

# 箱庭技術の教育への活用事例

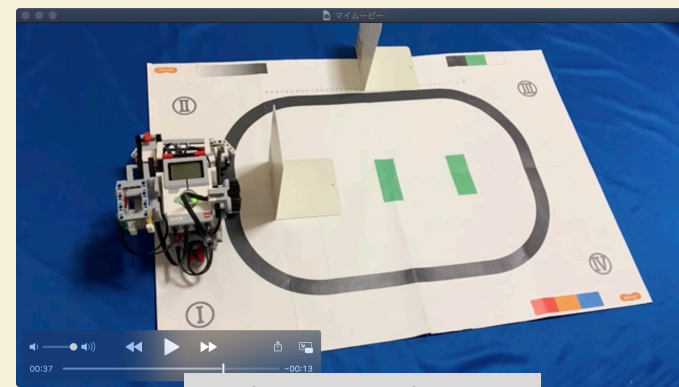
株式会社チェンジビジョン

久保秋 真

# 実機で演習していたのを箱庭で代替した

## • 2019年まで

- 個人またはグループで演習
- 実機（Mindstorms EV3 + EV3RT）を利用
- Astah\* Proで設計モデルを作成、変換による実装



実機で実行

## • 2020年

- 各自の自宅で演習（対面は実施できず）
