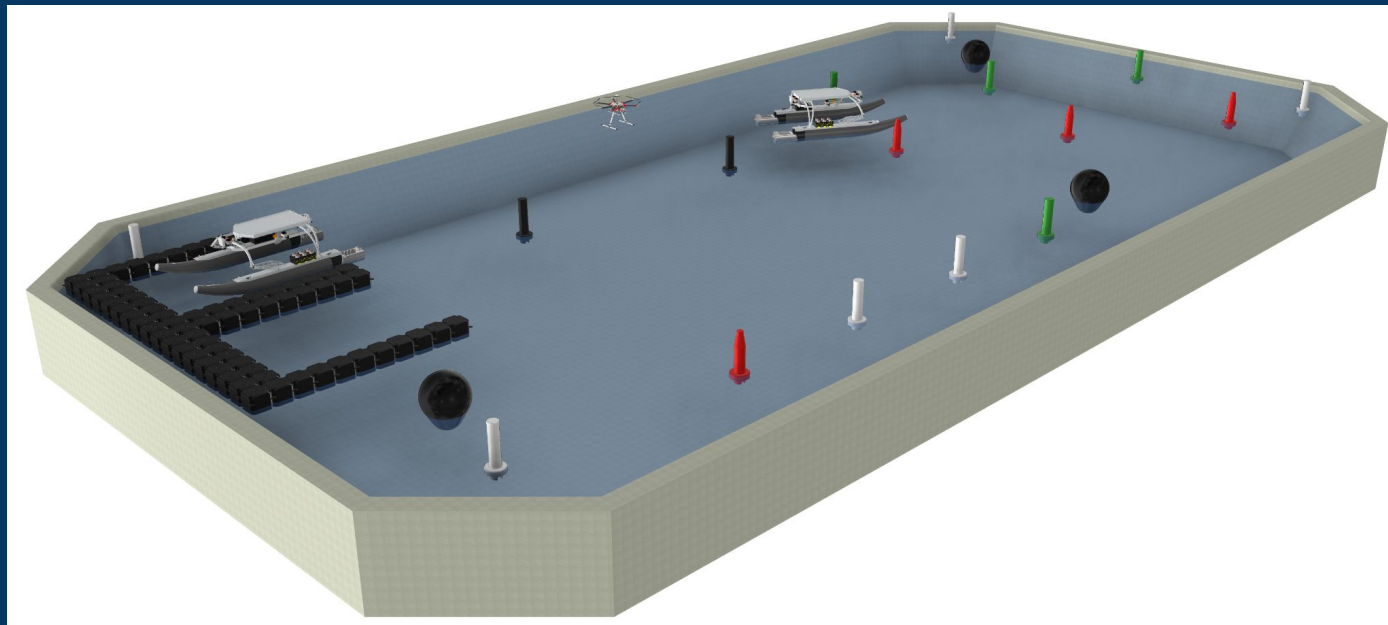


Maritime RobotX Challenge 2022 のためのビヘイビアプランニング開発

OUXT POLARIS

○  吉本幸太郎 小倉将人

Maritime RobotX Challenge



- 軽自動車サイズの船を完全自律で屋外で動作させる競技
- 自律ドローンとも協調動作
- フィールドに置かれる障害物の詳細位置は事前はわからない
- ベース機体は公式から提供、その他の開発資金はスポンサーから

OUCT Polaris



- MaritimeRobotXChallengeに参加を目的とする
 - 2014年6位入賞
 - 2022年大会では決勝進出を狙う
- 学生＋社会人の混成チーム
- 関西、関東にメンバーがいます
- 最近の主な活動
 - 月1回ペースで[勉強会開催](#)
 - [Open Source活動](#)
 - 小さな船での実験



