

ロボットシミュレータによる 遠隔操作体験

会津大学，株式会社FSK

Choreonoid | コレオノイド

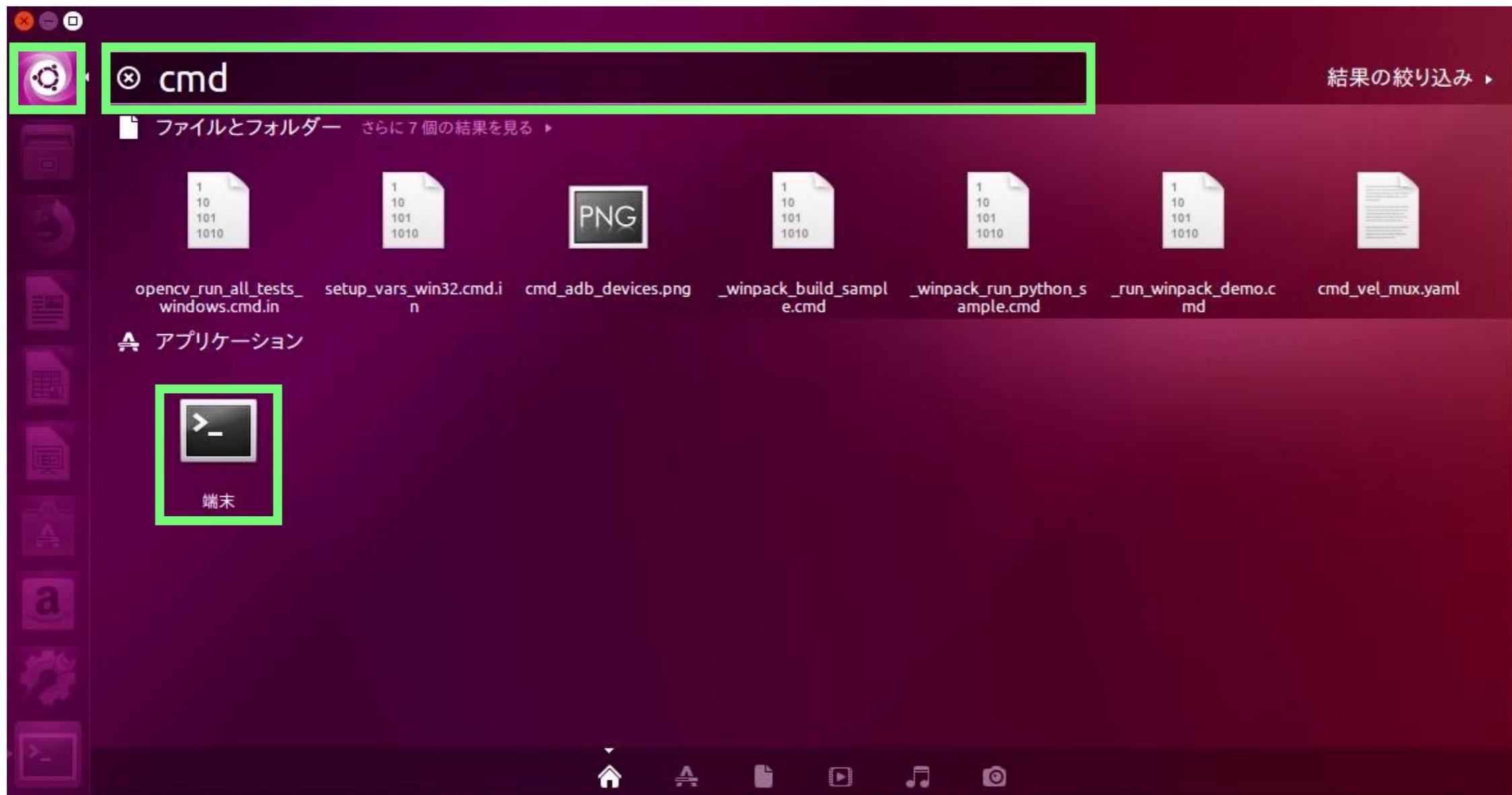
- ❖ Choreonoidホームページ <https://choreonoid.org/ja/>
- ❖ Choreograph（振り付けをする）と Humanoid（ヒューマノイド）を組み合わせて命名
- ❖ オープンソースのロボット用統合GUIソフトウェア
 - ❖ 国立研究開発法人・産業技術総合研究所（産総研）が開発
 - ❖ 現在は株式会社コレオノイドが引き継いで開発
- ❖ 動作振り付け機能， 動力学シミュレーション機能

動力学シミュレーション

- ❖ 物体の動作における力の影響をシミュレーションが可能
 - ❖ 自由落下, 衝突検知など
- ❖ ロボット動作のシミュレーションが可能
 - ❖ ロボットが床の上を走行する
 - ❖ ロボットがバルブを回す
 - ❖ ロボットに搭載されているカメラから状況を把握する
- ❖ 複数の物理エンジンが使用可能
 - ❖ AISTエンジン (標準搭載)
 - ❖ AGX Dynamics (有償)
 - ❖ Open Dynamics Engine

コマンドラインによる起動

- ❖ ランチャーから『コンピュータを検索』をクリックし、検索欄に『cmd』と入力して、『端末』をクリックし起動



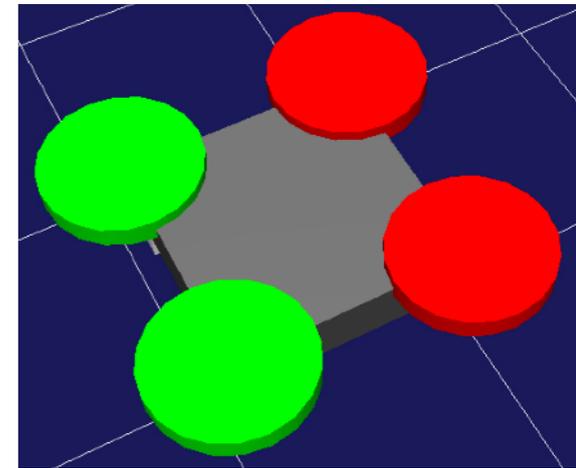
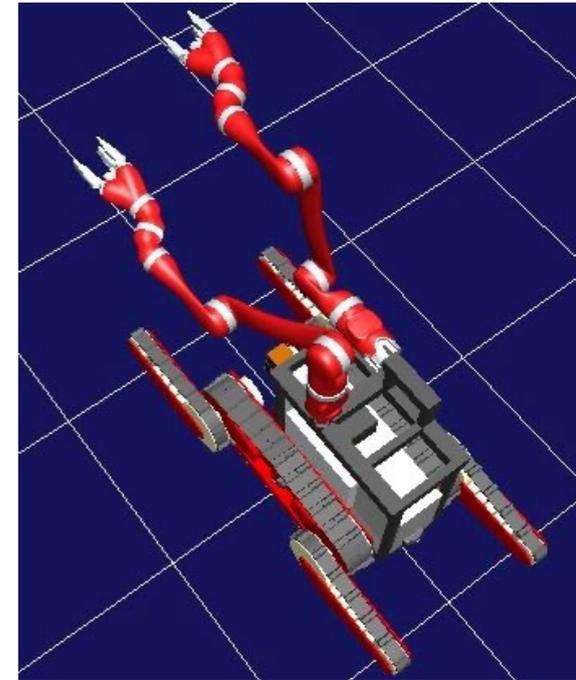
コマンドラインによる起動

- ❖ コマンドライン引数によるプロジェクトファイルの起動
\$ **choreonoid** [**プロジェクトファイルパス**]
- ❖ 例えば, ユーザ名が『**user1**』, プロジェクトファイルパスが『**/home/user1/choreonoid/sample/SimpleController/Tank.cnoid**』の場合は,
\$ **choreonoid /home/user1/choreonoid/sample/SimpleController/Tank.cnoid # 1行で書く**
- ❖ プロジェクトファイルを開いた後, Choreonoid上の  をクリックすることで, シミュレーションを開始できる
- ❖ Choreonoid上の  をクリックすることで, 停止できる

