

デベロッパー部門 アドバンストクラス

地区:北陸 地域:石川県 野々市市



所属:金沢工業大学 夢考房

チーム紹介,目標,意気込み

個性豊かな大学3年生の4人からなるチームで,チーム名である「蝉丸」の由来は百人一首から(らしい)です.

(この名前の本当の意味は誰も知りません.)

「夢考房」とは学生が自主的にものづくりを行う施設であり、その中で我々が所属する「夢考房プロジェクト」は学生がスケジュール管理や組織運営を自主的に行う創作グループです.

チーム3回目の出場で悲願のチャンピオンシップ大会!! ETロボコンに捧げた時間はどのチームにも負けません. 「学生らしさ」が私たちの強みです.

目指すはひとつ,全国制覇!!

モデルの概要

- ・本システムは「走行システム」と「ブロック情報算出システム」の二つのシステムを連動させる。各システムの 責務分担や,通信によるシステム間の通信内容を端的にま とめた
- ・本システムは、想定外の事故(Bluetooth通信不具合,ブロック並ベゲーム完了の遅れ)発生時もゲームを続行し、ボーナスタイムを獲得できる.必要な対応策を要求分析により導出し、各モデルに記載した.
- ・分析モデルにてブロック並べゲームに必要な走行パターンを全て抽出し,設計モデルに利用した.また,走行パターン毎に必要な時間を実験により確認し,戦略(要求モデル,分析モデル)にフィードバックした.

モデルの構成

P1 要求分析

全国大会優勝に必要なリザルトタイムを想定し目標とし,目標達成に必要な機能をまとめた.

P2,3 分析

「ブロック情報」と「経路情報」(1.2 システム配置参照)がゲーム攻略に必要であることを導き出した経緯を示した。

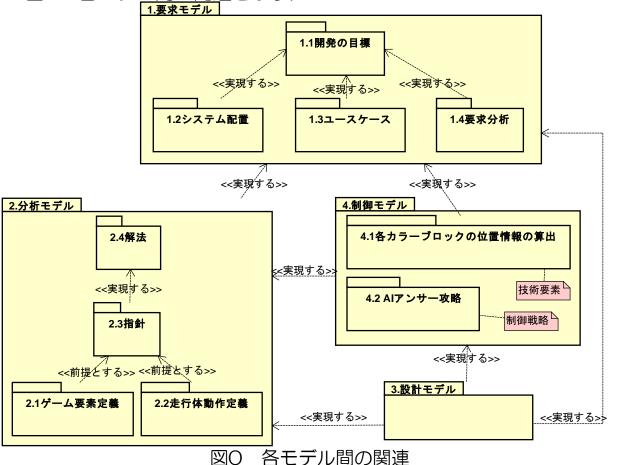
P3,4 設計

ブロック並べゲーム全体の制御を2つのシステム毎にまとめた.

P5 制御技術

- ①各カラーブロックの位置情報の算出 画像解析による必要な色の抽出,情報の算出過程などを示した.
- ②AIアンサー攻略

図Oに各モデル間の関連を示す.



1.1開発の目標

目標:Rコースのリザルトタイム-3秒を獲得する 【目標の詳細】

上記目標を達成するために,以下①~③を満たす必要がある.

①走行タイムの目標:25秒

1.2ユースケース

28-8-1

HacEV(走行シ

ステム)担当者

のユースケース記述を表1-2に示す.

走行体の平均速度とスピード競技区間の距離より、目標タイム を算出した

走行経路10.2m÷平均速度0.4m/s =25.5s ≒ 目標走行タイム25s

②ブロック並ベゲームの目標ボーナスタイム合計:23秒

本チームの戦略は、「パワーブロックは移動させず、パワースポットに4つ のカラーブロックをブロック有効移動する」、(指針2.3参照)

13秒(有効移動4つ) + 10秒(パワースポット設置5つ)

= 23秒(ブロック並ペゲームの目標ボーナスタイム)

ブロックの設置全パターンの内,99.35%において,上記目標達成を保証 する(残りの0.65%については指針2.3参照)

UC.1

図1-1 走行システム(ユースケース図)

表1-1 ブロック並べを攻略する(ユースケース記述)

4 つのカラーブロックを移動させ、「ブロック有効移動」

スピード競技 を攻略する

ブロック並べ を攻略する

直角駐車場を 攻略する

キャリプレー ションをする

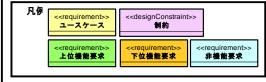
UC.2

Rコースを攻 略する

ブロック並べを攻略する

を成立させる。

③直角駐車場停止の目標:ボーナスタイム5秒



◆各区間攻略に必要な時間

ブロック情報算出システム

プロック情報 を算出する

を送信する

概要

UC3

120秒以内にRコースを完了するため、各区間攻略に必要な時間を検討し た.結果図1-4に示す.「ブロック並べ」「直角駐車場」を攻略するに関しては 実際に走行体を動作させ検討した.検討の詳細に関しては、「2.2走行体動 作定義」参照

120秒以内にRコースを攻略完了する

カメラ画像を取

プロック運搬車

序を決定する

図1-2 ブロック情報算出システム(ユースケース図)

ブロック情報を算出する

ブロック初期位置を決定する

ブロック運搬先 を決定する

表1-2 ブロック情報を算出する(ユースケース記述)

