

チームNo. 160 チーム名：恵庭戦隊トシキング 所属：公立はこだて未来大学

チーム紹介、目標、意気込み

2015年から恵庭を守るため、悪の組織～デスマーチ団～と戦い続けたトシキング、今年に入り活躍していた先輩隊員が去り、チームを設立した初期隊員を知っているメンバーはもういない。このあたりで一度、ボロボロになったモデルベースを見直さないといけない。走りのトシキングから、モデルと走行にバランスをとったトシキングに方向転換を試みる。今年はデスマーチ団の放ったコロナの影響で、新規メンバーも集まりづらく苦戦を強いられている。それでも負けるなトシキング！

モデルの概要

モデル	概要
要求モデル	今年はコロナによる影響も要求図に取り入れ、ブロックビンゴ競技における非機能要求に具体的な視点を取り入れた。
分析モデル	ゲームの構成要素では、ブロックビンゴエリアの分析をした結果、新たな要素を発見したため、クラス図に反映している。
設計モデル	要求モデルで発見した非機能要求を活かし分析モデルの指針をもとに設計の方針に反映した。

モデルの構成

1. 要求モデル

- 今のトシキングに足りないものとして、非機能要求の中で見つけた「モデルとプログラミングを紐づけた開発をする」ため、一から開発することを決め、以下の要求を洗い出した
- 分析ではブロックビンゴエリアの分析は行うが、経路探索はしない
- 走行体の動作定義は過去のモデルでは多く定義しすぎたため最小限の定義にした
- 設計において、継続して使用できる走行体システムを考えた

2. 分析モデル

- モデリングの訓練としてブロックビンゴエリアの分析をし、規約以外からも新しいクラスを発見することができた
- 走行体の動作定義では、動作を制御するプログラムはあったがモデルがなかったため、解釈できなくなり一から作りモデルにも適応する
- 指針・解法では、開発リソースが足りないことを要求モデルで見つけたため、経路探索を開発しないことにした。ただし、走行体の動作のみで攻略できる方法を考えた

3. 設計モデル

- 走行体システムを継続して使えるようにシステムへの要求と走行体への要求で考えた。その結果システムへの要求をEV3の状態管理・競技の状態管理で実現し、走行体への要求を動作モデル・判別モデル・デバイスドライバで実現した

競技規約より、今年の目標設定を図1.1.1に示す。また、各競技における時間配分を表1.1.1に示す。表1.1.1より走行競技30秒の目標設定理由は、去年のCS大会コースにおいて全長と各チームのタイムから平均速度を算出すると38cm/sだった。今年の大会は2019年のコースより長くコーナー数も多いため平均速度35cm/sで走行することを決めた。



表1.1.1 各競技における時間配分

1.2 機能要求

1.1より目標を達成するために必要な機能を図1.2.1に示す.

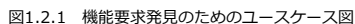


図1.2.1より発見した機能から非機能要求をの抽出をした図を図1.3.1から図1.3.3に示す。

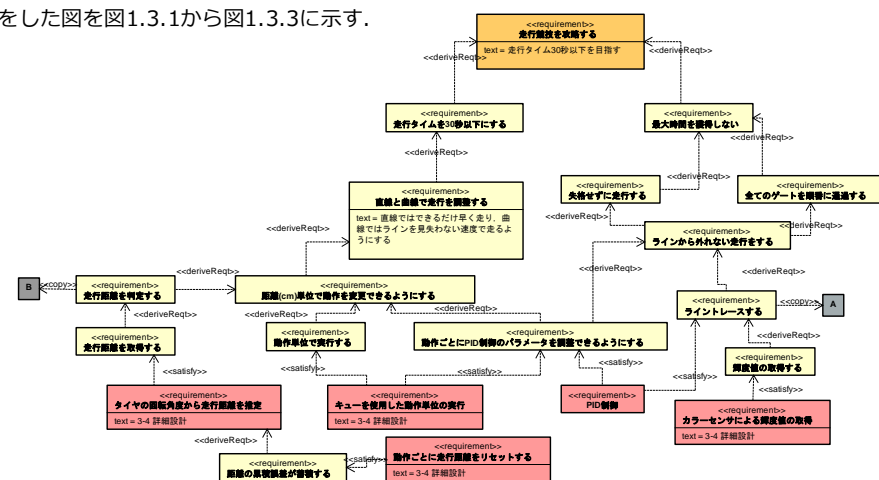
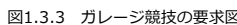


図1.3.1 走行競技の要求図



分析モデル

ブロックの上に数字が記入された紙が乗っている

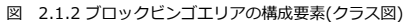
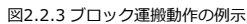