遠隔操作体験

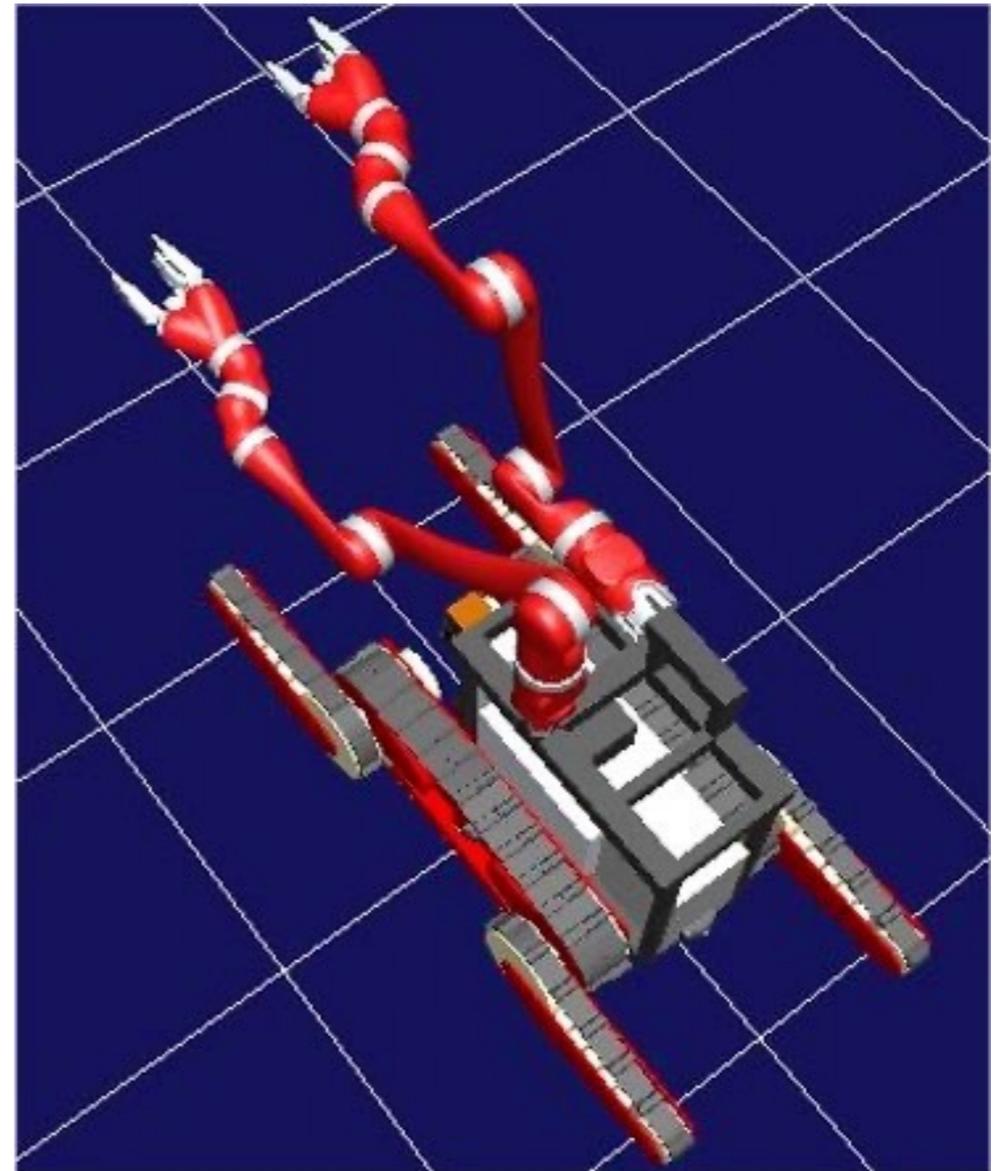
動力学シミュレーション

- ❖ シミュレーションで遠隔操作ロボットの操作訓練
 - ❖ 今回用意したコース: 5つ
 - ❖ ドローン操作コース: 2つ
 - ❖ クローラロボット操作コース: 3つ
- ❖ 各コースに練習用と本番用をそれぞれ用意
 - ❖ 練習用で十分に練習してから, 本番用に挑戦する
 - ❖ 練習用でカメラの視点を変更してロボットの動き, コースを確認する
 - ❖ 本番用はロボットからの視点のみで操作する



クローラロボット操作方法

- ❖ クローラロボットとして
ChoreonoidにサンプルとしてあるAizuSpiderを使用
- ❖ ログ(PS)ボタン押下で,
車体操作モードと
アーム操作モードの切り替え
- ❖ 切り替わり順
- ❖ 車体→アーム1→アーム2→
車体→...



クローラロボット操作方法 | 車体

❖ 車体の操作方法

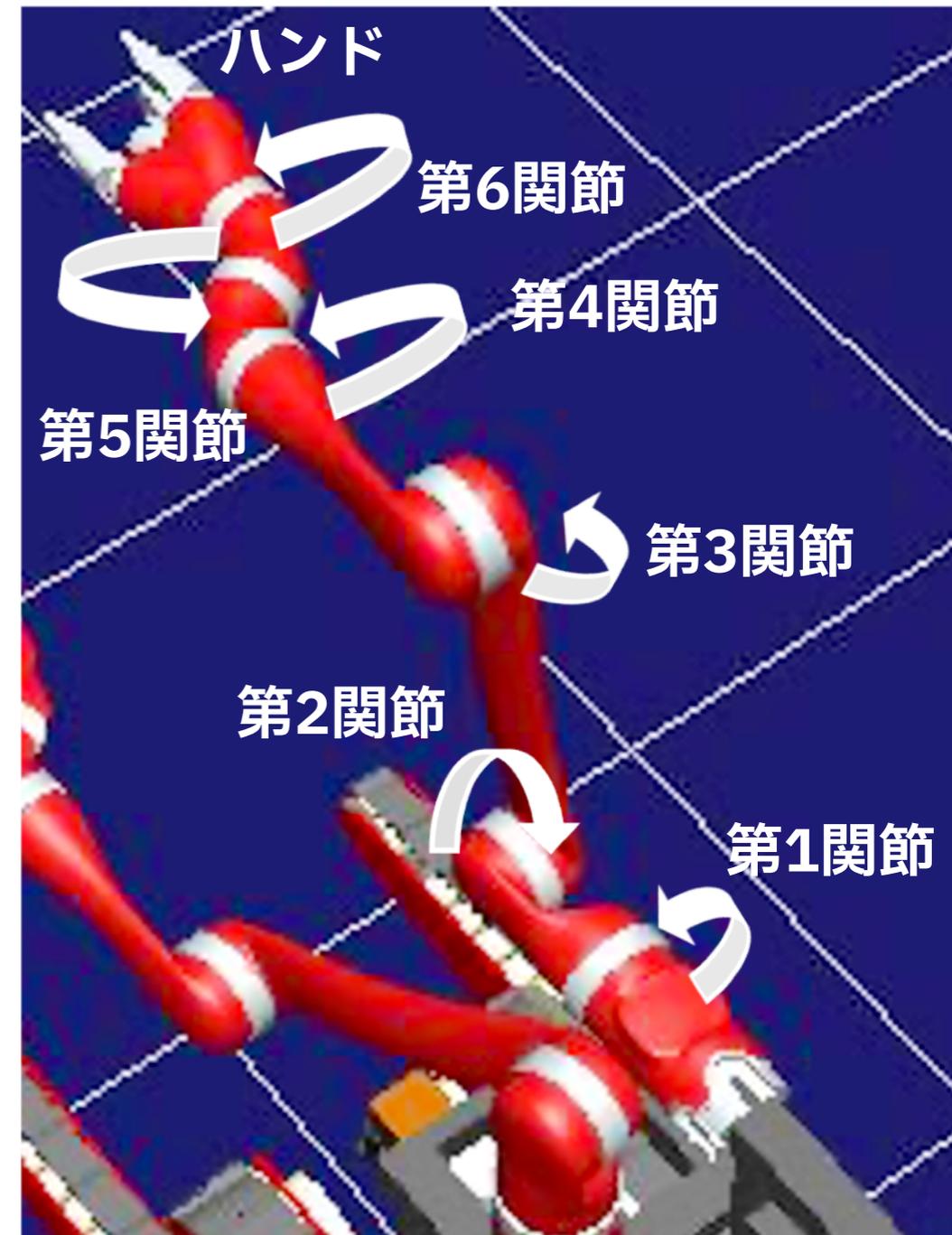
操作	動作
十字キー	クローラ操作 (前進, 後退, 旋回)
左スティック	クローラ操作 (前進, 後退, 旋回)
右スティック	全フリッパーの昇降
L2ボタン + 右スティック	左前フリッパーの昇降
R2ボタン + 右スティック	右前フリッパーの昇降
L1ボタン + 右スティック	左後フリッパーの昇降
R1ボタン + 右スティック	右後フリッパーの昇降
R3 (右スティック押し込み)	全フリッパーの傾きを揃える



クローラロボット操作方法 | アーム

❖ アームの操作方法

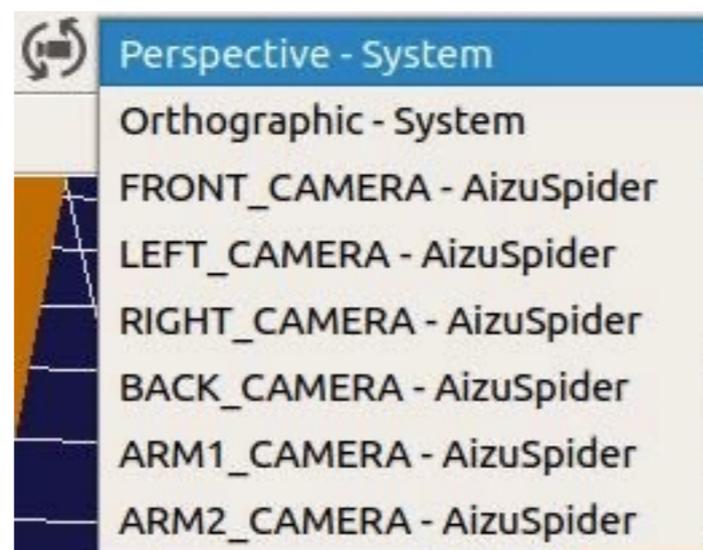
操作	動作
左スティック 左/右	アーム第1関節の操作
左スティック 上/下	アーム第2関節の操作
右スティック 上/下	アーム第3関節の操作
右スティック 左/右	アーム第4関節の操作
□/○ボタン	アーム第5関節の操作
L1/R1ボタン	アーム第6関節の操作
L2/R2ボタン	ハンドの開閉



