

チーム紹介、目標、意気込み

情報科学専門学校は今回が2回目の出場となります。
モデルシート・プログラミングともに自分たちで一つのものを完成させることを目標として、競技攻略を目指します！
悔いの残らない結果となるよう、全力で頑張っていきたいです。

モデルの概要

モデルの構成

1. 要求モデル

- 目標リザルトタイム-10.8sを達成するためのシステムに必要な機能をユースケース図に導出した。
- ユースケース図から機能に必要な「走行エリアを攻略する」「配置戦略を決定する」「ブロックを配置する」という要求から必要となる要求・機能・解法を抽出した。

2. 分析モデル

- 計算負荷を懸念し、局所的な最適経路の算出方法を選択した。
- 黒ブロックの配置方法、残り時間を考慮した攻略方法の変更を明確にした。

1. 要求モデル

ISCrobo2021

1-1 目標

目標リザルトタイム：-10.8s

事前にシミュレータ上で走行した結果から目標リザルトタイムを上記に設定した。
走行コースを複数のエリアに分割し、各エリアの目標を達成するための時間配分及び方針を定め、以下に記載する。エリアの分割範囲に関してはコースを基に図1-1.2に示す。
また、目標リザルトタイムの内訳を図1-1.2に示した。

エリア名	時間配分	方針
走行エリア 目標走行時間：19.2s	19.2s	確実にゴールまで走行可能な速度でラインに沿った走行から走行時間を算出。
ゲームエリア 目標ボーナスタイム：30.0s	100.8s	獲得し得る最大のボーナスタイムを設定。 走行中の残り時間により目標ボーナスタイムを再設定する。

目標リザルトタイム
＝19.2s(目標走行タイム)－30.0s(目標ボーナスタイム)
＝-10.8s
合計競技時間：120s

1-2 ユースケース分析

目標を達成するためにシステムに必要な機能をユースケース図で定義したものを図1-2に示す。
※ユースケース記述の記載は紙面の都合上、「コースを攻略する」のみとした。

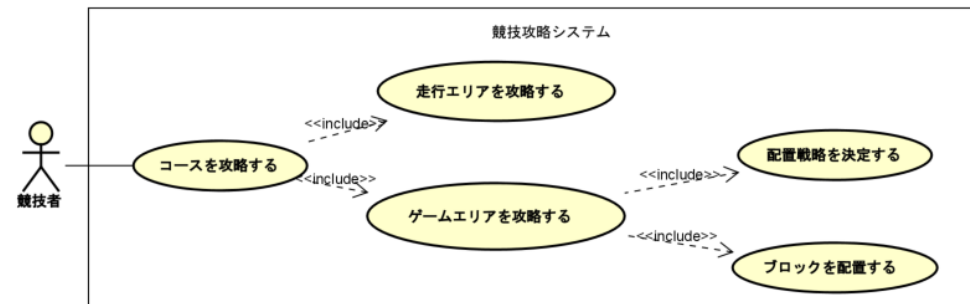


図 1-2 システムに求められる機能(UML:ユースケース図)

ユースケース名	コースを攻略する
アクター	競技者
事前条件	初期設定が終了している
事後条件	競技攻略システムが正常終了する
基本フロー	①アクターは競技攻略システムを実行させる ②システムは走行エリアを攻略する ③システムはゲームエリアを攻略する ④競技を終了する
例外フロー	競技時間の終了時に強制的にユースケースを終了する