ブロック情報を算出し,走行システムに送

⑨ブロック情報算出システムは、「ブロッ

ク情報 | を走行システムへ送信する

text = 競技規約より スピード競技25秒以内に完了 ブロック並べ80秒以内に完了 直角駐車場を15秒以内に完了 text = 「1.1 開発目標」参照 text = 「2.2走行体動作定義」参照 text = 「2.2走行体動作定義」参照

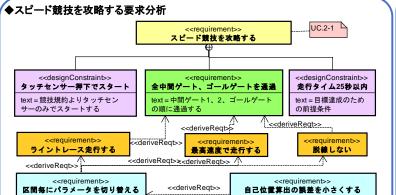
図1-4 各区間攻略に必要な時間検討結果

カメラ

◆ブロック情報を作成する要求分析 UC.3 <<designConstraint> <<designConstraint>> <<requirement>> 1分以内に入力 キャリブレーション時のみ入力可 ブロック情報を作成する text = 初期付置コード入力の制限 text = 初期位置コード入力の制限 <<deriveReqt>: <<deriveRent>> <<deriveReqt>> <<deriveReat>> <<deriveReqt>: <<deriveReat>> ブロック情報をPCからHackEVへ転送 ブロック運搬情報を算出する ブロックの位置情報を算出する カラーブロック各色の位置情報を取得する text = カメラからの画像を解析することで、カ text = ブロック情報をPCからHackEVへ text=初期位置コードより、全力 text = ダイクストラ法よりブ ブロック、全パワーブロック ラーブロック各色毎の位置情報を取得する ロック運搬情報を算出する Blutooth通信にて転送 の位置情報のみを取得する -derive Rent <deriveReqt>> <<designConstraint>> 通信データを少なくする 会場への対応 カメラ画像を解析する text = FV3のBluetooth通信はフ 詳細制御モ text = カメラ画像よりラーブロックの位置情 text = 地区大会と異なる会場 リーズのリスクがあるため通信を デル4.1参照 報とカラーブロック色情報を取得する 最小限にし通信にてフリーズのリ スクを軽減する

図1-5 ブロック情報を作成する機能要求・非機能要求

1.1開発の目標と1.3ユースケースより、目標、ユースケース、制約から機能要求、非機能要求を導出し、ブロック情報算出システムの要求図(図1-5)と走行



1.4 要求分析

システムの要求図(図1-6~図1-8)にまとめた.

図1-6 スピード競技を攻略する機能要求・非機能要求

UC.2-3 直角駐車場を攻略する 直角駐車場内で停止する 直角駐車場内まで移動する text=直角駐車場前のライン 直角駐車場を15秒以内に完了 まで移動する text = 「2.2走行体動作定義」参照 <<deriveReqt>> <<deriveRegt> 自己位置算出の誤差を小さくする 正確に枠内にて停止する ext = モータのエンコーダ値より走行積算距離 text=枠からはみ出ない、 を算出し、コース上の位置を算出するが、この 枠に接触しない 誤差を小さくする。 図1-7 直角駐車場を攻略する機能要求・非機能要求

◆直角駐車場を攻略する要求分析

◆ブロック並べを攻略する要求分析 UC.2-2 ポーナスタイム23秒獲得 プロック並べを攻略する text = 全カラーブロックを移動させる パワーブロックは移動させない ext=【前提条件】 目標のボーナスタイムを獲得するために、以下を前提条件とする ・パワーブロックは移動しない(必ず5か所のパワースポットが発生) · ブロック有効移動×4、パワースポット設置×5にて、ボーナスタイム23

text = モータのエンコーダ値より走行積

算距離を算出し、コース上の位置を算出

するが、この誤差を小さくする。

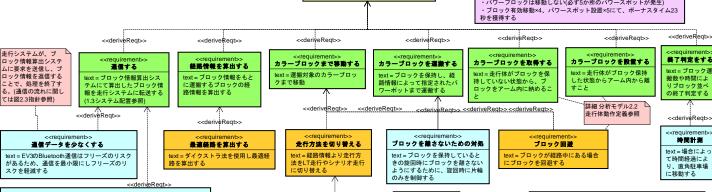


図1-8 ブロック並べを攻略する機能要求・非機能要求

自己位置推定

場所を把握できる

走行システムがゴールゲートを通過していること システムが起動していること 事後 全てのカラーブロックが色の一致したパワースポットへ設置されていること 基本 D走行システムは,ブロック情報算出システムへ「ブロック情報要求」を送信する ①スターター2は,ブロック情報算出システ ②走行システムは,ブロック情報算出システムより「ブロック情報」を受信する ムに初期位置コードを入力する ③走行システムは,経路情報を算出し,記録する ②ブロック情報算出システムは,①のデー ④走行システムは,ブロック並ベエリア進入を検知する タをもとにブロックの位置情報を算出する ⑤4つのカラーブロックが運搬完了するまで⑥~⑩を繰り返す ③ブロック情報算出システムは,カメラよ ⑥走行システムは,運搬対象ブロックまで移動する りカメラ画像を取得する ⑦走行システムは,ブロックを保持する ④ブロック情報算出システムは.③のカメ 8 走行システムは,ブロックを運搬先ブロック置き場まで運搬する ラ画像を画像解析により、「ブロック情 ⑨走行システムは,ブロックを設置する 報 | を算出する(表1-3参照) ⑩走行システムは,終了判定を行う ⑧ブロック情報算出システムは,走行シス 例外 系列 1.スタートSW押下から経過時間が105秒を超えている場合,ブロック並べを終了する テムより「ブロック情報要求」を受信する