コロナ禍での実験

小さい船作って小規模実験してます



行動計画ツールで船の行動を考えた

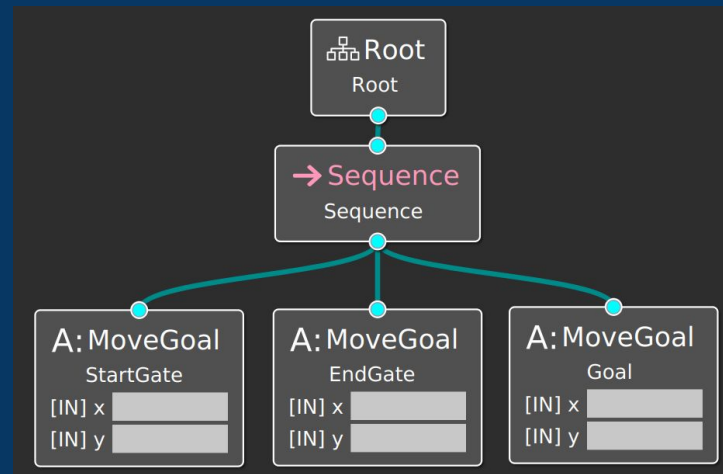
ビヘイビアツリー/BehaviorTree.CPP

運用がイマイチ上手くいかない！

専用ノードができてしまう

追加ツールを作って解決した！

やったね！他の皆も使えるようにするよ！



行動計画ツールで船の行動を考えた

有名な記述手法

- 有限状態機械(Finite State Machine)
- ビヘイビアツリー(Behavior Tree)

いずれも**行動計画の分割/再利用**を目指している

OUXTでは**ビヘイビアツリー**を採用
BehaviorTree.CPPライブラリを活用

ビヘイビアツリーを採用した理由

目標

行動計画を分割して再利用したい！

それ、本当ですか？？

FMSの 実際

遷移条件は状態内に記述される ⇒ 再利用はほぼ不可

状態を追加するたびに関係する状態に条件記述追加が必要

BTの 対策

遷移条件と動作が独立

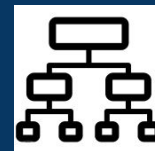
様々な再利用を助ける仕組み

※銀の弾丸ではない

BehaviorTree.CPP / Groot

BehaviorTree.CPP

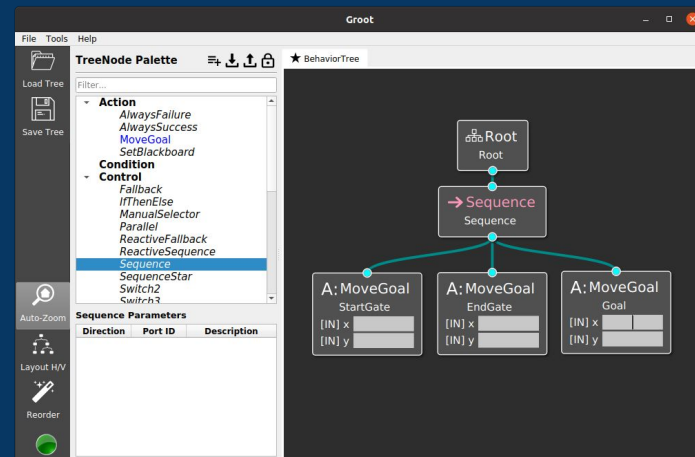
C++で記述されている行動計画ライブラリ
navigation2で採用されている



