

CS 知的システム演習

# ライントレーサーロボット の試行

---

松吉 俊

# 準備するもの

---

- 組み立てたレゴロボット
- USBケーブル
- ノートPC (with 有線ネットワーク)
- Moodleからダウンロードした関連ファイル
- [ゴールを作る担当のみ] 9797基本セット

**各自が記録した  
基本セット番号とノートPC番号のものを利用する**

# leJOSへの移植

---

- 演習8で作成した"学籍番号\_R1.java"を移植し、紙のマップmap1上でライトレーサーロボットを走らせる
- MyRobotForNXT.javaからファイル名を変更する時に、**内部のクラス名も合わせて変更すること**

```
public class T12CS099_R1 extends Robot {  
    /** leJOS での起動用 main 関数 */  
    static void main(String[] args)  
    {  
        try {  
            // 時間計測  
            Long time = System.currentTimeMillis();  
            // ロボットオブジェクトを生成して実行  
            new T12CS099_R1().run();  
            :  
        }  
    }  
}
```

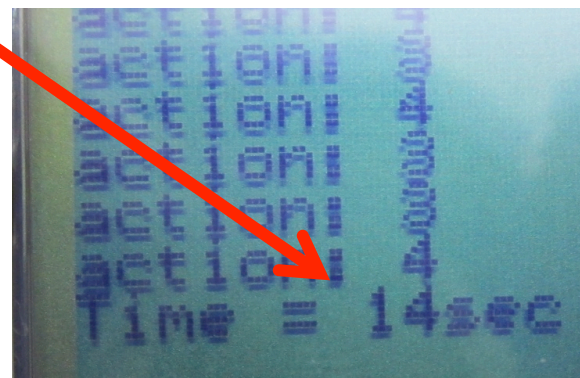
# 移植と実行

---

1. デスクトップに作った各自のフォルダー内に、  
"学籍番号\_R1.java"を置く
  - USBメモリーやオンラインストレージを利用する
  - フォルダー内に、新しく配布したRobotクラスもあるはず
2. 「コマンドプロンプト」上でコンパイルする
3. "学籍番号\_R1"クラスをロボットに転送する
4. ロボットにおいてプログラムを実行する

# “学籍番号\_R1”クラス

- 学習したQテーブルと、センサーからの入力に従って、ゴールを目指すライトレーサー
- (おそらく) map1専用
- 超音波センサーの前方20cm未満に物体があると、**動作時間**を表示して終了する
- **緊急時は、超音波センサーの前に手をかざして停止させる**
- 最終手段として、四角い黒いボタンを長押しして強制終了させる