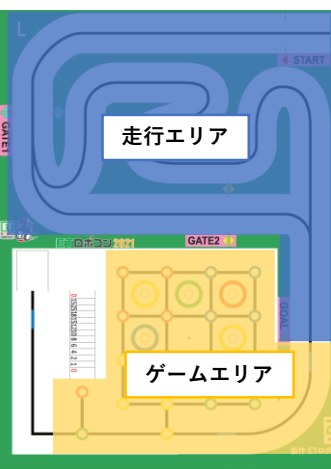


図 1-1.1 エリア分割範囲

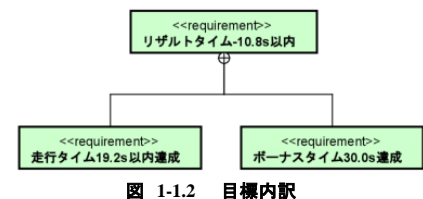


図 1-1.2 目標内訳

1-3 要求分析

ユースケースに必要な要求を以下の要求図に導出した。
また、要求の種類及び要求に対する機能などの凡例、品質特性とそれぞれに対応するアイコンを図1-3に示す。

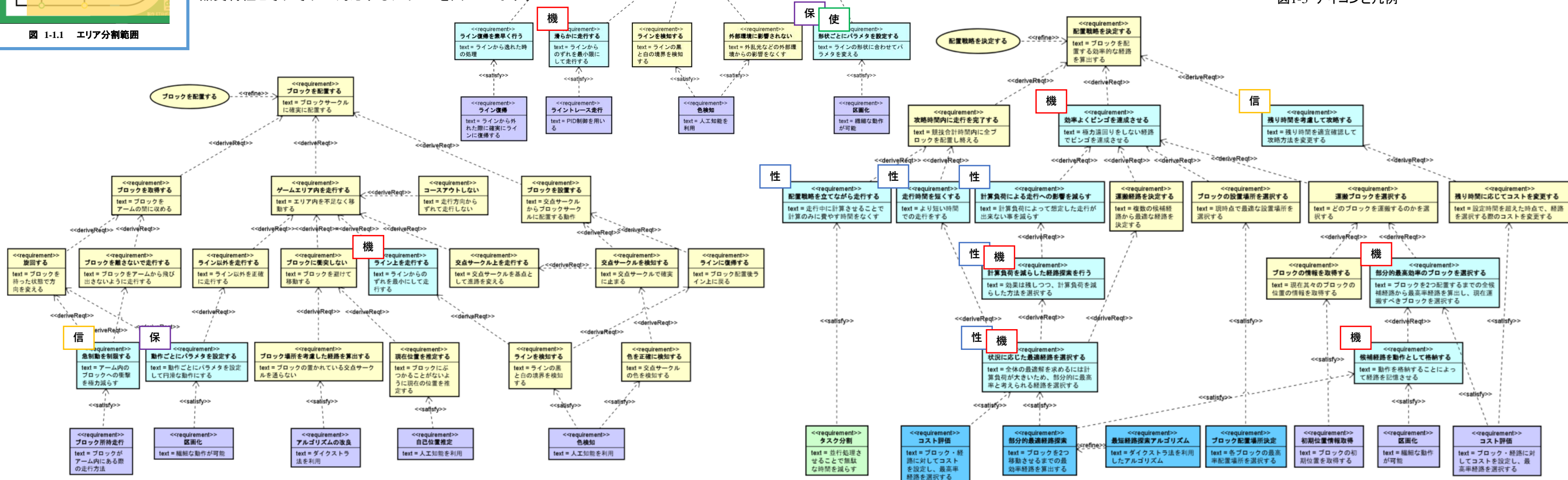



図1-3 アイコンと凡例

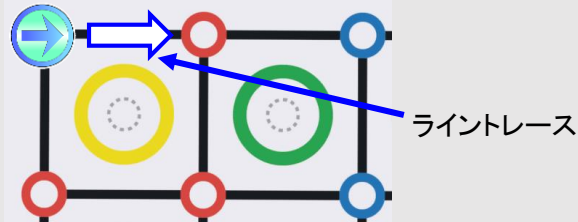
2. 分析モデル

2-1 走行体の動作定義

ビンゴエリア内の移動、ブロックの設置などの区画制御を実装するため、以下の動作に定義することにした。

のアイコンは走行体を表す

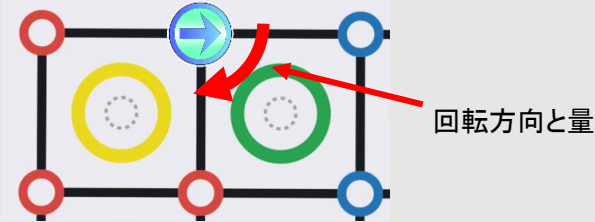
<1> 交点サークル間の移動



[動作] 交点サークル間の移動はライトレースで行う

[制約] ブロックを所持している場合停止した際に慣性でブロックが離れないよう速度を落として走行する。

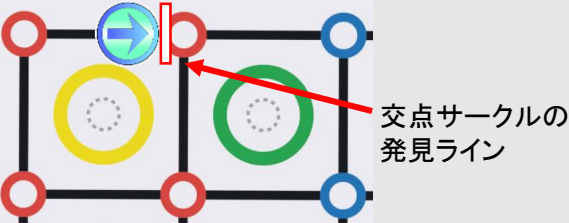
<2> 交点サークル上での向き変更



[動作] 交点サークルの中心に向きを90° 刻みで回転する

[制約] この動作ができるのは交点サークル上のみ
ブロックを所持している場合遠心力をブロックが離れないよう回転速度を落とす

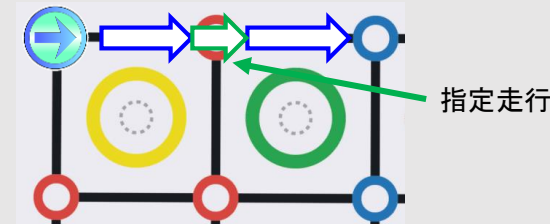
<4> 交点サークル到着時の調整



[動作] 自己位置推定の機能を用いて交点サークルの中心へ移動する

[制約] <1>動作中に交点サークルを検知した際、<3>場合でない限り必ずこの動作に遷移する。
急停止はせず緩やかに減速する。

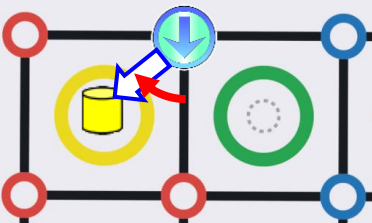