「1.1開発の目標」を実現するためにシステムが実現するユースケースを定義した。走行システムが実現するユースケース図を図1-1に示す。ブロック情 報算出システムが実現するユースケース図を図1-2に示す「ブロック並べを攻略する」のユースケース記述を表1-1に示す「ブロック情報を算出する」

プロック特等質用システム

UC.2-1

UC.2-2

UC.2-3

pc(ブロック情

報算出システム)担当者

1.3システム配置

(ブロック運搬先.ブロック運搬順序)

(2.3 指針)

システムの配置を図1-3に示す「ブロック情報算出システム」と「走行システム」の2つのシステムにて構成される. HackEV(走行体)のみにて全処理を実行すると,走行体に負荷が掛かるため,表1-3に示す内容にて,処理を分割した

表1-3 システム毎の役割分担

はUC2-3「直角駐車場を攻略する」に移行する

ブロック情報算出システム (PC) (走行システム) ブロック情報(下記)を算出 経路情報の算出 ・ブロック(パワーブロック,カラーブロッ ク各色)初期位置算出 ブロック操作など) ・カラーブロック運搬情報を決定

· 走行体制御(移動)

基本系列①から4秒以内に基本系列②が完了していない場合,走行システム

HackEV

り提供 startupOfmjpg_streamer.sh 今回の開発対象 8 ※Raspberry Pilこは Webカメラが接続されるが、省略 ブロック情報算出システム 钉 走行システム 🗊 <<artifact>> ブロック情報

ブロック情報算出シス テムにて算出・決定す る情報 (表1-3参照) 経路情報とは、走行シ ステムにて算出される エリア内を通る経路の

(表1-3参照)

図1-3 システムの配置

非機能要求まとめ

車場に移動することとする

Bluetooth通信失敗時の対処

text = Bluetooth通信が失敗した場合、要求待ちの走行体が動かなくなっ てしまうため、最低限のボーナスタイムを獲得することのため、直角駐

text = 車速、旋回量などのパラメー

タを区間毎に適切なパラメータに認

導出した非機能要求のまとめを表1-4に示す。 ISO9126の内容に乗っ取り非機能要求を導出し

表1-4 導出	た、祇園の都台工制変する非機能要求もある。 出した非機能要求のまとめ	
目	内容	L
	最低限の通信にてデータを転送することでフリーズの	[[1
	可能性を低くする.	時
	区間毎にパラメータを適切な値にすることで脱線の可	ľ
☑間毎に適切なパラメータに切	能性を低くし,区間毎に最高の速度で走行出来るよう	Βlυ
り替える	にする.	┖

項目	内容
自己位置誤差を小さくする	自己位置誤差を小さくすることで、区間毎のパラメータの切り替えや、ブロック並べにて走行方法の切り替えに誤差がなくなり、脱線の可能性を低くする.
	ブロック並べ実行中.一定時間経過にてゲームを中断し.直 角駐車場へ向かう.詳細は「2.3指針」参照
	通信不具合などでブロック情報を取得できなかった場合, 直角駐車場に移動する.

自己位置算出の観差を小さくする

text = モータのエンコーダ値より走行積算距離を算出し コース上の位置を算出するが、この誤差を小さくする。



2. 分析モデル

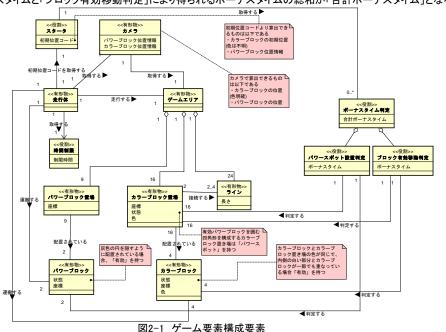
2.1ゲーム要素定義

ブロック並ベゲームを解くにあたり、必要な情報とそれらの静的な関係を図2-1に示す

本ゲームを解くにあたり、事前情報として「パワーブロック位置情報」、「カラーブロック位置情報」が必要である.

「パワーブロック位置情報」に関し、「走行体」は「スタータ」が入力する「初期位置コード」または「カメラ」より取得すること ができる「カラーブロック位置情報」に関し、「走行体」は「カメラ」「初期位置コード」より取得することができるが、「初期位 置コード」では色情報が不明である.

「パワーブロック」は属性として「状態」をもち、「パワーブロック置き場」に、パワーブロックがパワーブロック置き場の灰色 の円を完全に隠すように置かれている場合、そのパワーブロックは「有効パワーブロック」として扱われる。(「カラーブロッ ク」に関しては省略する)ボーナスタイムは「ボーナスタイム判定」により,算出される、「パワースポット判定」によって得ら れるボーナスタイムと「ブロック有効移動判定」により得られるボーナスタイムの総和が「合計ボーナスタイム」となる.