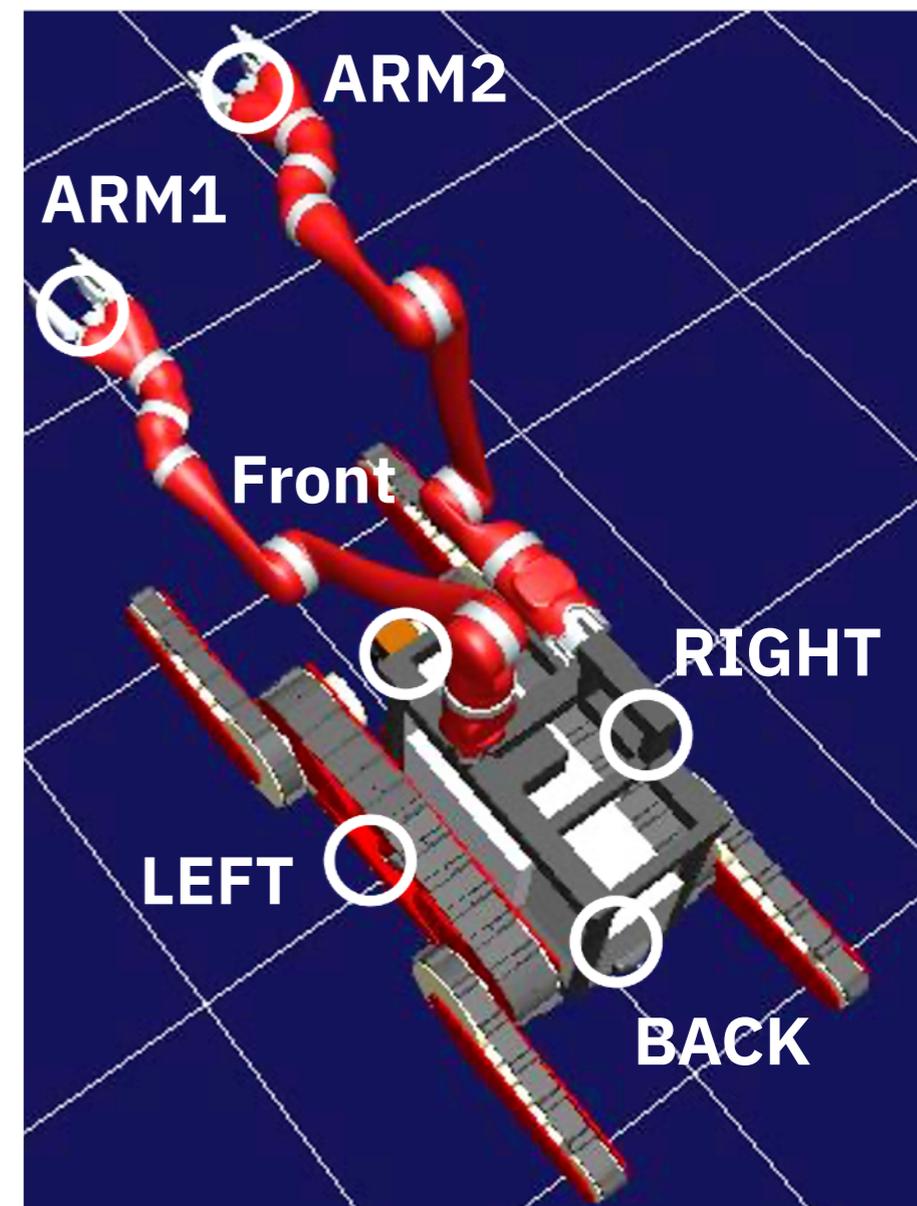
クローラロボット操作方法 | カメラ

- ❖ AizuSpiderにはカメラが搭載されており、描画用カメラ選択から変更できる

操作	搭載場所
Front_CAMERA	車体の前方に搭載
LEFT_CAMERA	車体の左側に搭載
RIGHT_CAMERA	車体の右側に搭載
BACK_CAMERA	車体の後方に搭載
ARM1_CAMERA	左のアームに搭載
ARM2_CAMERA	右のアームに搭載



**ツールバーの
描画用カメラから
使用するカメラ選択**



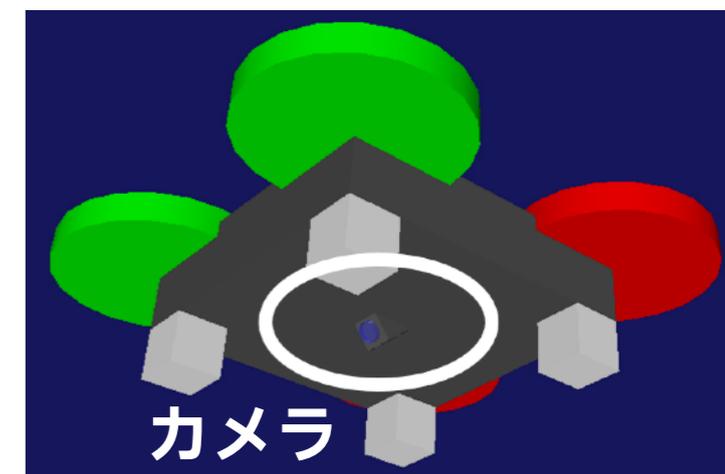
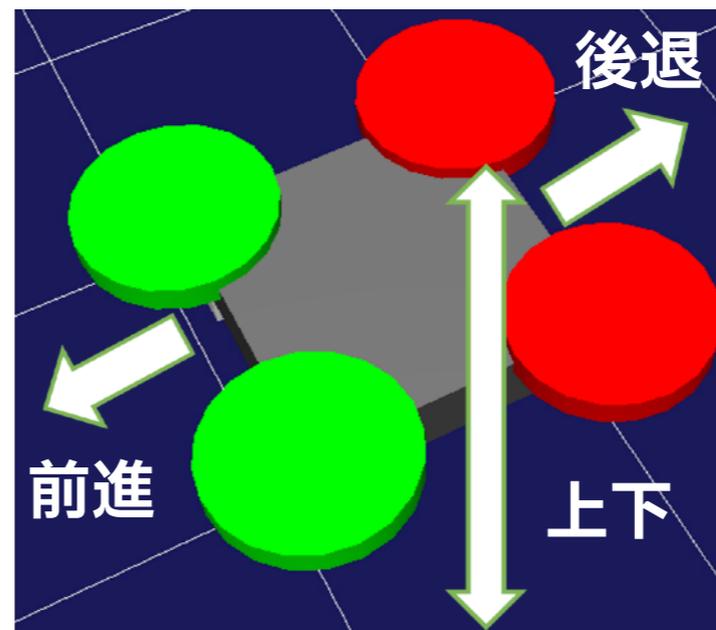
Perspectiveカメラの視点変更

- ❖ Perspective（透視投影）カメラ
 - ❖ **遠近感のついた画像**を表示
- ❖ Perspectiveカメラは**視点変更が可能**
 - ❖ 左ボタンドラッグ: 視点の**回転**
 - ❖ ホイールスクロール: 視点の**拡大・縮小**
 - ❖ ホイールドラック: 視点の**平行移動**
- ❖ 視点を変更してロボットの動きやコースを確認してください

ドローン操作方法 | 機体

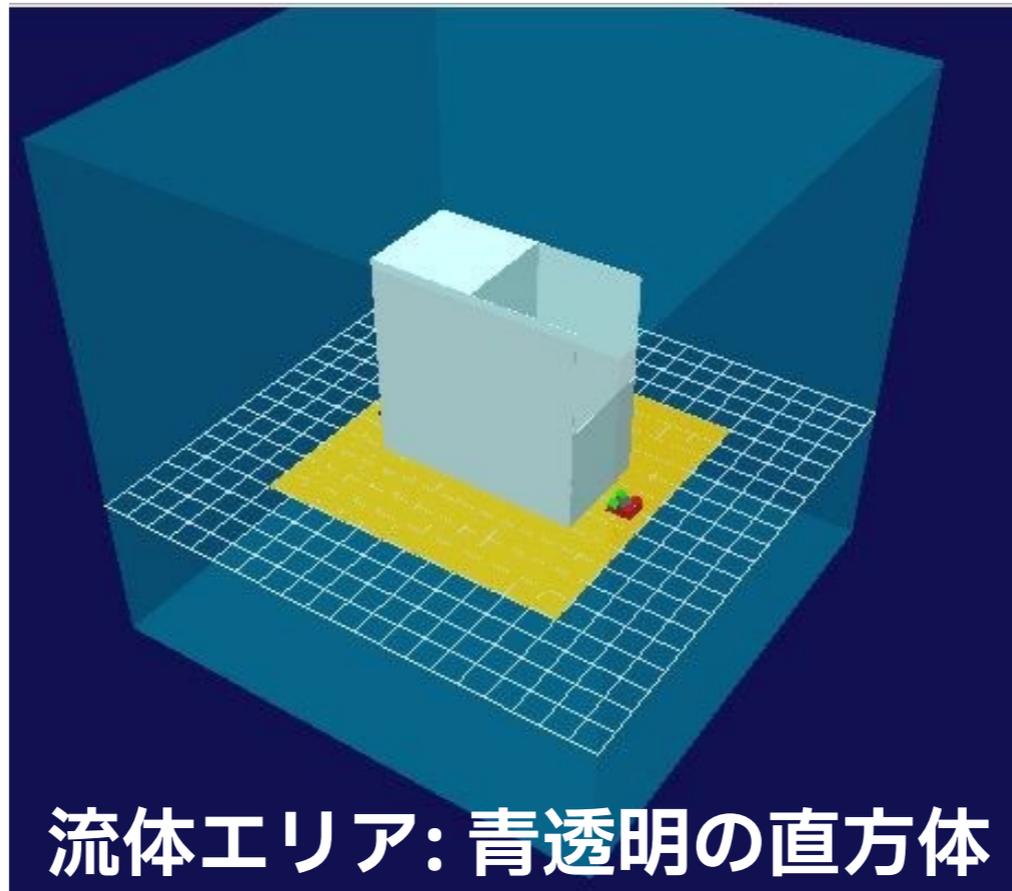
- ❖ JAEA製作のHAIROWorldPluginのUAVモデルを使用
<https://github.com/k38-suzuki/hairo-world-plugin>
- ❖ PS4コントローラで操作可能
- ❖ UAVモデルの下部に向きを変更できるカメラが接続
- ❖ ドローン操作方法