Groot

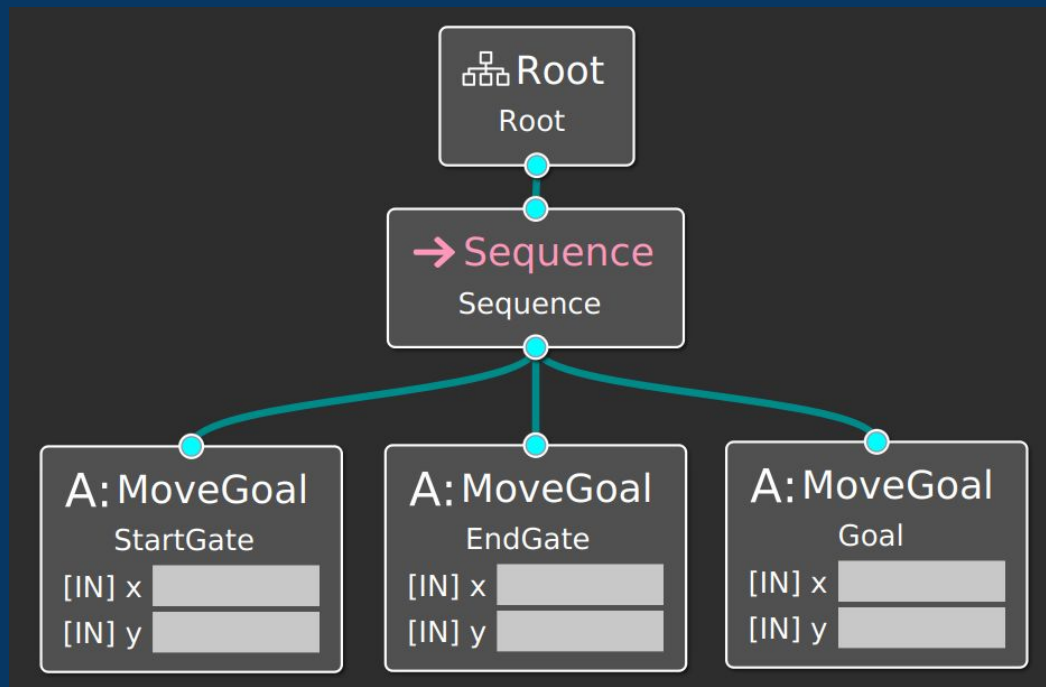
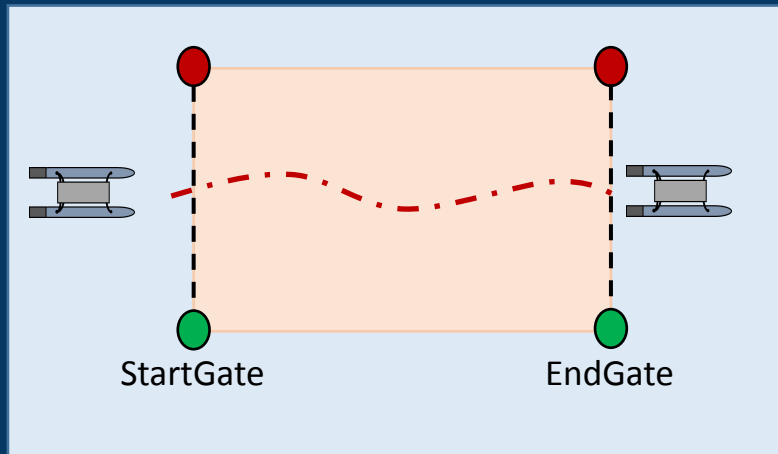
BehaviorTree.CPP用のGUIツール