図2.2.2 交点サークル上における直進・旋回動作の例示



判定モデル	説明
D. 距離	ライトレース動作または直進動作と組み合わせて動作の終了条件にする
F. 色	ライトレース動作または直進動作と組み合わせて動作の終了条件にする

図2.4.1 ブロックの運搬経路

3.1 設計意図・方針

1.3および2.2より設計意図・方針は表3.1.1に示す。

表3.1.1 設計意図・方針

要件	意図	設計方針
来年度以降も同じコード・モデルが使用できる	開発時間短縮のため、GitHubを使用して誰でもすぐに開発できるようにしたい。 初めて参加するメンバにもモデルを使用して説明することによって教育時間短縮をしたい なるべくすぐに開発に取り組める環境を提供し、メンバのやる気を維持させたい	走行体システムの責務を状態管理と動作モデル・判別モデル・デバイスドライバに分ける
機能の追加・変更に対応できる	後から新しい要素や戦略を追加するとき走行体システムの変更箇所を少なくしたい	パッケージごとの関係において相互参照しないように誘導可能性を一方向にする

3.2 構造設計(全体)

3.1の設計意図・方針より、走行体システムの全体像を図3.2.1に示す。また、各パッケージの責務を表3.2.1に示す。

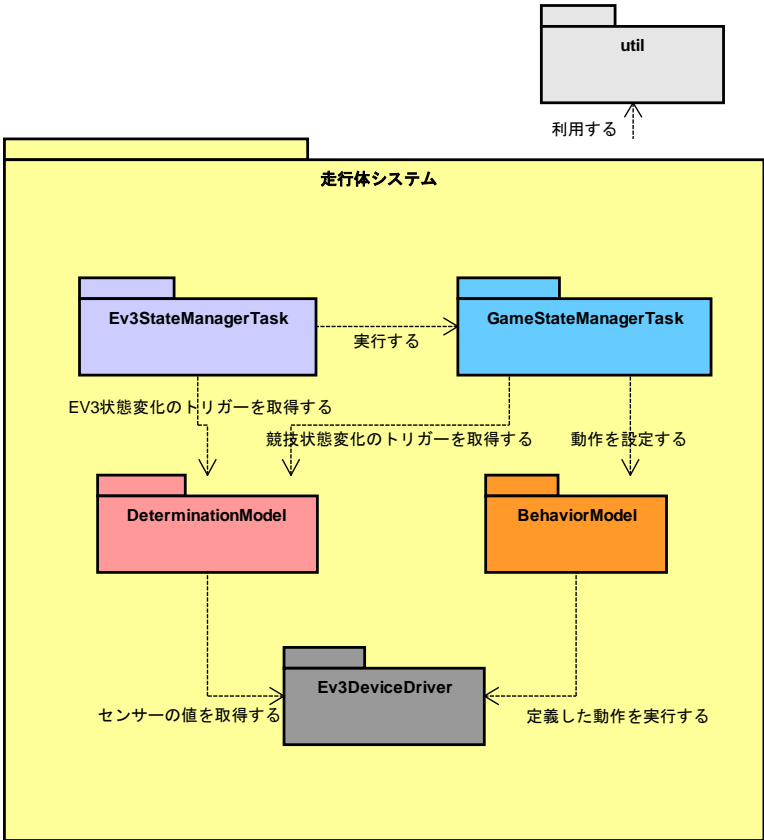


図3.2.1 走行体システムの全体像

表3.2.1 走行体システムの全体像

配置	パッケージ名	責務
走行体システム	Ev3StateManager	走行体の3状態(「ゲーム開始前」・「ゲーム中」・「ゲーム終了」)を管理する
	GameStateManager	3競技「走行競技」・「ブロックビンゴ競技」・「ガレージ競技」を管理する
	DeterminationModel	2.2で定義した検知条件から状態の変化や動作モデルの終了条件について判定する
	BehaviorModel	2.2で定義した動作から走行体の動作を生成する
その他	Ev3DeviceDriver	EV3APIを使用して、ハードウェアへのアクセスを抽象化する
	util	共通で使う機能を管理する

3.3 振舞設計(全体)

3.1, 3.2より走行体システムにおけるタスク管理について表3.3.1に示す。

表3.3.1 走行体システムのタスク管理

タスク	周期	優先度	責務と設計意図	実行パッケージ
EV3_STATE_MANAGER_TASK	200ms	高	走行体の3状態を管理するタスク。 走行体はいつでも止められる必要があるため、優先度を一番高く設定している。	走行体の状態管理 判定モデル
GAME_STATE_MANAGER_TASK	10ms	中	3つの競技区間を管理するタスク。 走行体の状態管理によってこのタスクは開始・停止をするため、走行体の状態管理タスクより低めの優先度を設定している。	競技区間の状態管理 判定モデル 動作モデル

3.4 構造設計(詳細)