操作	動作
左スティック 左/右	前進, 後退
左スティック 上/下	左右の旋回
右スティック 上/下	ドローンの高度上下
十字キー上下	カメラの向き変更
L1/L2	ズーム+/ズーム-



ドローン操作方法 | 流体エリア

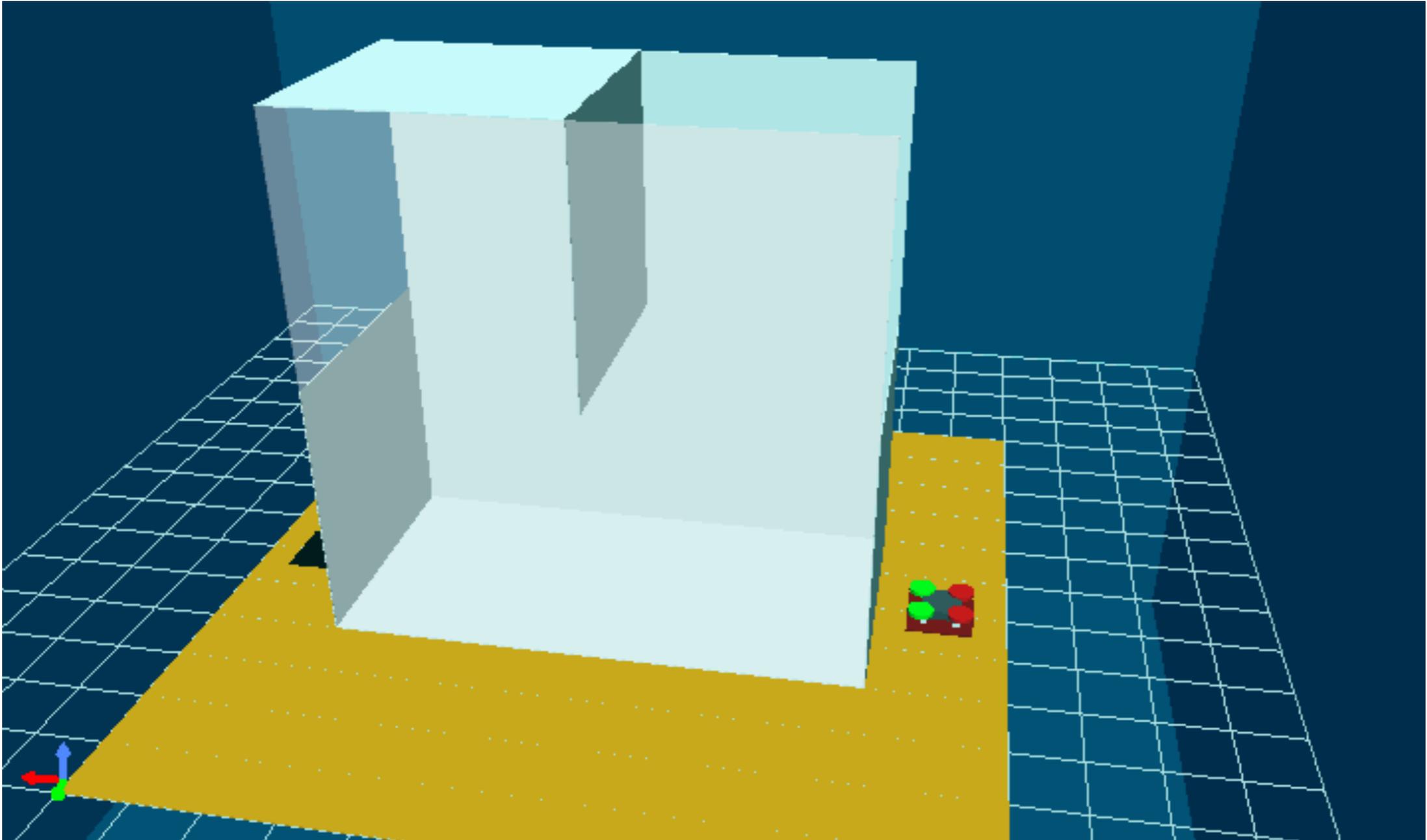
- ❖ ドローンは流体エリア内のみ移動可能
- ❖ 流体エリア
 - ❖ 水・空気が存在するエリアを仮想的に定義した領域
- ❖ ドローンはエリア外に移動するとコントロールを失う



流体エリア: 青透明の直方体

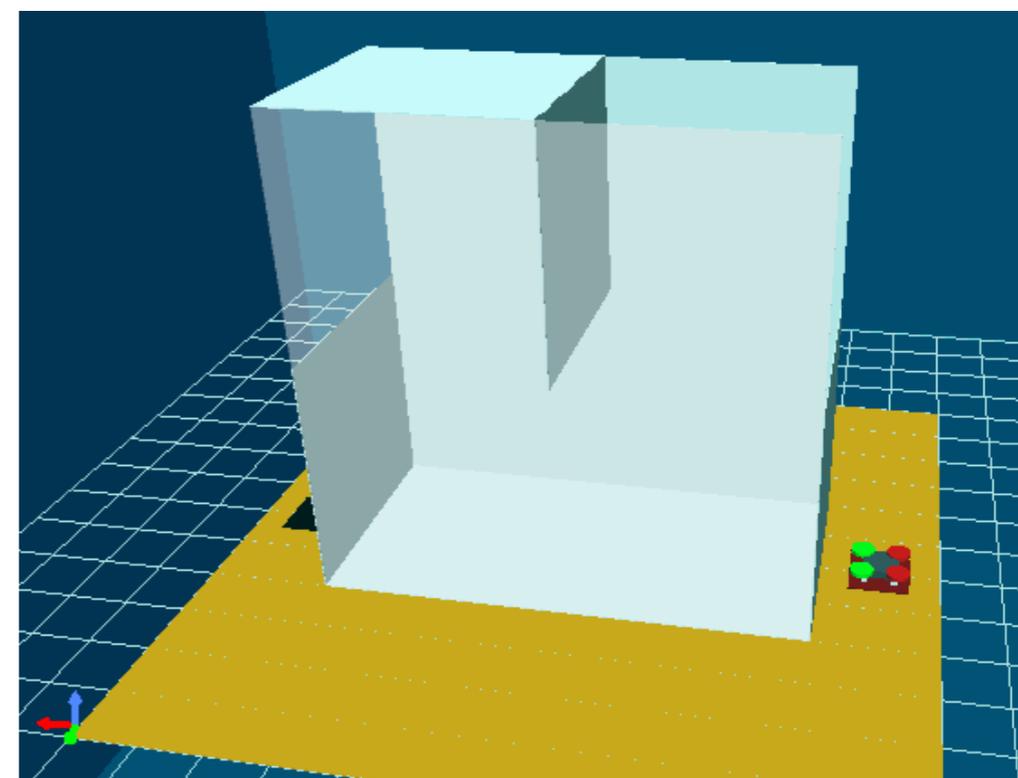
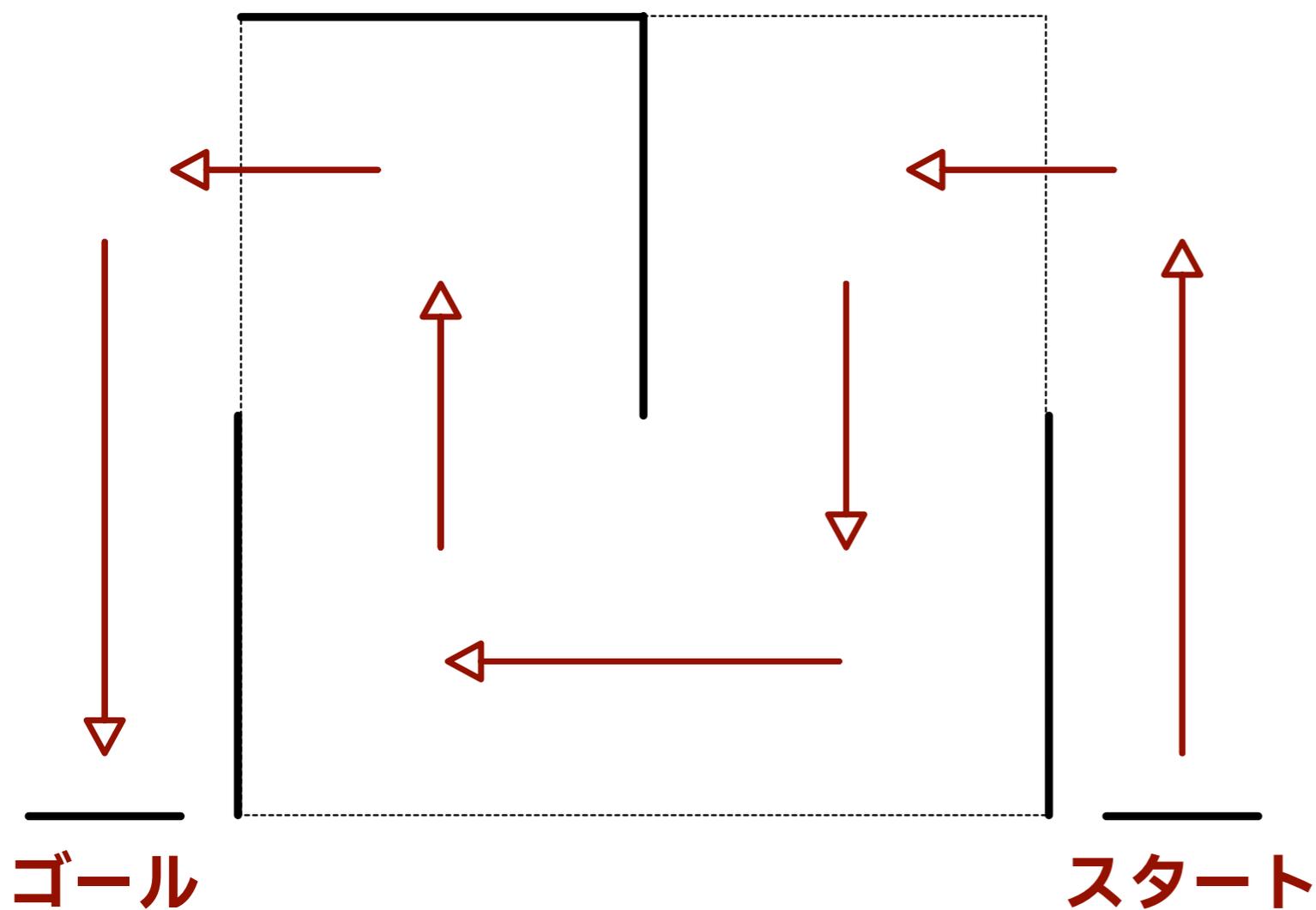
ドローンコース1 | コース外観