- 各自に実機を用意できない（代わりに箱庭を利用）
- 実機と同じように設計、実機と同じように実装



箱庭上の環境で実行

# ソフトウェア開発の演習

(早稲田大学 大学院1年向け)

# ソフトウェア開発演習

- 半期（15回）の講義

- UML、オブジェクト指向、実装はすでに講義・演習している

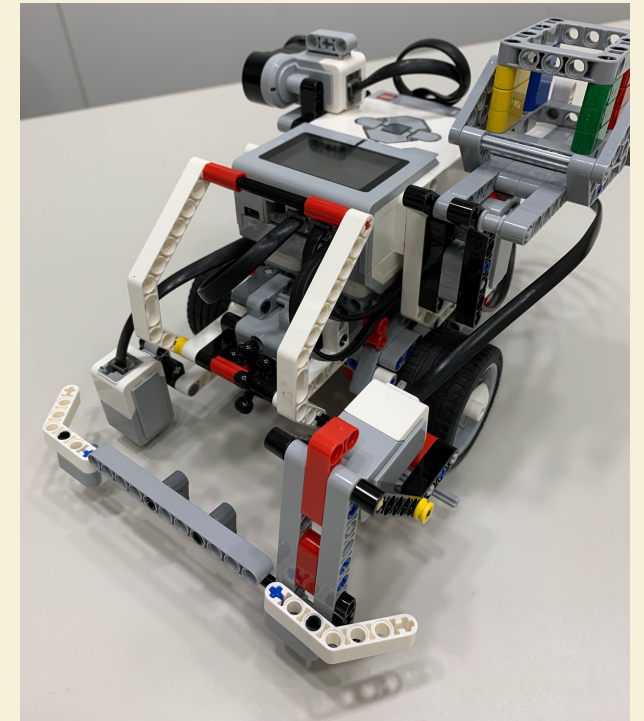
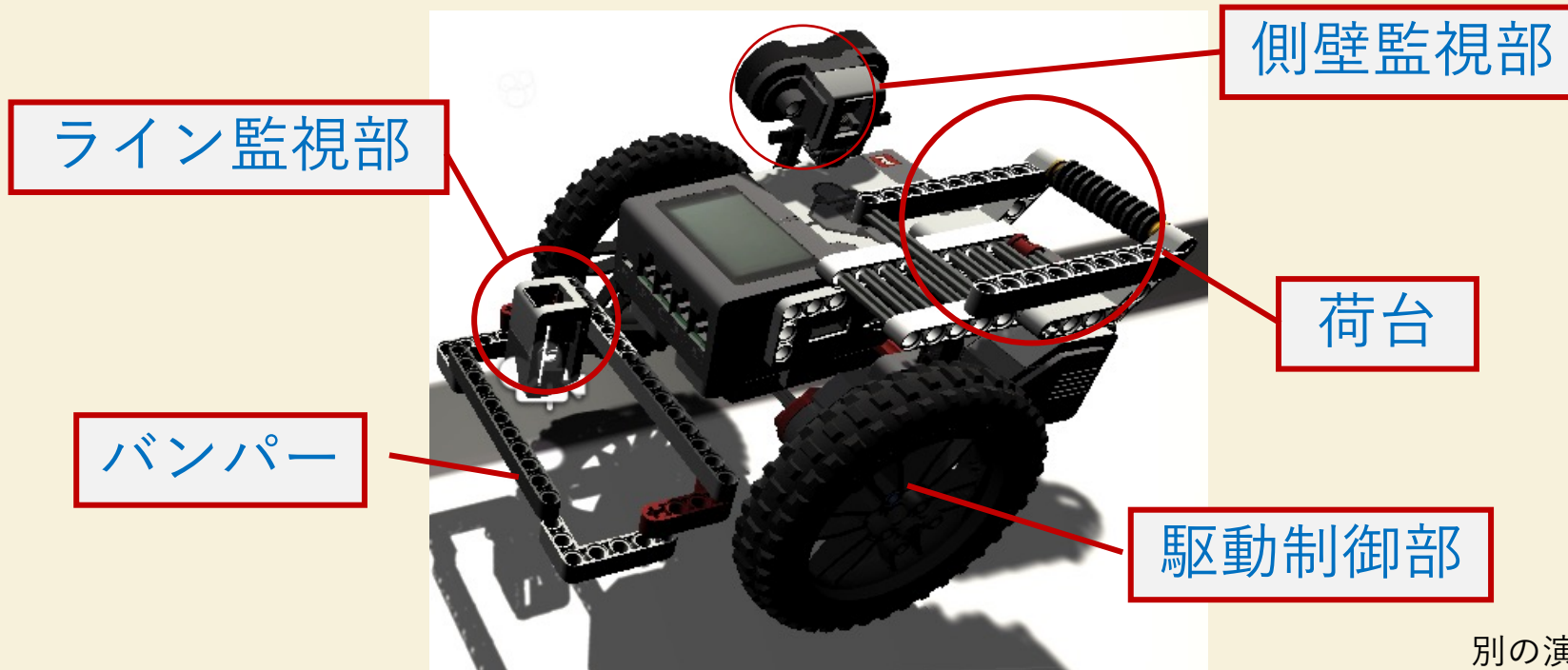
- 設計と実装の対応づけを重視