図3.2.1における走行体システムの詳細構造を図3.4.1に示す。走行システムの詳細構造にはブロックビンゴ競技に必要な部分のみを記載している。また、EV3APIやutilパッケージなども省略している。

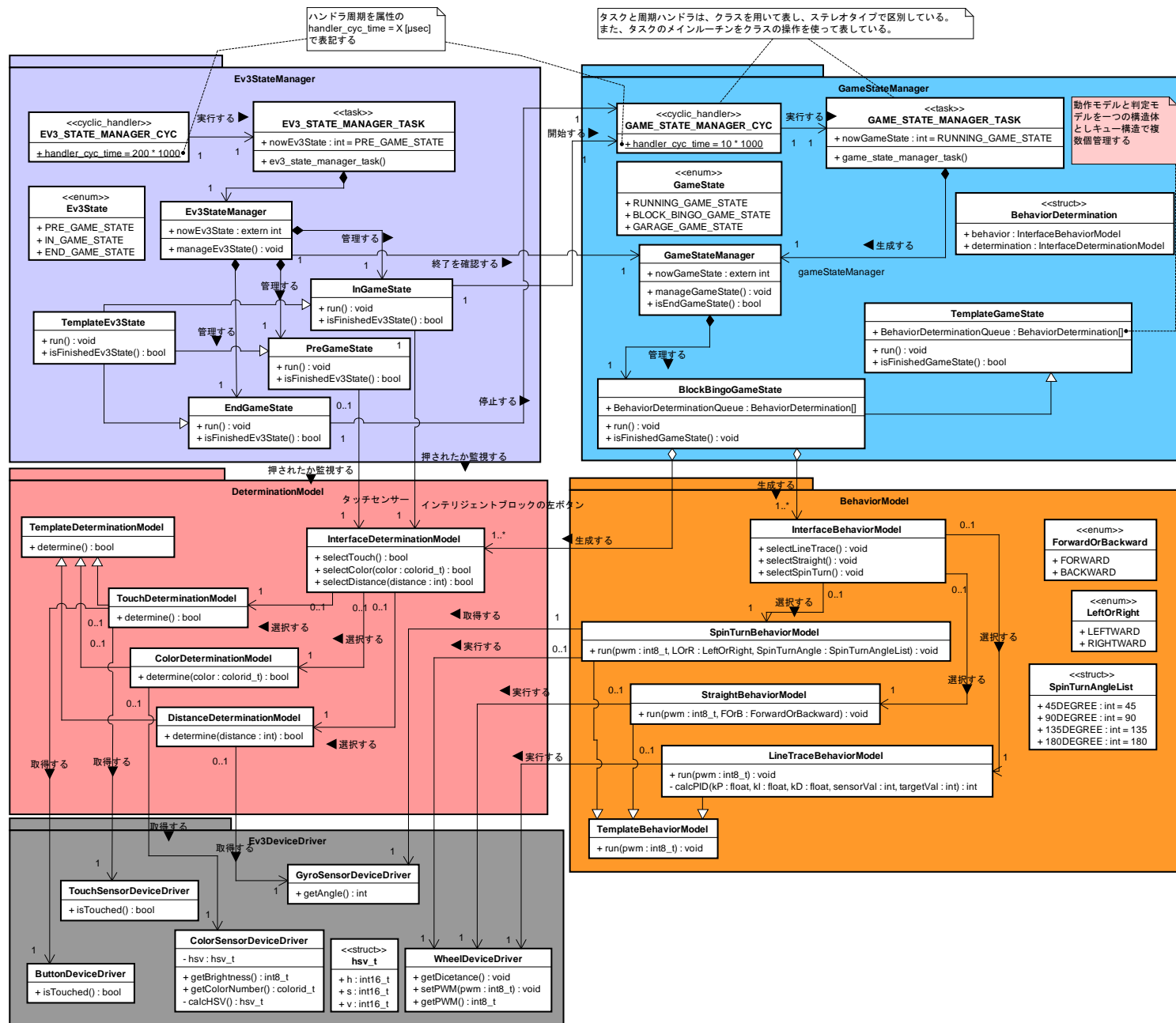


図3.4.1 走行体システムの詳細設計

3.5 振舞設計(詳細)

図3.4.1における走行体システムの振舞を図3.5.1, 図3.5.2に示す。図3.5.1は、ブロックビンゴ競技に着目した全体の振舞を示している。図3.5.1は、具体的な動作の一例を例示している。