❖ ミッション: 障害物回避飛行



ドローンコース1 | コース説明

- ❖ スタート地点からゴール地点まで指定されたコースを通り移動
- ❖ スタートからゴールまでの時間を計測



ドローンコース1 | 起動方法

- ❖ プロジェクトファイルを，以下の場所に配置しています

~/ICTROBOT/DroneCourse1/

- ❖ 練習用プロジェクトファイル

\$ **choreonoid ~/ICTROBOT/DroneCourse1/DroneCourse1-1.cnoid**

\$ **choreonoid ~/ICTROBOT/DroneCourse1/DroneCourse1-2.cnoid**

- ❖ 本番用プロジェクトファイル

\$ **choreonoid ~/ICTROBOT/DroneCourse1/DroneCourse1-3.cnoid**

- ❖ 練習用プロジェクトファイルを起動して，操作訓練を行う

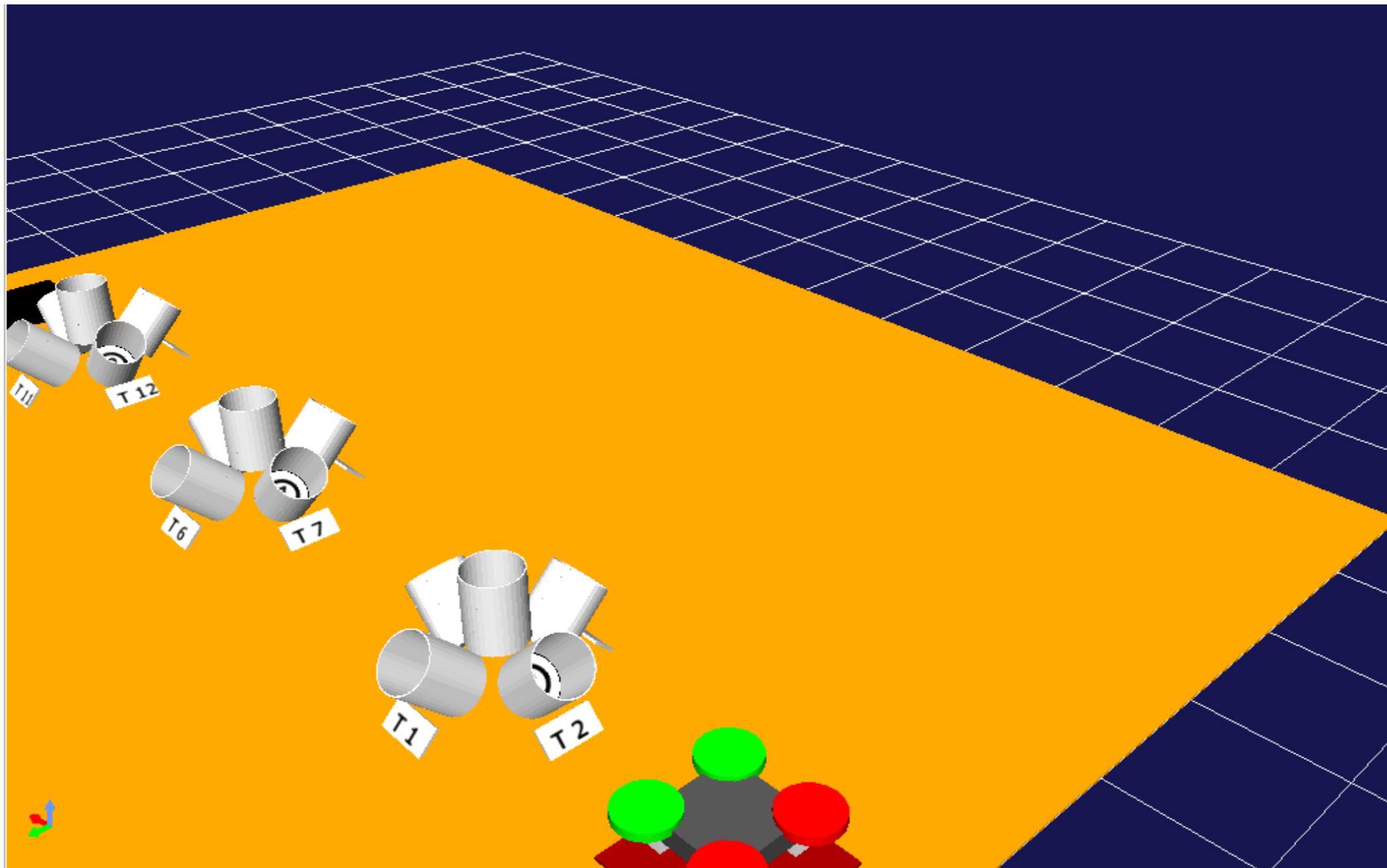
- ❖ 十分に練習できたら，本番用プロジェクトファイルを起動して操作してください

ドローンコース1 | 結果

❖ シミュレーション時間: [秒]

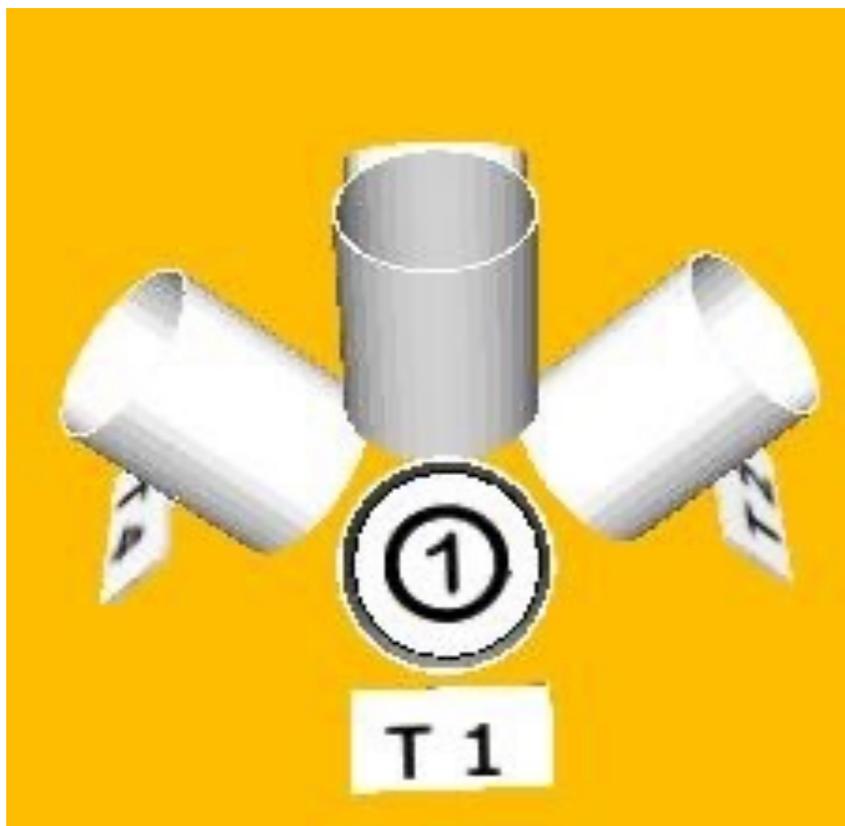
ドローンコース2 | コース外観

- ❖ ミッション: Position (ピンポイントの位置合わせ)