2.3指針

◆ボーナスタイム獲得戦略

ボーナスタイム獲得の指針を決めるために必要となる項目を挙げた

1.ブロック並ペポーナスタイム獲得戦略

「パワーブロックは移動させず、パワースポットに4つのカラーブロックをブロック有効移動する」

パワーブロックを移動しない理由は、隣接するパワーブロック置き場にパワーブロックを配置し た場合、パワースポット二つが重複するため、最大4秒のボーナスタイム(「パワースポット設置」 図2-3 例外配置図 ×2)を獲得できる.しかし.パワーブロック置き場まで移動するには.非ライントレース走行が必要

であり、脱線のリスクがあるため、パワーブロックは移動しないことにした

上記については、全パターンのうち99.35%可能であるが、 0.35%の不可能な場合については図2-3のように

中央に色を合わせて配置することで最低限のボーナス タイムの獲得を目指す.

2.カメラと初期位置コードを使用する

初期位置コードからパワーブロックとカラーブロックの 位置を取得する(色不明).カメラによりカラーブロックの 色を明確にする.(4.1各カラーブロックの位置情報の算 出参照)

初期位置コード入力し、ゲーム実行までの流れを図2-4

3.3つ目のブロック運搬完了後の経過時間が105秒以上 の場合、直角駐車場に移動する

直角駐車場攻略には約15秒必要である.(表2-1参照)4 つ目のブロックを運搬し、「パワースポット設置」+「ブ ロック有効移動」のボーナス(4秒)を獲得するより,直角 駐車場停止(5秒)の方がボーナスタイムは多いので、3 つ目のブロック運搬完了後の経過時間が105秒以上の 場合,直角駐車場に移動する.

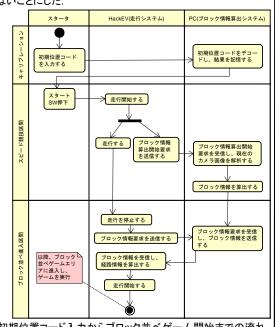


図2-4 初期位置コード入力からブロック並ベゲーム開始までの流れ

2.2走行体動作定義

|◆ルール上の制約

①制限時間120秒(実際はゴール までの走行や直角駐車場がある ため,差し引いて80秒前後) ②ブロックの色情報の取得はキャ リブレーション時には行えない. ③パワーブロックコードは8種類

◆走行体動作

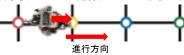
◎基本走行動作

・ブロック置き場を通過する(直進する.曲がる)

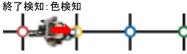
【直進する】、【曲がる】は①、②は同様の動 作だが,③で走行方法が変わる

① ライントレース走行する 走行方法:ライントレース走行

終了検知:300mm走行(◆コースの形状より)



② 色検知するまで直進走行する 走行方法:直進走行



【直進する】

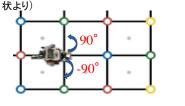
③ 直進走行する 走行方法:直進走行

終了検知:60mm走行(◆コースの形状より)



③ 次に向かう方向に旋回する

走行方法: 旋回走行 終了検知:90°,-90°旋回(◆コースの形



◎ブロックの操作

◆コースの形状

・ブロック置き場 1区間

縦(座標コード0-4)400mm

横(座標コード0-1)450mm

・カラーブロック置き場の

角度は.90°である。

直径60mm

図2-2参照

・ブロックを取得する(直進する.曲がる)

10

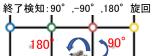
図2-2 コース形状

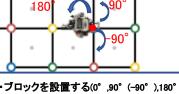
【直進する】はブロック置き場を通過 すると同様のため省略する

① ライントレース走行する 走行方法:ライントレース走行 終了検知:300mm走行

② 色検知するまで直進走行する 走行方法:直進走行 終了検知: 色検知

③ 次に向かう方向に旋回する 走行方法: 旋回走行





(1.(2.(3)は同様の動作であるが.(4)以降は 次に向かう方向によって動作が変わる. 向かう方向は0°.90°(-90°),180°

がある. ① ライントレース走行する 走行方法:ライントレース走行 終了検知:300mm走行

② 色検知するまで直進走行する

走行方法:直進走行 終了検知: 色検知

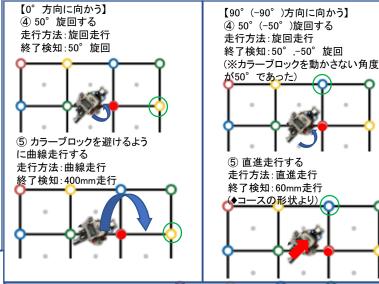
③ 60mm後進走行する 走行方法:後進走行



◆走行結果

実際の走行結果より、ブロック運搬に必要な走行方法の時間を表2-1に示す





◎ブロック回避方法

進行方向にブロックが設置してある場合の対処方法として,ブロック回避がある. この時、走行体の状態には【ブロックを保持していない場合】、「ブロックを保持している】 の2つの状態が考えられる

【ブロックを保持していない場合】はブロックを設置する動作と同様のため省略する 【ブロックを保持している場合】

ライントレース走行を240mmする

【180°方向に向かう】

④ 180° 旋回する

走行方法: 旋回走行 終了検知:180°旋回

- ① ライントレース走行する走行方法:ライントレース走行終了検知:240mm走行(◆コース の形状より)
- ② 以降は、ブロックを設置する動作の④以降と同じである.