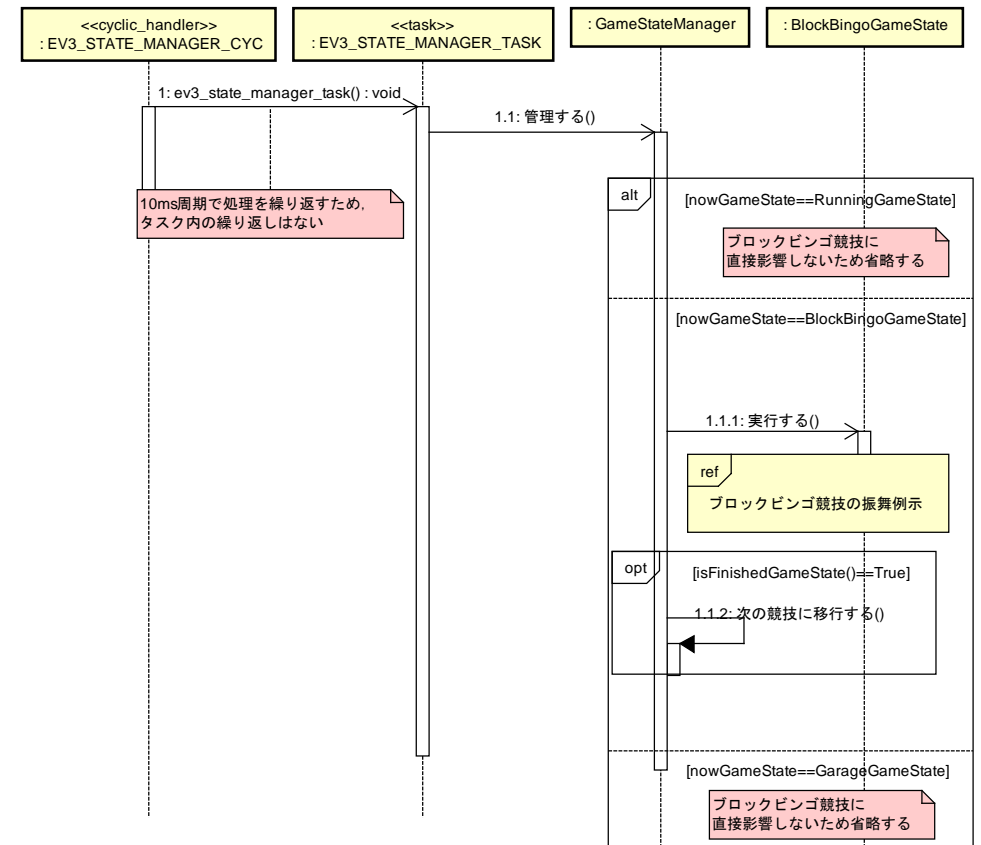


図3.5.1 ブロックビンゴ競技に着目したシステム全体の振舞

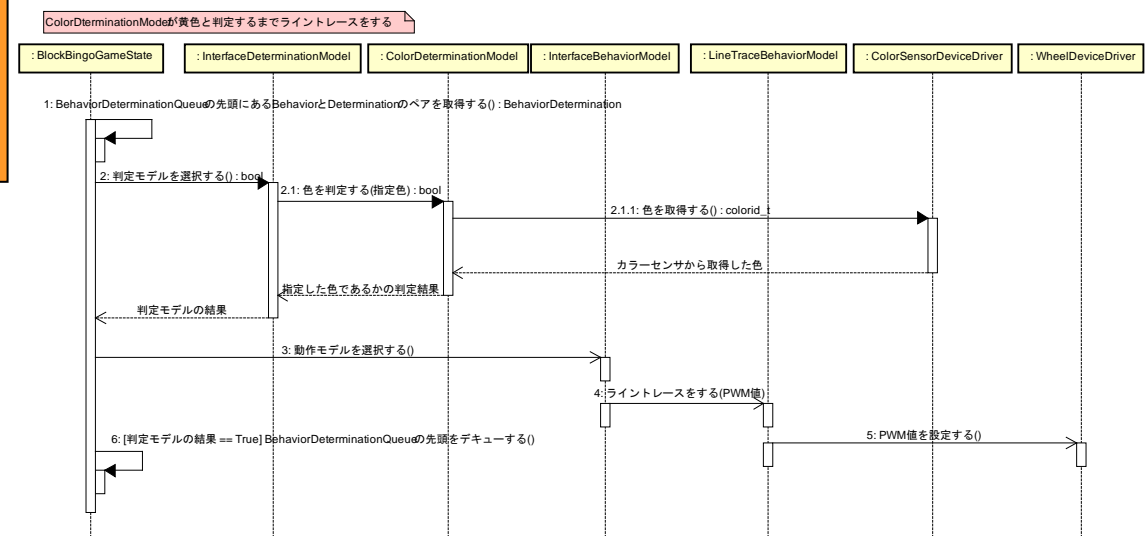


図3.5.2 ブロックビンゴ競技の振舞例示