機能：
モデリング
モニタリング
ログ再生



Demonstrate Navigation and Control

⇒ 2つのブイの真ん中を通り抜ける



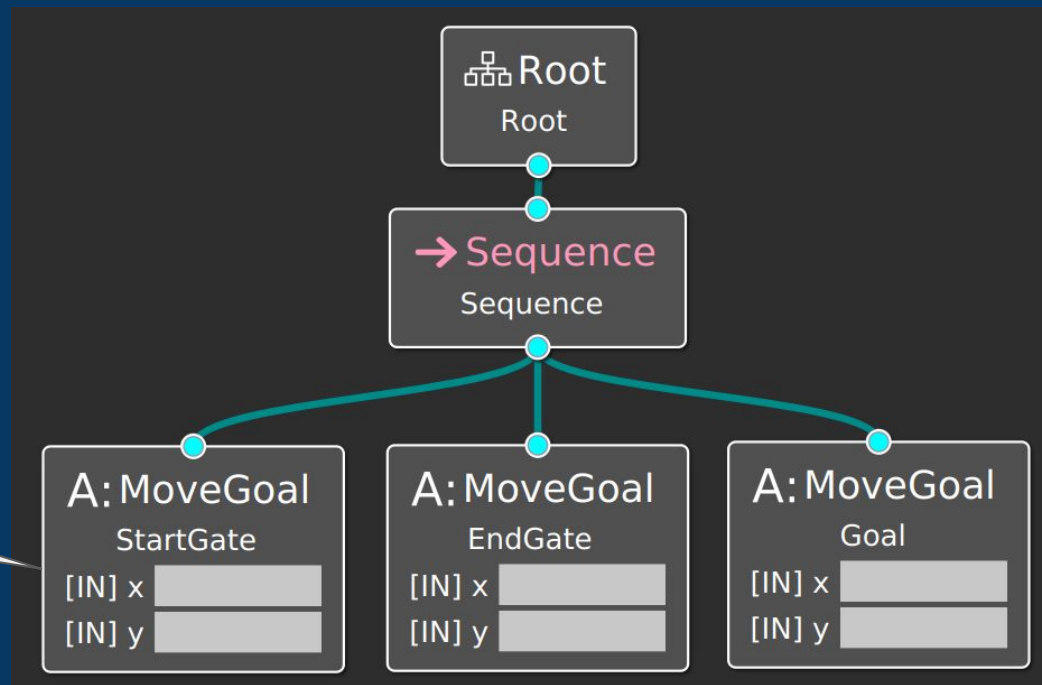
運用がイマイチ上手くいかない！

Demonstrate Navigation and Control

⇒ 2つのブイの真ん中を通り抜ける

真ん中の座標は
どこで計算する？

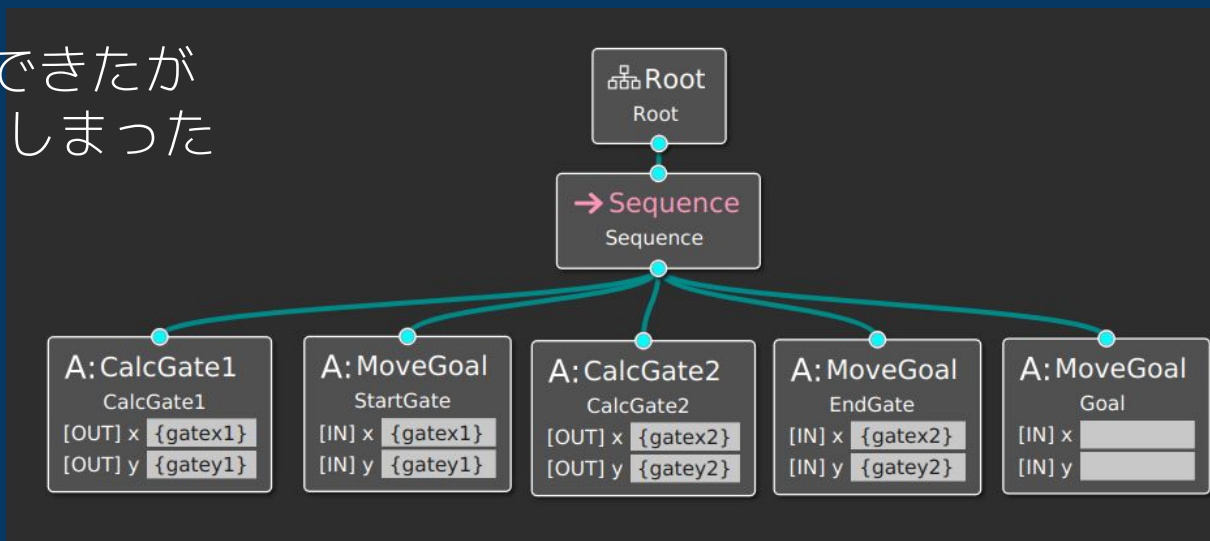
最悪手打ちだけどブイは移動するので
避けたい...