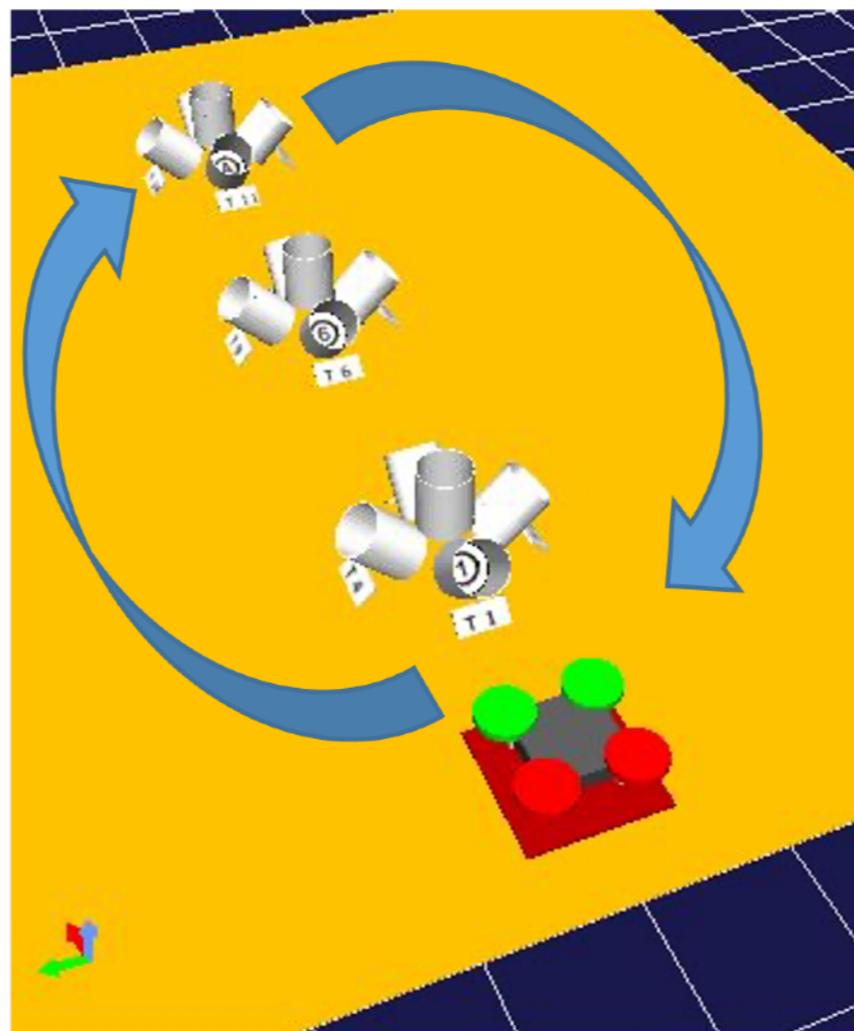


ドローンコース2 | コース説明

- ❖ ドローンを操作してバケツの中の文字を確認
 - ❖ 文字とバケツの中の黒い淵も確認
- ❖ T1～T15の順番でバケツの中を確認してスタート地点に帰還



中の文字と中の黒い淵を
見える位置に移動する



全バケツの中を確認後、
スタート地点に戻る

ドローンコース2 | 起動方法

- ❖ プロジェクトファイルを，以下の場所に配置しています

~/ICTROBOT/DroneCourse2/

- ❖ 練習用プロジェクトファイル

\$ **choreonoid ~/ICTROBOT/DroneCourse2/DroneCourse2-1.cnoid**

- ❖ 本番用プロジェクトファイル

\$ **choreonoid ~/ICTROBOT/DroneCourse2/DroneCourse2-2.cnoid**

- ❖ 練習用プロジェクトファイルを起動して，操作訓練を行う

- ❖ 十分に練習できましたら，本番用プロジェクトファイルを起動して操作してください

ドローンコース2 | 結果

❖ シミュレーション時間: [秒]

T1:	[秒]
T2:	[秒]
T3:	[秒]
T4:	[秒]
T5:	[秒]

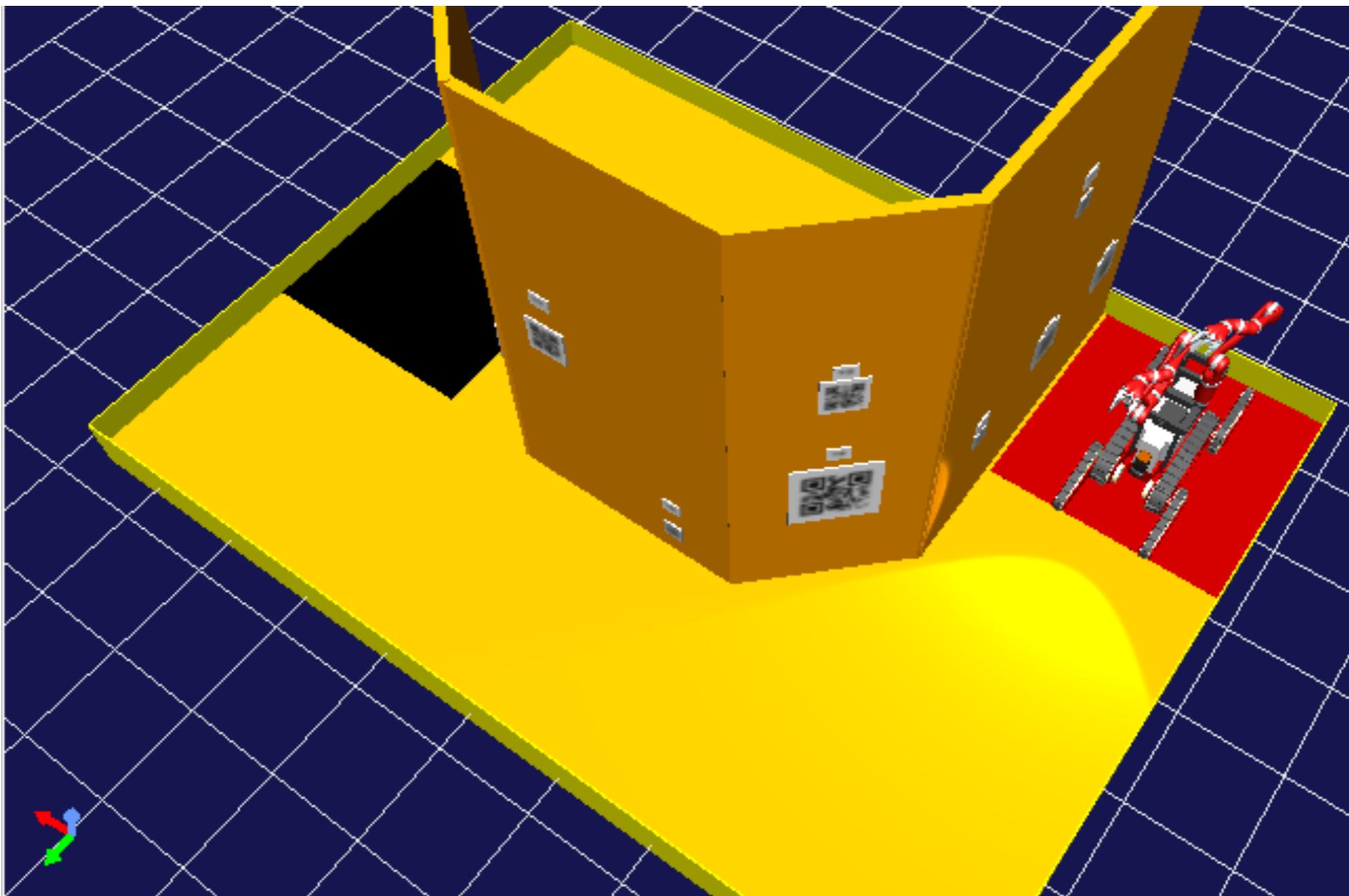
T6:	[秒]
T7:	[秒]
T8:	[秒]
T9:	[秒]
T10:	[秒]

T11:	[秒]
T12:	[秒]
T13:	[秒]
T14:	[秒]
T15:	[秒]

※ T5: T1~4の中央, T10: T6~9の中央, T15: T11~14の中央に存在

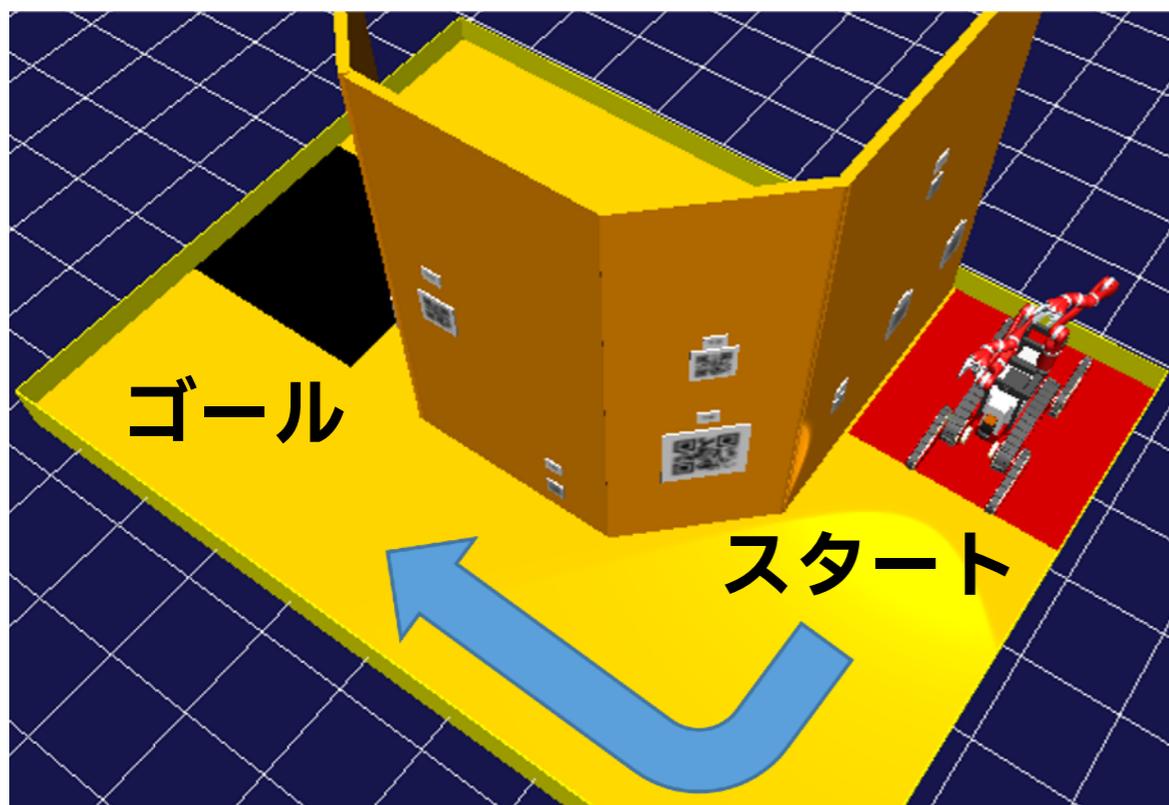
クローラロボットコース1 | コース外観

❖ ミッション: 壁面検査



クローラロボットコース1 | コース説明

- ❖ クローラロボットを操作して、壁面に貼ってあるQRコードを確認しながら、スタート(赤)からゴール(黒)まで移動する
- ❖ QRコードの確認は、各自のスマートフォンを使用する