<3> 直進での交点サークル跨ぎ



[動作] 交点サークルを跨いで直進する場合は交点サークルの間は指定の走行方法に切り替える。

[制約] ライトレース時と同じ速度を保つ

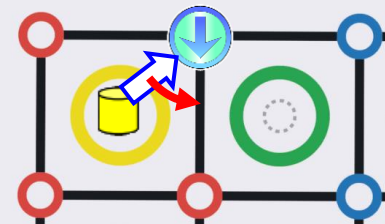
<5> ブロックをサークルに配置



[動作] 目的のサークルの方向に45回転し前進してブロックを配置する
配置後は元の場所に戻る

[制約] 交点サークル上でブロックを所持し走行体の向きとサークルが隣接している場合のみ動作できる

<6> 配置後の動作



[動作] 自己位置推定の機能を用いて前進した量を後退し、<5>動作前の走行体の向きに回転する

[制約] <5>動作後必ずこの動作に遷移する

2-2 指針

最大のボーナスタイムを獲得を攻略の指針としてゲームを攻略する際に重要となる課題を3つを挙げ、課題と解決策を下記の表にまとめた。

課題	解決策
制限時間内で最大のボーナスタイムを得られる探索法を計算時間も考慮して選択する	走行体の区画動作ごとにかかる時間を評価して最適ルートを算出していくが、どのようにルートを算出するかを下記の方法3つを考えた ・全ての経路を探索して最適ルートを算出 → 計算量が多く時間がかかる ・ブロックごとのルートを算出 → 最終的に効率が悪い可能性がある ・ 局所的に最適なルートを算出する → ◎ ブロック取得から配置までの時間が上位3つまでのルートに絞るなど局所的な探索をすることで、計算量を大幅に減らすことができる。
制限時間内に最大ボーナスタイムを得られない場合	走行中の残り時間から最大ボーナスタイムを得ることが難しいと判断し攻略方法を変更する。詳しい内容は解法で説明する。 また、このような場合になる可能性を踏まえ、センターブロックを含むビンゴは多く、且つ配置には時間がかかるため最優先で配置を試みる。
黒ブロックの配置方法	黒ブロックのみが4×4の交点サークル外にあるため、ビンゴをより効率的にするためにゲームエリア侵入時に先に黒ブロックをセンターマークに配置する。