- 荷物を運搬するシステムの開発
- 開発プロセスに沿って開発
  - 設計が形骸化しないための分析設計の活用方法を演習を通じて学ぶ
- 適用領域の用語による設計モデルを実装まで紐づける
  - 変換による実装※

※変換ルールを使って設計モデルからコードを得る作業。  
手動で変換することで、モデル変換の役割や意味を実感してもらう。

# 使用する機器

- 荷物を運搬する自動搬送ロボット
  - 駆動制御部による走行
  - 荷台、ライン監視部、側壁監視部、バンパーを装備



別の演習 : <https://youtu.be/A24WsSucrKw>

5

# 演習の課題

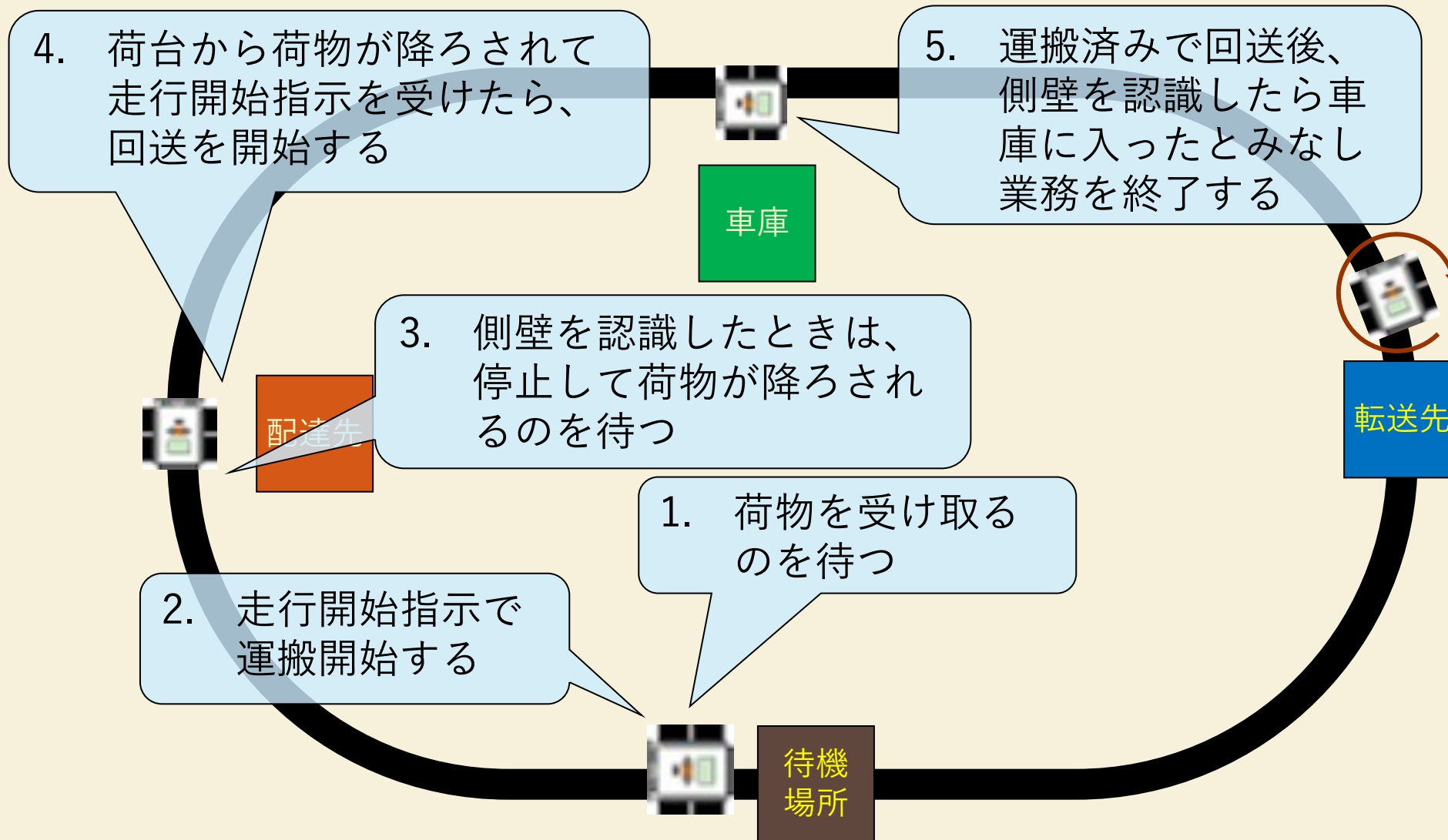
- 自動搬送システム

- 状況に応じて「配達」「転送」「回送」の各業務を処理する

- 配達業務

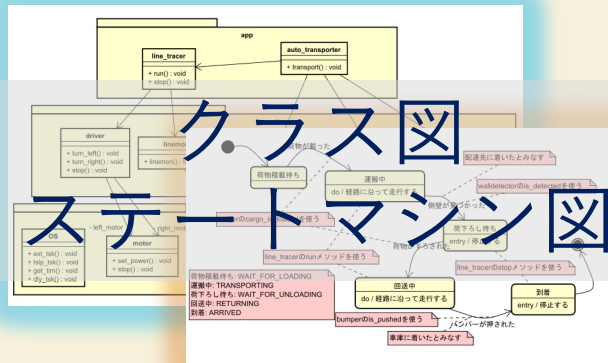
- 待機場所で荷物を受けるのを待つ
- 走行開始指示を受け取ったら、配達先へ荷物を運搬する
- 配達先に到着したら停止する
- 荷物が下ろされるのを待つ
- 荷物が下ろされたら、車庫へ回送して停止する

# 業務の流れ（のうち配達業務を抜粋）



# 実装方式の設計

- 「変換による実装」の方式を決める
  - 内部設計はこの方式を使うことを前提にする
  - 設計の前に、この方式を使う実装を演習しておく



クラス図  
状態マシン図

図からコードへの変換ルール

```
void auto_transporter_transport(void) {  
    switch(p_state) {  
        // 略  
        case WAIT_FOR_UNLOADING:  
            line_tracer_stop();  
            if (!carrier_cargo_is_loaded()) {  
                p_state = RETURNING;  
            }  
            break;  
        case RETURNING:  
            line_tracer_run();  
            if (bumper_is_pushed()) {  
                p_state = ARRIVED;  
            }  
            break;  
        case ARRIVED:  
            line_tracer_stop();  
            break;  
    }  
}
```