2.4解法

◆ブロック情報の算出方法

以下の情報をPCにて算出し、HackEVに転送する「1.3 システム配置」参照

・ブロック初期位置の決定

初期位置コードとカメラからの画像を使いブロックの初期位置を算出する。 (詳細は4.1制御モデル参照)

ブロック運搬先とブロック運搬順序を決定する

ボーナスタイム獲得戦略とルール上の制約③より、ブロックの運搬先の候補は最大2か所であり、1つは 1か所の場所があるため、8通りある.

また、4色のカラーブロックの運搬順序の全パターンは24通りある。

よって,運搬先と運搬順序を合わせた全パターンは192通りある.

その192通りを初期位置から運搬先の走行体の移動距離が最適となる運搬先と運搬順序を採用する 走行経路にはいくつものパターンが考えられるため、走行経路に「コスト」という独自のパラメータにて評画 価し、最適な経路を算出する.経路探索にはダイクストラ法を用い、【コストの算出詳細】に示すコストを付 加し、最適な運搬先と運搬順序を算出する.(①②使用) 図2-7に例の流れを示す.

◆経路情報の算出

以下の情報をHackEVにて実施する.

経路情報とは走行体がPCより得たブロック情報より、ブロック初期位置からブロック運搬先へ移動 する際にどのラインを通るかをあらわす情報であり、この情報をもとに走行体は移動する. 走行経路にはいくつものパターンが考えられるため、走行経路に「コスト」という独自のパラメータ にて評価し、最適な経路を算出する、経路探索にはダイクストラ法を用い、【コストの算出詳細】に示 すコストを付加し、最適な経路情報を算出する.(①使用)

【コストの算出詳細】

①経路にブロックが設置されている箇所はコスト高

走行体を動作させ,実験した結果,ブロック回避(「2.2走行体動作定義参照」)には多 くの時間を要したよって、走行体の移動経路にブロックが設置されている場合はコス ト高とする.

例:走行体が,図2-5の移動先(緑の丸)に移動する場合,ブロック有がコスト大 ②最後に設置するブロックは直角駐車場に近いほどコスト低

ブロック並ベゲームから,直角駐車場へスムーズに移動するため,ブロック並ベ ゲームの出口(ここから直角駐車場へ向かう)を11番ブロック置き場とする.最後に配 置するブロックは11番ブロック置き場に近いほどコスト低とする.



図2-5 経路にブロックが配置さ れている場合のコスト評価例

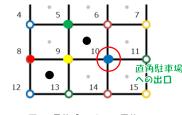
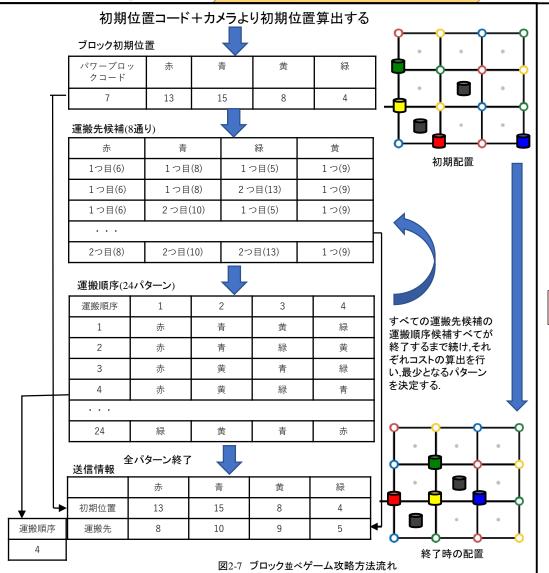


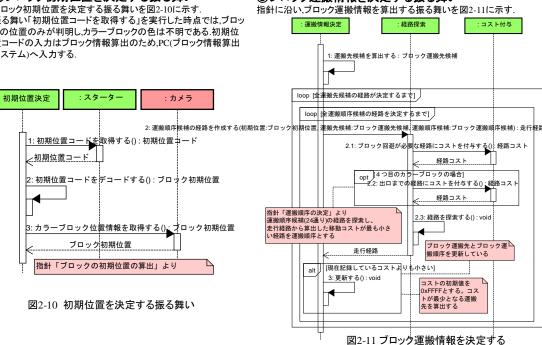
図2-6 最終ブロックの配置位 置に関するコスト評価例

2. 分析モテル

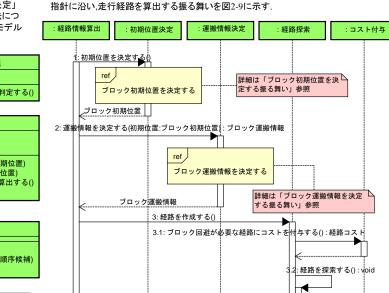


②ブロック初期位置を決定する振る舞い ブロック初期位置を決定する振る舞いを図2-10に示す 振る舞い「初期位置コードを取得する」を実行した時点では、ブロッ クの位置のみが判明し、カラーブロックの色は不明である.初期位 置コードの入力はブロック情報算出のため、PC(ブロック情報算出 システム)へ入力する.