クローラロボットコース1 | 起動方法

- ❖ プロジェクトファイルを，以下の場所に配置しています

~/ICTROBOT/CrawlerRobotCourse1/

- ❖ 練習用プロジェクトファイル

```
$ choreonoid ~/ICTROBOT/CrawlerRobotCourse1/  
CrawlerRobotCourse1-1.cnoid # 1行で書く
```

- ❖ 本番用プロジェクトファイル

```
$ choreonoid ~/ICTROBOT/CrawlerRobotCourse1/  
CrawlerRobotCourse1-2.cnoid # 1行で書く
```

- ❖ 練習用プロジェクトファイルを起動して，操作訓練を行う

- ❖ 十分に練習できたら，本番用プロジェクトファイルを起動して操作してください

クローラロボットコース1 | 結果

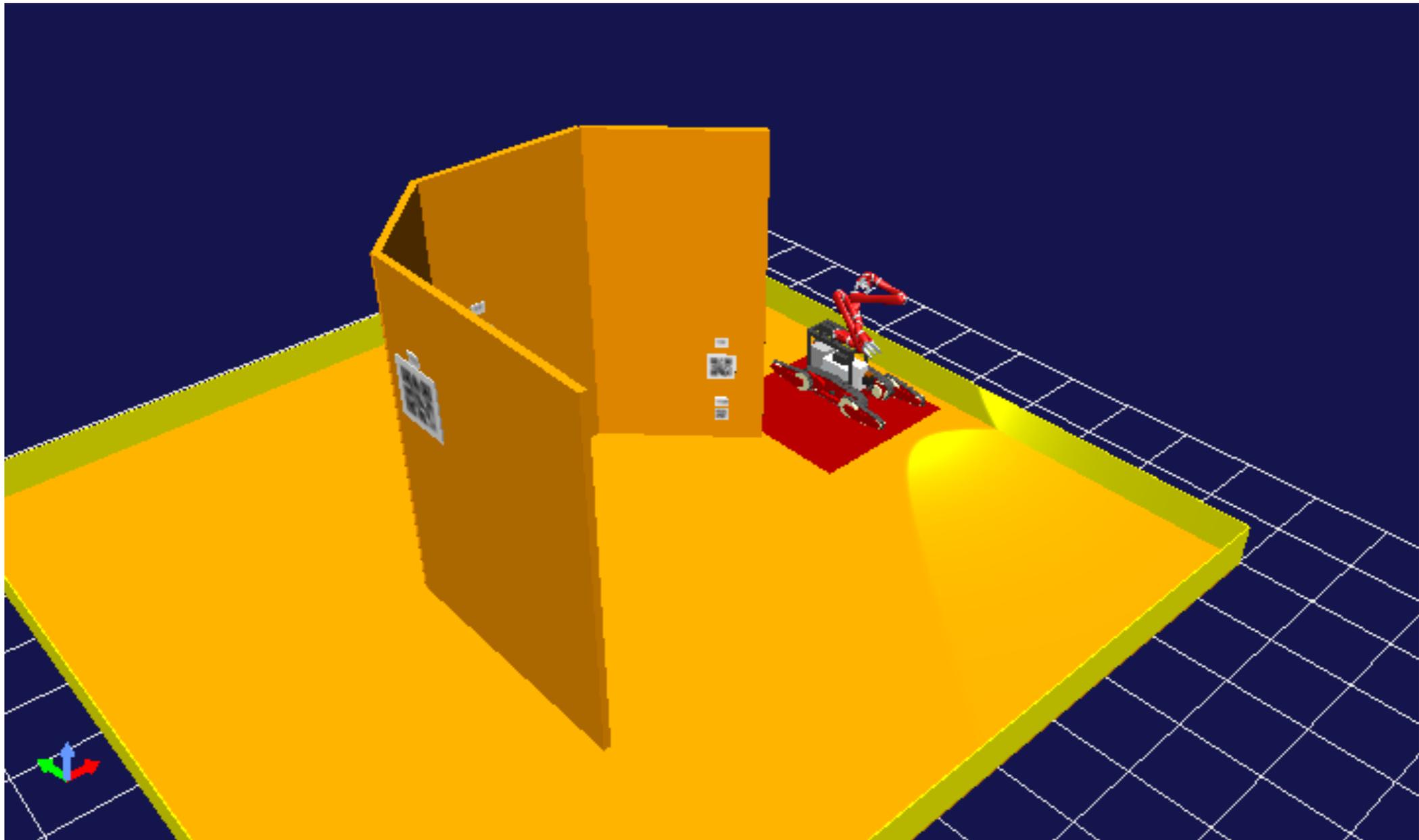
❖ シミュレーション時間: [秒]

T1:	
T2:	
T3:	
T4:	
T5:	

T6:	
T7:	
T8:	
T9:	
T10:	

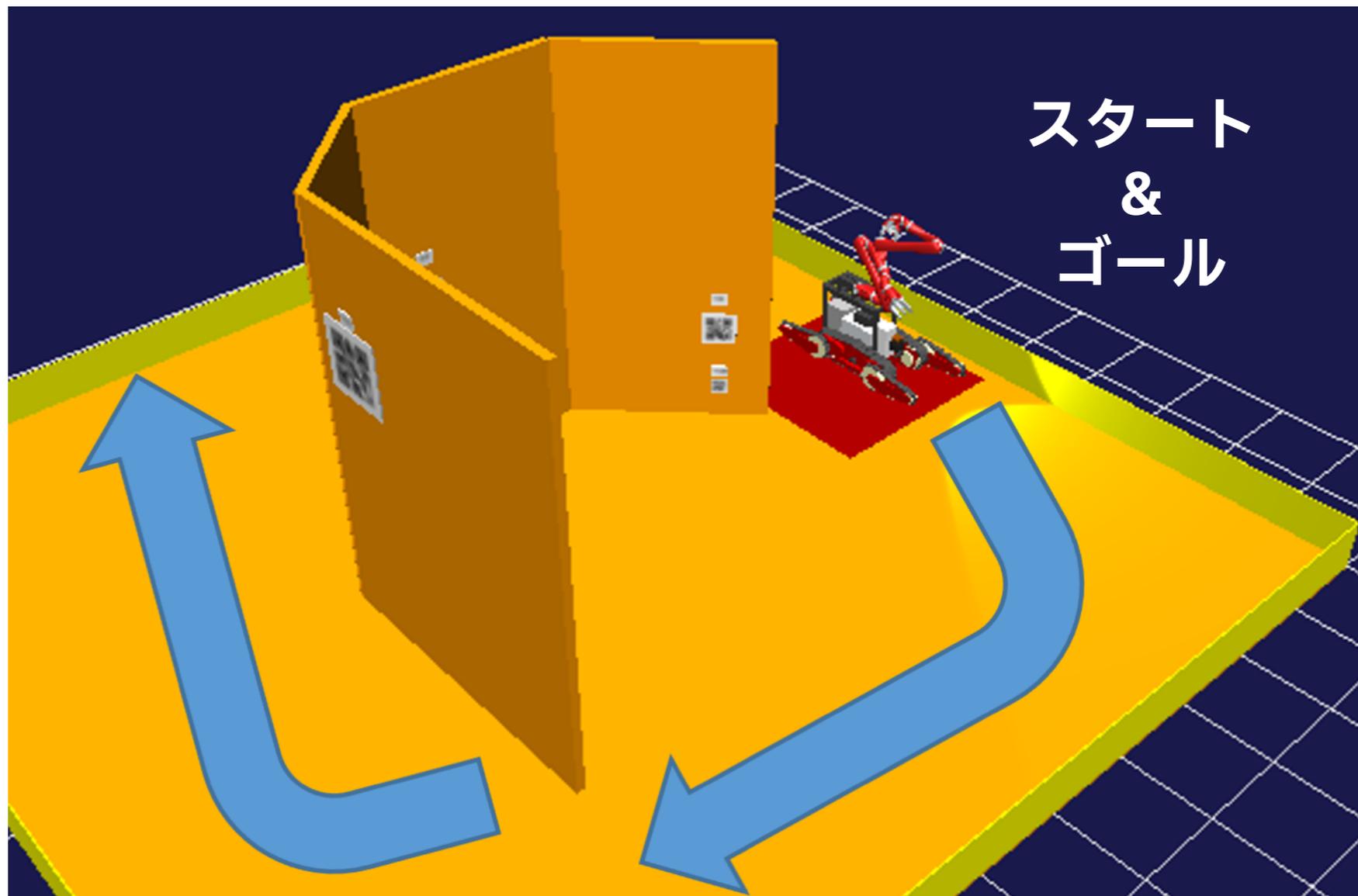
クローラロボットコース2 | コース外観

- ❖ ミッション: 壁面検査 (1週 / 裏面あり)