2-3 指針に基づいた解法

1. 最短経路算出方法

最短経路算出方法としてダイクストラ法を利用し以下の手順で経路を算出する。

- ①ブロックが置かれていない交点サークルにコスト(100)、ブロックが置かれている交点サークルにはコスト(50)を設定する。
- ②走行体の現在地となる交点サークルのコストを0に設定し、走行体の向きを考慮し動作を意識したコストを隣接する交点サークルに設定する。
隣接する交点サークルでコストを設定する交点サークルと設定される交点サークルのコストを比べて、設定される交点サークルのコストの方が小さい場合はすでに割り振り済の交点サークルとしてコストを設定しない。
※ただし、コストが50である交点サークルは隣接する交点サークルにコストを設定しない。
- ③②を繰り返すことに予って、現時点でブロックにぶつかることなく到達できる交点サークルまでのコストを設定することが出来る。
- ④設定されたコストを現在地点から小さい順にたどることで、最短経路を算出することが出来る。

2. 残り時間を考慮した攻略方法の変更

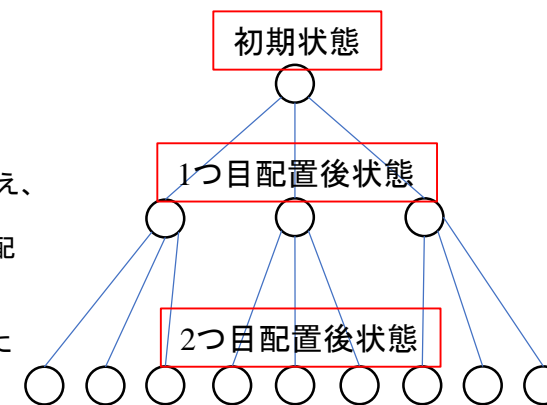
残り時間内に全てのブロックを配置することに対して難しいと判断する時間を設定しておき、その時間を過ぎた時点で最短経路算出時の動作に対するコストの基準を変更することによって、時間内に最も多くボーナスタイムを得るようにする。

3. 候補となるブロックの決定

候補となるブロックは、他のブロックに阻まれることなく到達できるブロックの中で最短で到達できるブロック上位3つまでとした。
1.最短経路算出方法により、コストを設定した際にコストが初期値ではないブロックの中でコストが小さい順に3つまでを選択する。

4. 局所的な最適経路の算出

各状況のブロックの置かれている場所を、4×4の交点サークルと8つのブロックサークル及びセンターマークに分けてコードを作成する。
これをブロック位置コード1、ブロック位置コード2とした。
全てのブロックを配置するまでの経路を全て洗い出すことは計算負荷がとても大きくなると考え、2つのブロックを配置するまでの経路の中でもっとも最適な経路を選択することとした。
木構造のように経路の探索を行う為、経路を選択する前の「初期状態」、「1つ目のブロックを配置した後の状態」、「2つ目のブロックを配置した後の状態」をブロック位置コードとして格納し、前の状態に戻すことを可能とした。
また、候補となるブロック全てに行うのではなく、1つ目のブロック・2つ目のブロックともに現在地から最短で到達できるブロック3つまでの経路を探索することとした。



3. 設計モデル

ISCrobo2021



情報科学専門学校
Information Science College

4. 設計モデル

ISCrobo2021



情報科学専門学校
Information Science College

5-1 ブロック配置の組み合わせの走査

ビンゴエリアにおいて分析モデルの指針、目標ボーナスタイムの決定のため、ブロックサークルに配置する組み合わせを計算させるプログラムを作成した。条件は下記の通りである。

- ・一つのブロックサークルに配置できるブロックは最大2つまで。
- ・センターマークに配置できるブロックは最大1つまで。
- ・ブロックは全て配置することを前提とする。