解決策？

ブイの真ん中を計算するノードを作成

MoveGoalノードは再利用できたが
新しい専用ノードが増えてしまった



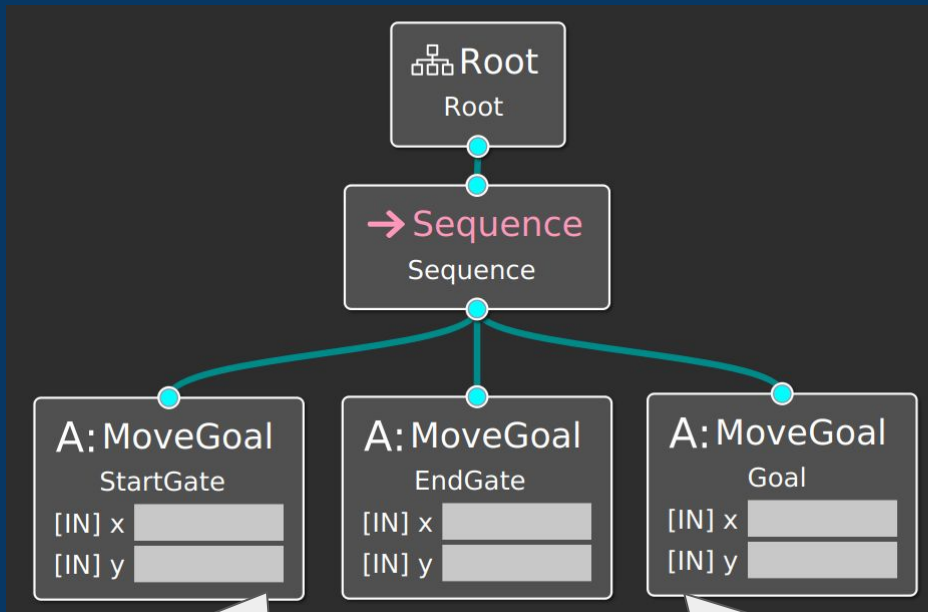
黒板機能：

出力した値は共有データ領域に保存，再利用できる

追加ツールを作って解決した！

こうすればいいのでは？（理想）

入力欄に計算式を打ち込む



本家リポジトリに改変を加える必要がある

⇒この方法は断念

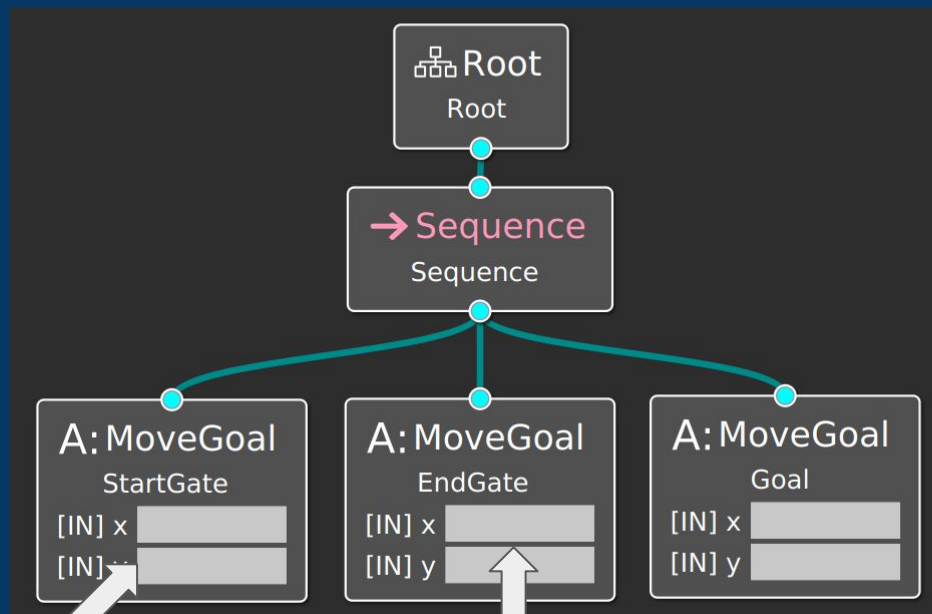
$x : (\text{buoy1_x} + \text{buoy2_x}) / 2$
 $y : (\text{buoy1_y} + \text{buoy2_y}) / 2$