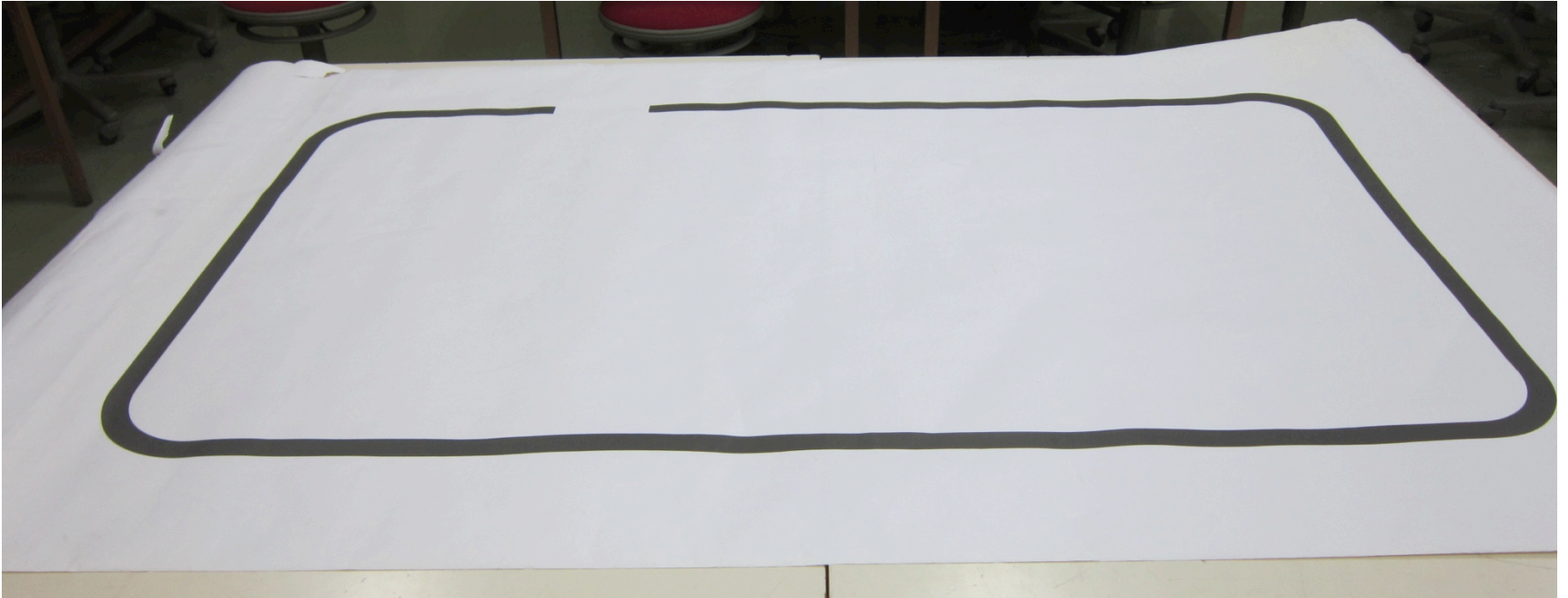


**ゴールまでの時間を競う**

# 紙のマップmap1

---

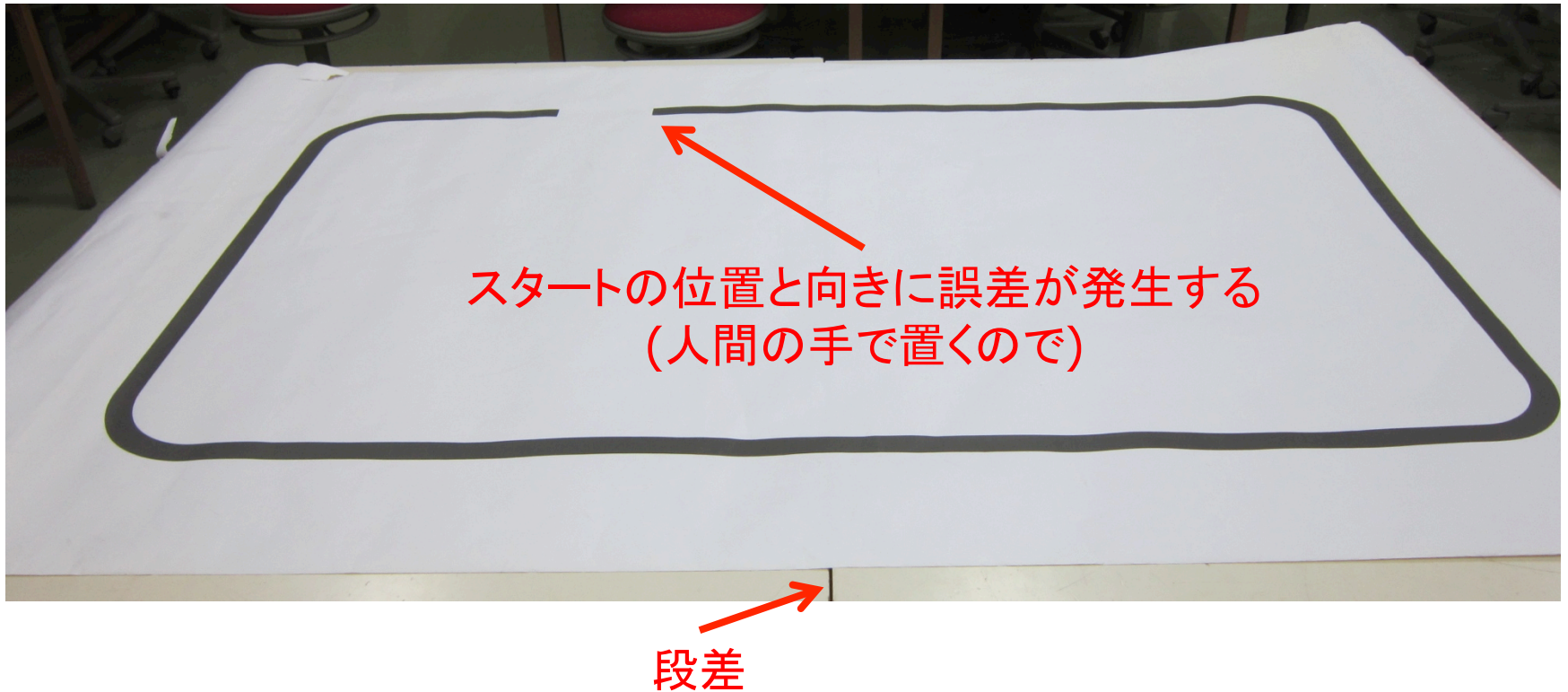
- 机の上に広げる



# 紙のマップmap1

---

## ● 机の上に広げる

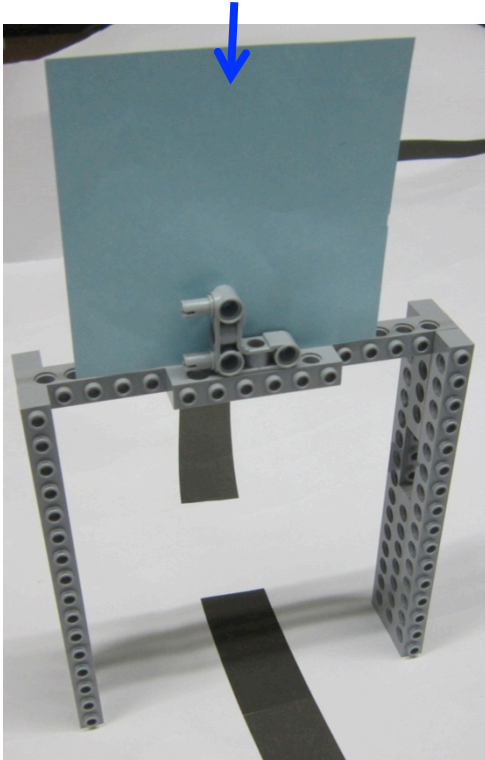




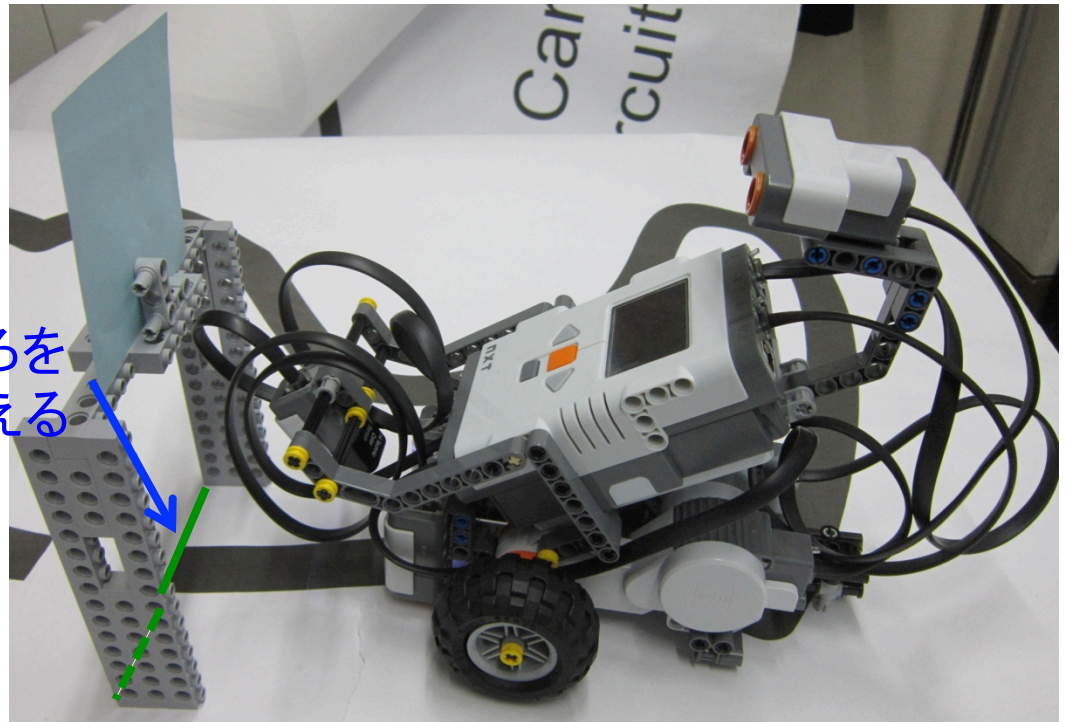
# ゴール

- Moodle上のgoal.lxfに従って組み立てる

付箋をはさむ



後ろを揃える





# 試行

---

- 最低3回、map1を完走する
  - レポート課題を受け付ける条件
- Moodleの「map1試行結果」に、各自が回ごとの完走時間を記録する
  - 途中リタイアの結果は記録してはいけない
  - Moodleでは、すべての操作がログに残る
  - 不正があった場合、厳しく罰する
- プログラムを修正した場合、累計3回記録すればよい

# map1試行結果への記録(1/3)

## 1. 「エントリを追加する」を押す

### map1試行結果

- 完走時間(秒)を入力してください
  - 各記録の右端にある「歯車」アイコンをクリックすると、入力・更新できます
  - ラインからのずれが大きすぎる時は、登録を控えてください