この条件を元に実行した結果は表5.1.1の通りになった。

表5.1.1

結果	
探索数	48,149,640
最高ボーナスタイム	30.0秒

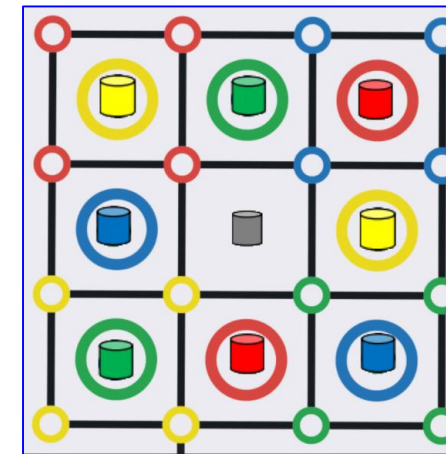
走査した探索数からボーナスタイムあたりの組み合わせ数をボーナスタイム24.0秒から30.0秒にスコープした表を表5.1.2に示した。
この表から分かることは、30.0秒の組み合わせは1通りのみであり、2位以降は27.0秒と大きく秒数を放している。そのため30.0秒の組み合わせを狙うのが高得点の鍵となるのが分かる。

また、ゲームエリア攻略中に走行体が目的のブロック配置が間に合わない場合他の組み合わせの配置に変更する必要があるが指針ではセンターブロックを優先的に配置する必要があるので、必然的に配置できない組み合わせがでてくる。その可否と例を図5.1.3に示す。

表5.1.2

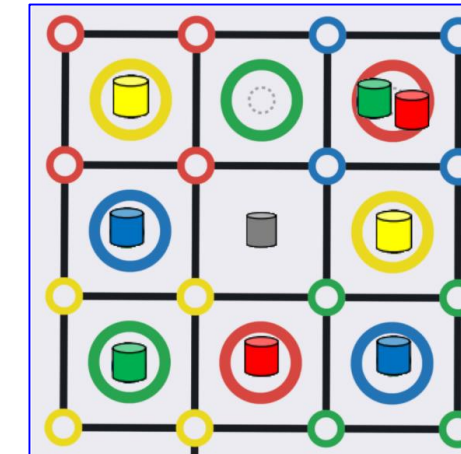
ボーナスタイム(秒)	組み合わせ数
24.0	42
24.5	0
25.0	4
25.5	0
26.0	108
26.5	0
27.0	4
27.5	0
28.0	0
28.5	0
29.0	0
29.5	0
30.0	1

図5.1.3 - 1



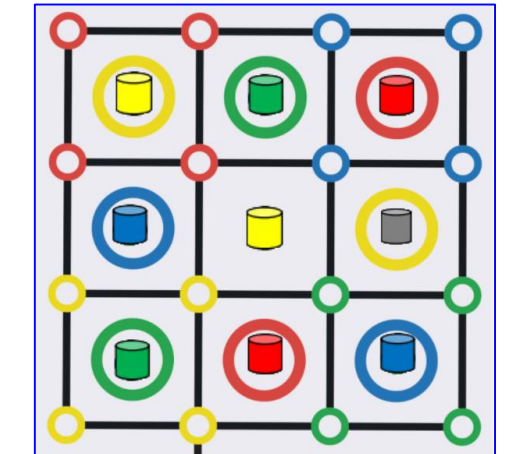
ボーナスタイム30.0秒
可能の組み合わせの配置(目標)

図5.1.3 - 2



ボーナスタイム26.0秒
可能な組み合わせの配置

図5.1.3 - 3



ボーナスタイム27.0秒
不可能な組み合わせの配置

各図の下には獲得できるボーナスタイムと可否の組み合わせかどうかを表記している。目標である図5.1.3-1の組み合わせは他の配置と比べ大差がない。ボーナスタイムが27.0秒の組み合わせ数の4種類は全て図5.1.3-3のようにになっているため、必然的に目標以外は4秒の差ができることになる。制限時間内に間に合わない場合の対応に加え経路探索と走行体の動きを洗練にすることでリスクの低減にもつながり、目標配置に到達しやすくなると考えた。