③ブロック運搬情報を決定する振る舞い



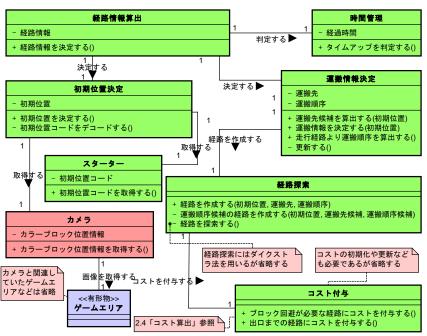
指針に沿って問題を解くために必要な要素を追加したクラス図(図2-8)にまとめた、さらに、それら要素 を使って問題を解くための方法・手順をシーケンス図(図2-9~図2-11)にまとめた.指針で定めたブロッ クの初期位置決定、運搬先の決定と運搬順序の決定、経路情報の算出、はそれぞれ「初期位置決定」 「運搬先決定」と「運搬順序算出」「経路情報算出」としてクラスに追加したまた経路の探索方法につ いては「経路探索」と「コスト付与」にクラスを追加した.クラス図シーケンス図の色分けは3.設計モデル のパッケージモデルに準ずる



走行経路

①走行経路を算出する振る舞い

図2-9 指針に沿い、走行経路を算出する振る舞い



3。 設計モデル

3.1設計意図・方針

【ブロック並ベゲーム開始条件】

走行体システムがブロック並べ進入区間を走行中に,カラーブロック置き場8(競技規 約「7.4.3【DA】初期位置コード」参照)の色検知をすることで、開始とする.

図2-8 ブロック並ベゲーム攻略方法のクラス図

【ブロック並ベゲームの進捗管理】

ブロック並ベゲームを2つの状態パラメータ「シナリオ」、「ミッション」にて進捗管理し、 実行する.詳細を以下に示す.

◆シナリオ

運搬したブロック数を状態「シナリオ」として管理する.

ブロック並ベゲームのシナリオの流れを「図3-1.1」に示す

エリア進入完了後,ブロックの運搬を実行する.

4つのブロック運搬完了にてゲーム終了とする.

「ブロック4個目運搬中」が完了にて,ブロック並べゲーム終了とする.

また,「ブロック3個目運搬中」完了時にスタートからの経過時間を確認し,105秒以上経 過していた場合も,ブロック並ベゲーム終了とする.

ブロック並ベゲーム終了後,ゲームエリア退出を実行する

(「2.3 指針参照」)

シナリオ「ブロック1個目運搬中」~「ブロック4個目運搬中」毎に

ミッションを実行する(詳細は下記,「ミッション」参照)

シナリオ型を図3.1-3にて定義し,構造モデル(図3.2-1)にて利用する

◆ミッション

ブロックの初期位置まで移動~ブロックを捕獲~

ブロック移動先まで移動~ブロックを設置

以上の動作を1セットとし、状態「ミッション」として管理する.