  - ほんの少しのずれならば、登録して構いません
- 最低3回、記録を入力してください
  - 最大5回まで記録できます
  - 6回以上試行した場合、気に入らない結果を上書きして構いません
- ルールを守って楽しくライントレース!!

一覧表示個別表示検索エントリを追加する

# map1試行結果への記録(2/3)

- 学籍番号と1回目の時間を入力し、  
「保存して表示する」を押す

### 新しいエントリ

学籍番号:

1回目:

2回目:

3回目:

4回目:

5回目:

# map1試行結果への記録(3/3)

3. 2回目以降を入力する場合、「一覧表示」を選ぶ
4. 歯車をクリックする
5. 時間を入力し、「保存して表示する」を押す



# シミュレーターとの違い

---

- 行動に時間がかかる
- スタートの位置にロボットを置く時に、毎回、微妙にずれる
- 紙にしわがある
- 段差がある
- それほど影響しないが、ゴールの位置が、毎回、微妙にずれる

# 頑健なロボットを作るために

## ● Model.javaでスタート位置と向きを定めている

```
/** マップの名前配列 */
private String[] mapNames = { "map1-rect.png", "map2-circ.png",
"map3-grid.png", "map4-grid.png", "map5-motegi.png", "map6-monte.png",
"map7-fuji.png", "map8-suzuka.png" };
/** マップごとのロボットの開始X座標 */
private double[] startX = { 330, 450, 88, 135, 410, 94, 410, 580 };
/** マップごとのロボットの開始Y座標 */
private double[] startY = { 130, 138, 450, 480, 435, 320, 205, 117 };
/** マップごとのロボットの開始時の向き */
private double[] startDir = { 90, 110, 0, 90, 95, 10, 90, 90 };
```

**startX[0]とstartY[0]の値を±10ほど、  
startDir[0]の値を±5ほど動かし、  
頑健かどうかを事前に確かめることができる**



# 第8回の出席確認

---

- "学籍番号\_R1.java"をMoodle上で提出する
  - 一番良いと思うプログラムを提出する
  - 最終評価には直接関係しません

# 片付け

---

- ロボットとUSBケーブルを青いコンテナに入れる
  - 不要な部品は、基本セットの箱の中に入れる
- ノートPCは棚に返却する
  - LANケーブルも返却する

---

● Special thanks:

● 山本 泰生先生

● 鍋島 英知先生