$x : \text{between}(\text{buoy3}, \text{buoy4}).x$
 $y : \text{between}(\text{buoy3}, \text{buoy4}).y$

こうすればいいのでは？（現実）

外部に計算ツールを作り、
黒板機能で共有

本家ライブラリに干渉せず
実装可能

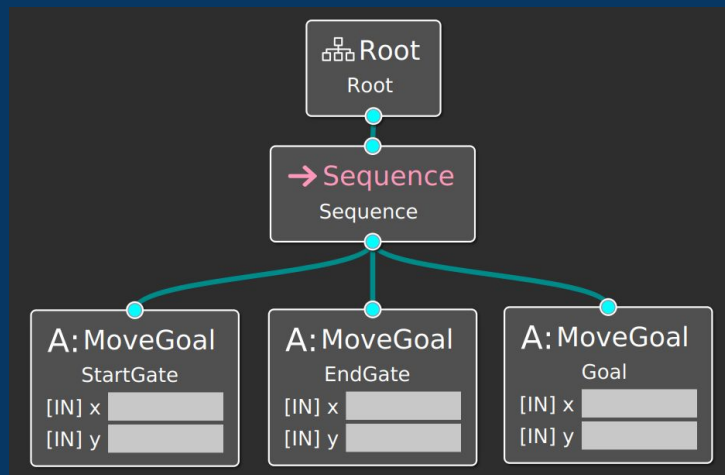


計算ツール

$$\begin{aligned}x1 &= (\text{buoy1_x} + \text{buoy2_x}) / 2 \\y1 &= (\text{buoy1_y} + \text{buoy2_y}) / 2\end{aligned}$$

$$\begin{aligned}x2 &= \text{between}(\text{buoy3}, \text{buoy4}).x \\y2 &= \text{between}(\text{buoy3}, \text{buoy4}).y\end{aligned}$$

実装したものがこちら！



```
plugins:
- package: robotx_behavior_tree
  name:
  - example_action
  - move_goal_action
behavior:
description:
package: robotx_bt_planner
path: behavior_trees/example.xml
blackboard:
- input: next_goal
  type: position
  eval: "add(1, 2*49)" # <- luaで評価
variables:
objects:
- uuid: 00000000
  attributes:
  - red
  - buoy
  pose:
  position:
  x: 10
  y: 0
  z: 0
  orientation:
  x: 0
  y: 0
  z: 0
  w: 1
- uuid: 00000001
  attributes:
  - green
  - buoy
```

設定ファイル

設定ファイルの読み込み

計算ツール

黒板を經由した情報の共有
一定周期ごとの情報更新

$$x1 = (buoy1_x + buoy2_x) / 2$$
$$y1 = (buoy1_y + buoy2_y) / 2$$

$$x2 = \text{between}(buoy3, buoy4).x$$
$$y2 = \text{between}(buoy3, buoy4).y$$

lua言語による柔軟な実行時評価

プラグイン機能

使用者がluaの関数を簡単に追加できるようにする

テスト機能

文法ミス, 意味的なミスの検出

⇒シナリオテストを準備中

一般化/リリース

OUXT関連機能をプラグインとして切り離す
リリースしてaptでバイナリを取得出来るように

メンバー募集

特に学生！



OUXT Polarisでは自律航行をやってみたい人を募集しています！

- ボールランチャー設計
- 電子回路設計
- ファームウェア実装
- プランナー実装
- 物体認識系の実装
- 運営（スポンサーとのやりとりなど）
- フィールド実験



上の内容等に興味のあるかたは  @OUXT_Polarisまで！