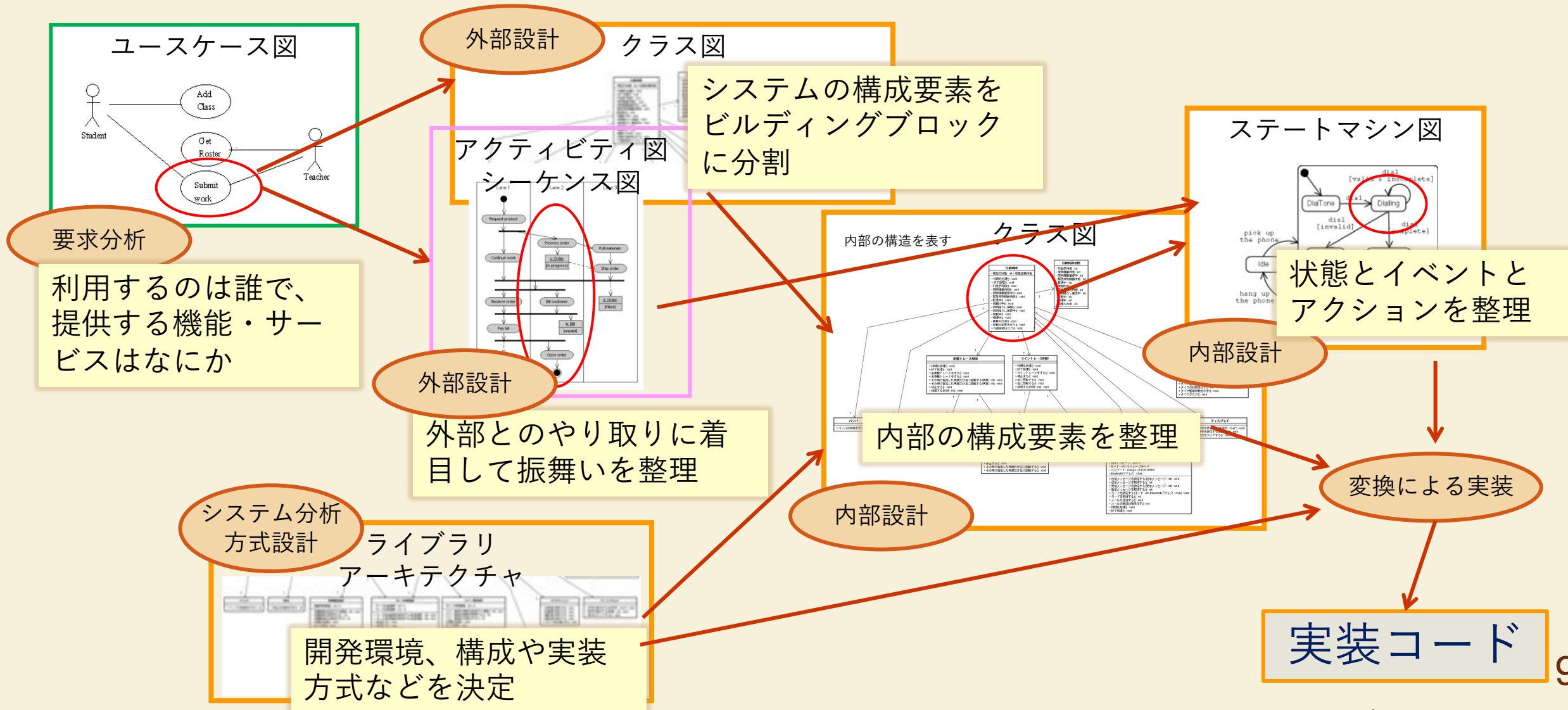
変換による  
実装

```
void auto_transporter_transport(void) {  
    switch(p_state) {  
        // 略  
        case TRANSPORTING:  
            line_tracer_run();  
            if (walldetector_is_detected()) {  
                p_state = WAIT_FOR_UNLOADING;  
            }  
            break;  
        case WAIT_FOR_UNLOADING:  
            line_tracer_stop();  
            break;  
    }  
}
```