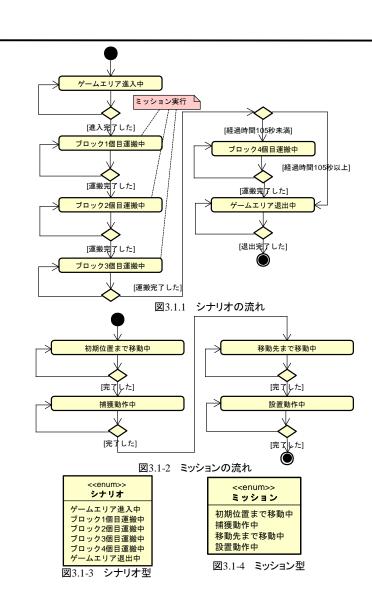
ミッションの流れを「図3.1-2」に示す.本ミッションの流れを

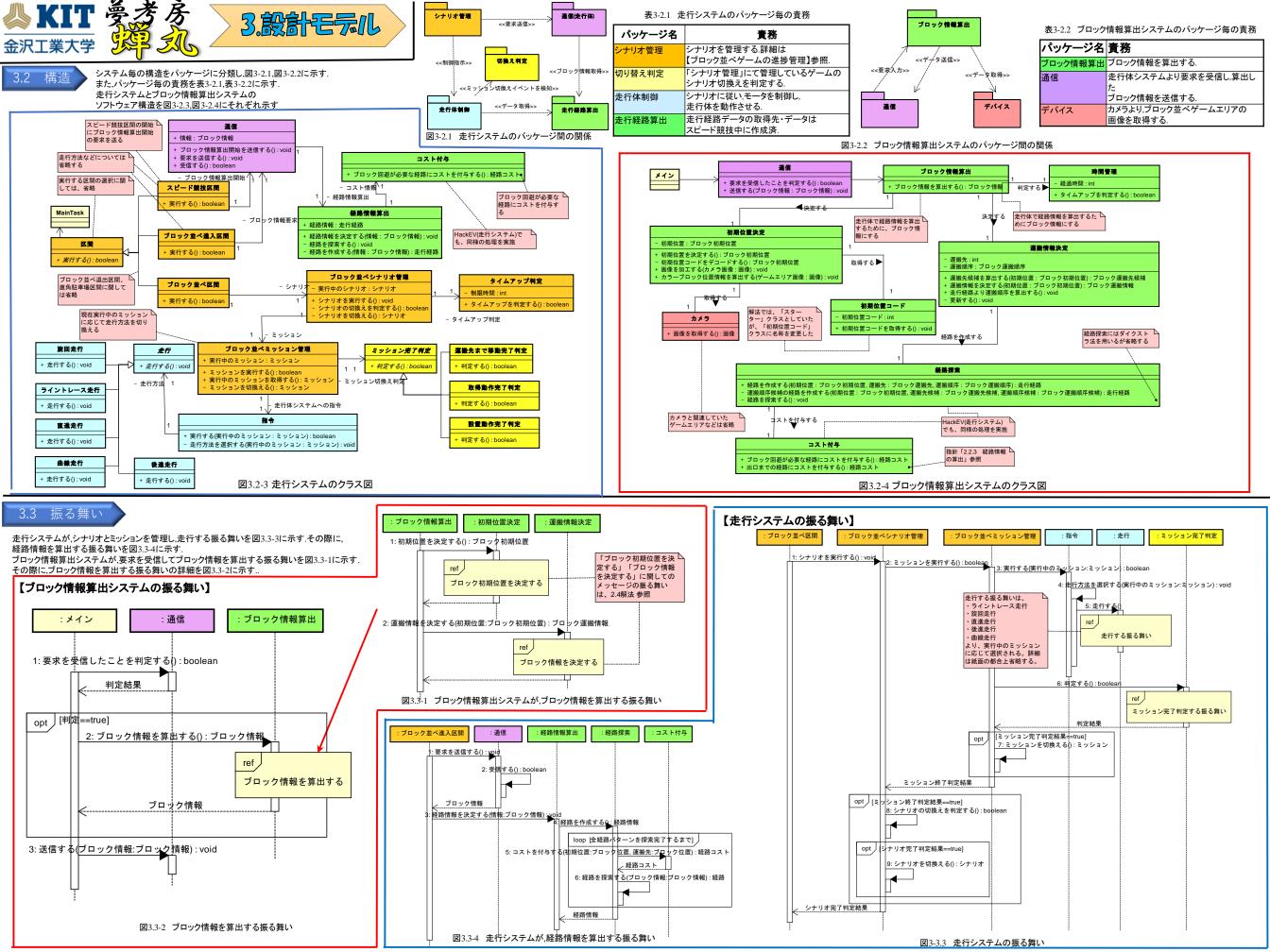
全て終了したら、次のシナリオを実行する.

ミッション型を図3.1-4にて定義し,構造モデル(図3.2-1)にて利用する

【ブロック並ベゲーム終了後の処理】

ブロック並ベゲーム終了後(上記シナリオ参照),走行体はブロック並ベゲームの出 ロ(カラーブロック置き場11)へ移動する.









4期間 毛元ル

~4.1 各カラーブロックの位置情報の算出

【テーマ】

「ゲーム攻略に必要な制御技術」として、「各カラーブロックの位置情報の検出」をテーマに選択した。

4.1.1 本制御の必要性

ブロック並べゲーム攻略には,カラーブロック各色毎の初期位置情報(どのブロック置き場に設置されているか)が必要 であるが、キャリブレーション中に入力可能な初期位置コードでは、カラーブロックの色情報が含まれていない、また、カ ラーブロックがコース上に設置されるのは、キャリブレーション終了後であるため、スターターによるカラーブロックの色情 報の入力は不可能である.カメラシステム(「図4.2-1経路情報算出までの流れ」のブロック情報算出システム)を用いて、 ブロック並ベエリアの画像から各カラーブロックの位置情報を取得する必要がある。

4.1.2 カラーブロック各色毎の位置情報確定までの手順

以下にカラーブロックの各色毎の位置情報が確定するまでの手順を①~⑤に示す.

- ① カメラより取得した画像を1280×720分割し、各画像の行列をX軸、Y軸に割り当てる.
- ② カメラより取得した画像を,必要な色のみを抽出する(4.1.3 課題①参照)
- ③ ②にて色を抽出した画像に対し,初期位置コードより分かっているカラーブロック置き場に対応した部分のみを画像 解析する(4.1.3 課題②参照)(4.1.3 課題③参照)
- ④「赤」「緑」「青」「黄」の順に,ブロック置き場に対応した座標範囲内に対象とする色に対応した画像が200ピクセル以 上存在したら,そのブロックが設置されていると判定する.(4.1.3 課題③の例参照)(4.1.3 課題④参照)

4.1.3 画像解析によるブロック位置情報確定までの課題

カメラシステムにて得られた画像より、カラーブロック各色の配置を算出するにあたり、以下の課題を対策する必要がある。

【課題①】画像解析に必要な色のみを抽出

カメラ画像をそのまま使用した場合、外光の影響により色の読み取りが困難である。 (図4.1-1 カメラより取得した画像)

【課題①の対策】

カメラ画像をHSV空間色にて分析し.画像解析に必要な色のみを抽出することにより. 必要な色を明確にすることができ、後の画像解析が容易になる.

「必要な色」とは、カラーブロックの色「赤」「青」「黄」「緑」である.

図4.1-2にHSV空間色分析により必要な色のみを抽出した結果を示す.