クローラロボットコース2 | コース説明

- ❖ 壁面の表面と裏面のQRコードを確認しながら、コースを一周する
- ❖ QRコードを確認は、各自のスマートフォンを使用する



クローラロボットコース2 | 起動方法

- ❖ プロジェクトファイルを，以下の場所に配置しています

~/ICTROBOT/CrawlerRobotCourse2/

- ❖ 練習用プロジェクトファイル

```
$ choreonoid ~/ICTROBOT/CrawlerRobotCourse2/  
CrawlerRobotCourse2-1.cnoid # 1行で書く
```

- ❖ 本番用プロジェクトファイル

```
$ choreonoid ~/ICTROBOT/CrawlerRobotCourse2/  
CrawlerRobotCourse2-2.cnoid # 1行で書く
```

- ❖ 練習用プロジェクトファイルを起動して，操作訓練を行う

- ❖ 十分に練習できたら，本番用プロジェクトファイルを起動して操作してください

クローラロボットコース2 | 結果

❖ シミュレーション時間: [秒]

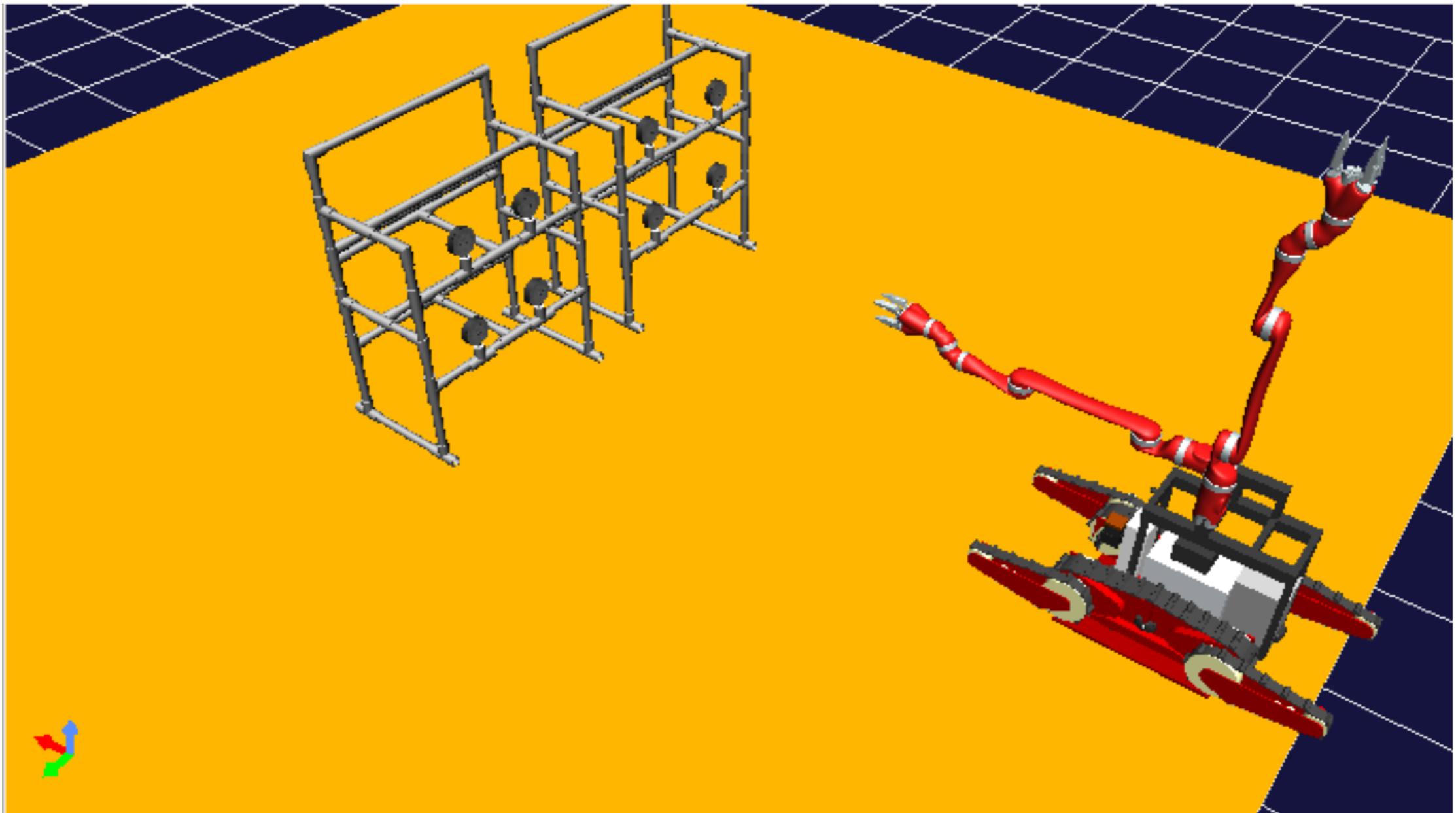
T1:	
T2:	
T3:	
T4:	
T5:	

T6:	
T7:	
T8:	
T9:	
T10:	

T11:	
T12:	
T13:	
T14:	
T15:	

クローラロボットコース3 | コース外観

❖ ミッション: メータ読み取り



クローラロボットコース3 | コース説明

- ❖ クローラロボットを操作してパイプのメータの値を確認
- ❖ すべて確認できたらスタート地点に戻る



クローラロボットコース3 | 起動方法

- ❖ プロジェクトファイルを，以下の場所に配置しています

~/ICTROBOT/CrawlerRobotCourse3/

- ❖ 練習用プロジェクトファイル

**\$ choreonoid ~/ICTROBOT/CrawlerRobotCourse3/
CrawlerRobotCourse3-1.cnoid # 1行で書く**

- ❖ 本番用プロジェクトファイル

**\$ choreonoid ~/ICTROBOT/CrawlerRobotCourse3/
CrawlerRobotCourse3-2.cnoid # 1行で書く**

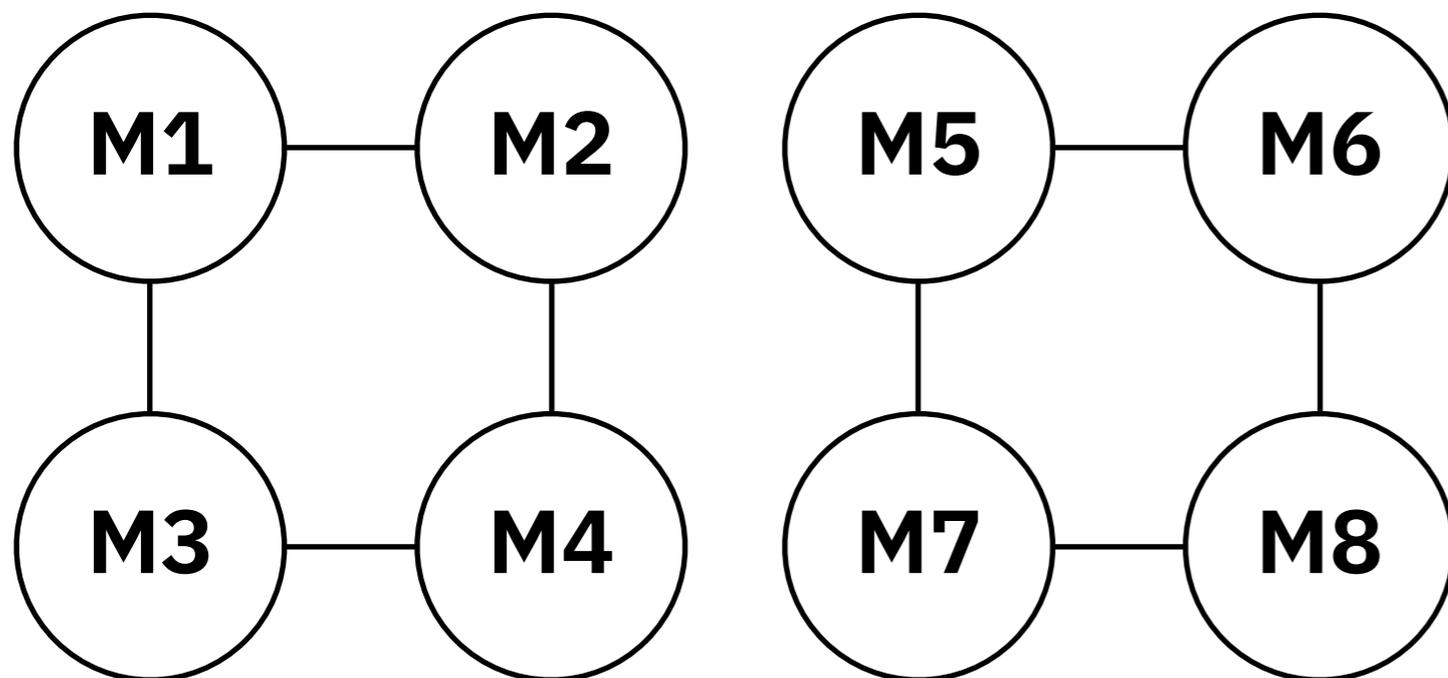
- ❖ 練習用プロジェクトファイルを起動して，操作訓練を行う

- ❖ 十分に練習できたら，本番用プロジェクトファイルを起動して操作してください

クローラロボットコース3 | 結果

❖ シミュレーション時間:

[秒]



M1:	
M2:	
M3:	
M4:	
M5:	
M6:	
M7:	
M8:	