実装コード



# 開発プロセスに沿って開発



# 動作例の紹介

- 実機でのデモ
- 箱庭でのデモ

## まとめ（早稲田大学）

- 箱庭を使えたことで、各自の自宅で演習できた
  - 実機が配布できなくても、動作を確かめられた
- 組み込みでも、アプリ開発なら実施できる
  - 依存しているのは、RTOSやセンサ・モータのAPIだから
  - 走行、荷物の上げ下ろし、壁の認識ができると、あとは業務のアプリケーションを作ることができるから

# プロジェクト演習

(日本大学生産工学部3年)

# プロジェクト演習

- クォーター（週1回、2コマ）の講義
  - プログラミングもモデリングもあまり慣れていない
- グループによる開発の体験を重視
  - 観光スポットを遊覧するサービスの開発
  - グループで実際に動くものを作る体験
    - 目的・目標・成果を共有する
    - 頼んでやってもらう、頼むとき渡す・受け取る成果物の役割の実感
- 開発プロセスに沿って開発
  - UMLの図に精度は求めない（困ってもらって必要性を感じてもらう）

# 演習のお題

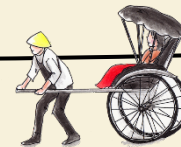
- 大学周辺の観光スポットを巡るサービスの提供
- 専用道を走るオートライドで乗って巡る
  - 利用者は予め遊覧先を選んで乗車する
  - スポットに着いたら利用者は遊覧、再び乗ったら次のスポットへ移動
  - 観光しないスポットはスキップする



K2蒸気機関車



旧大沢家住宅



# 動作例の紹介

- 実機でのデモ
- 箱庭でのデモ

## まとめ（日本大学）

- 箱庭を使えたことで、各自の自宅で演習できた
  - 実機が配布できなくても、動作を確かめられた
- 仲間で互いに工夫して助け合う以前のつまづき
  - 環境を用意して動かすこと自体が大変だった
  - プログラミングに不慣れだと、環境が複雑なだけでわからない感じに支配されてしまう



箱庭を使って演習して得られたこと

# 実は、学生の反応は厳しいものだった…

- 実機ほどには「動いた！感」がない…
  - やはり、実物にかなわないか…
- PCの性能が低いと、動かすのが億劫になる
  - Zoomで画面共有すると、さらに重くなる（デモができない）
- 環境構築ができない
  - 演習を始めるまでに至らず（そして履修をあきらめてしまう）
    - WSLがインストールできない、シェルが使えない、…etc.
- 動かないときの原因がわからない
  - config ファイルの誤りなど、問題の切り分けができない

# 実機を使う「体験」の代わりにはならない

- 設計やコード作成の演習としては便利
  - 箱庭は、どこでも動作検証に使えるのがよい、とのこと
- 実機のときのように夢中にはなってくれない
  - 学生の使うPCでは、動かすのが億劫
    - シミュレーションの環境が起動、動作とも重いようだ
  - 1度動かすと、あとはあまり動かす気持ちにならない
    - 実機ほどは外乱の影響がないので、動きに変化がない
  - 他の人の動かしている様子が、横目に入ってこない
    - 自分のPCで動かすので、他との違いや自分の不出来に気づかない

# 実機では軽んじられる側面は、試しやすい

- 実機では感じない・観測できないことを試せる
  - 短い時間に起こるセンサーのオフ・オン等を確認められる
    - シミュレーションのティックを変えて顕在化させると気づかせるなど
- 実機では環境のせいにするが、そうはしにくい
  - 周囲の明るさが違うから、コースにシワがあったから…など
    - そのような変化がない環境で動作できるので、別の原因だと気づける

# まとめ

# まとめ

- 箱庭は、システムの設計・実装の演習に使える
  - 実機のないテレワーク環境でも、演習できる
- 「動かす体験」には向かない場合がある
- 環境構築は開発に慣れていない学生には難しい
- 動作環境は学生のPCやネットでは負荷が高い

環境構築や動作負荷の問題は、森さんの発表にあるように、  
今後はもっと改善されるでしょう  
箱庭の改善と発展に期待しています！