセット）を利用する。今回は画像による認識なので、畳み込みニューラルネットワークのモデル構造を作り、MNISTのデータで学習を進める。バッチサイズは200,エポック数は10、活性化関数は出力層ではsoftmax、それ以外ではReLUを使用した。

テスト用データ1万個で性能を確かめたところ正解率95%であったため、このモデルを十分利用可能と判断し、使用した。

モデルの利用（数字の判別）

モデルを利用するためにはモデルに渡す画像を学習時のものと同じように処理しておく必要がある。事前に行う処理を図1に、処理を行う前後の画像の例を図2,図3に示す。

処理を行ったあとの画像をモデルに渡すことで数字を判別することができる。

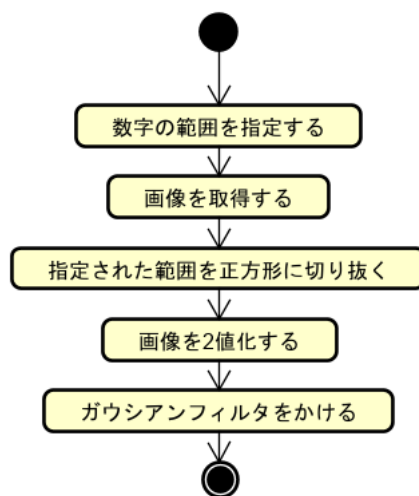


図1



図2

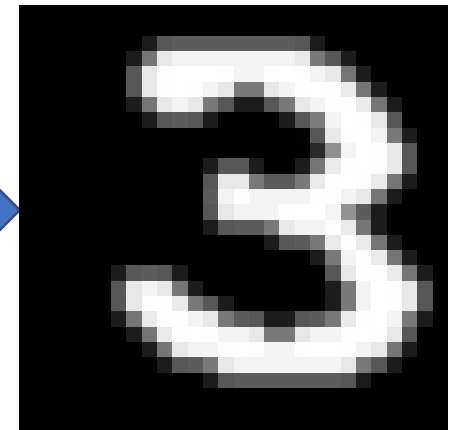


図3