※ HSV空間色分析にて抽出されるパラメータ(色相,彩度,明度)の組み合わせにより. カラーブロック色を判定する(例:赤の場合は,色相:15より小または 168より大)



図4.1-1 カメラより取得した画像

図4.1-2 必要な色のみを抽出した画像

【課題②】ブロック並ベゲームエリア外の情報を除去

カメラより取得した画像には、ブロック並べエリア以外の画像も含まれており、そこにブロッ クと同じ色が存在する場合がある、大会当日はオブジェなどがエリア周辺に設置されるこ とが予想され、画像解析に影響する可能性がある.)

【課題②の対策】

画像(図4.1-2)を1280×720分割した上でX.Y軸座標にて管理し.各ブロック置き場の座標 のみをHSV空間色分析の対象とする.

画像4.1-3に対象とした部分を青色で囲った図を示す.(囲っているのは大まかな位置であ る)※例:図4-1.3に示すブロック置き場10の座標は,X軸(445より大 かつ 470より小).Y軸

(240より大 かつ 315より小)である. 【課題③】処理負荷の軽減

すべてのカラーブロック置き場を判定すると処理に負荷がかかる 【課題③対策】

初期位置コードより分かっているカラーブロック置き場に対応した部分のみを画像解析 する.図4.1-4は.例として.初期位置コードよりカラーブロック置き場「1,6,9,11」の部分にカ ラーブロックが置かれていることが分かっている場合に画像解析する部分をオレンジ色

で囲んだ図である



図4.1-3 必要な色のみを抽出した画像



図4.1-4 必要な色のみを抽出した画像

【課題④】誤判定の対策

判定基準によっては、カラーブロック置き場の色などで誤判定をする場合がある。 【課題④対策】

色の判定基準をブロック置き場に対応した座標範囲内に対象とする色に対応した画像 が200ピクセル以上存在したら、そのブロックが設置されていると判定することで、他のカ ラーブロックや置き場の色などの誤判定がなくなった

4.2 AIアンサー攻略

【テーマ】

「AIアンサーの攻略」として、「数字判定」をテーマに選択した.

4.2.1 数字判定の手順

AIアンサーの攻略の手順を図4.2-1と図4.2-2に示す.

図4.2-1は走行体の動きの流れを赤矢印によって示した図である. 数字部分(①~⑥)で黒色の検知を行 い判定する.右出題数字の①⑤⑥部分は検知内容により.走行動作が変わる(図4.2-2参照).そして.右出 題数字の④が終了時に数値を決定し、解答エリアに移動する.

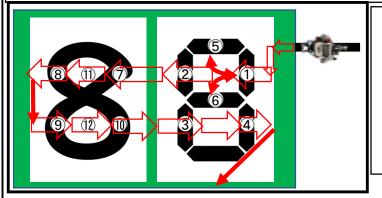


図4.2-1 攻略流れ画像

4.2.3 左数字判定

表4-2.2に左出題数字の判定を示す.

左出題数字も同様に考え,検知する箇所を変えることにより, 左出題数字も判定できるようにした.

変えた理由として、検知する箇所を右出題数字と同様の場合. 左出題数字「7」は②を検知することができると判定できるた め、5が必要なくなる、そのため、「3」は①と④だけでよいため、 右出題数字の⑥も必要なくなる。

しかし、「1」を判定することができなくなるため⑪を追加した. さらに、右出題数字の「4」の判定を⑨と⑩のみで行うと、「6」と 被ってしまうため①を追加した.

左出題数字として「2」「3」「4」「5」は曲線が多く判定するの が難しかったがズレを無くすことにより判定できるようにした. (4.2.4課題①参照)

4.2.4 数字判定までの課題

数字判定をするにあたり、以下の課題を対策する必要がある. 【課題①】

数字部分(①~②)を読みとる際に走行体の向きにズレが生 じていた場合.判定できなくなる.

【課題①の対策】

走行体にズレが生じるのは出題数字に進入前に旋回時に. 出題数字の上に敷いてあるクリアマットに走行体の尻尾が 引っかかってしまうためであっため、クリアマットに走行体が完 全に乗った状態で旋回するようにした.

表4.2-1 右出題数字の判定

2

•

lacktriangle

表4.2-1に右出題数字の判定を示す. 右出題数字は7セグメントから構成され ており、各セグメントの有無を検出すると よいしかし、全セグメントを検出するの ではなく.図4.2-1の右出題数字の通り. 最大6つのセグメントを検出することで 数字を判定することができる. 流れを図4.2-2に示す.

4.2.2 右数字判定

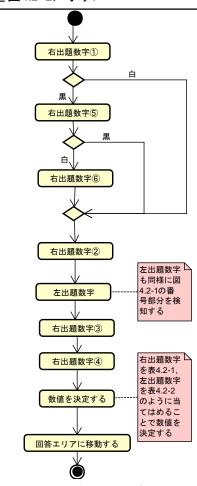


図4.2-2 攻略流れ(アクティビティ図)

表4.2-2 左出題数字の判定

nex / v / ic												
3	4	5	6	数字	7	8	9	10	(1)	12)		
•	•			0	•	•	•	•				
	•			1					•			
•				2	•		•					
	•		•	3	•			•				
	•			4						•		
	•			5		•		•				
•	•			6		•	•	•				
	•	•		7	•	•		•				
					•							