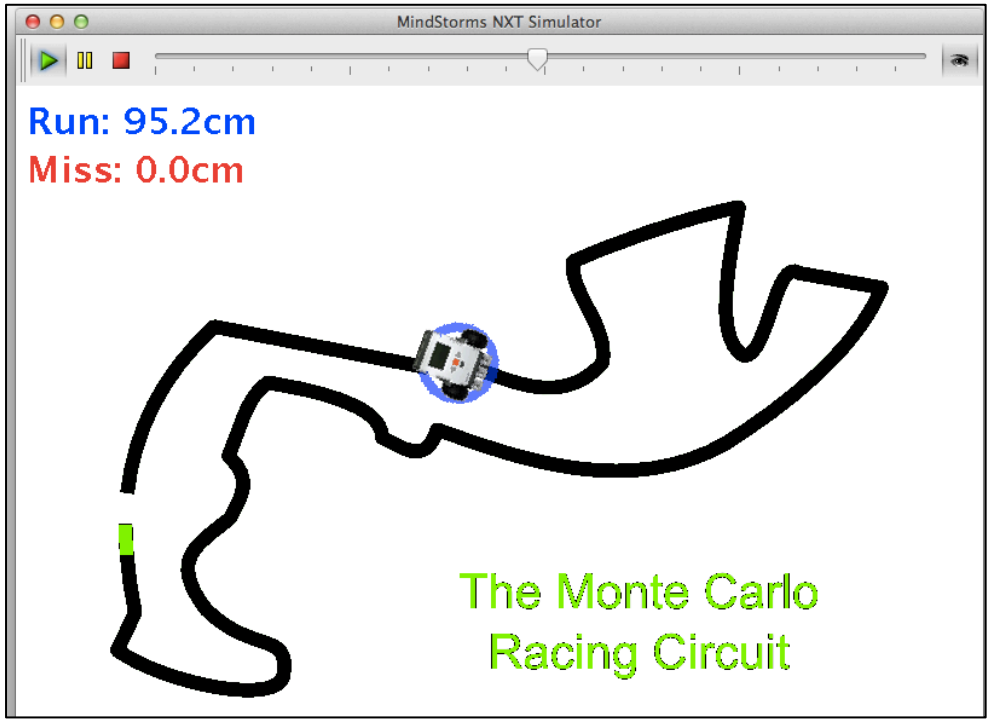
CS 知的システム演習

ライントレーサーのプログラム

松吉 俊

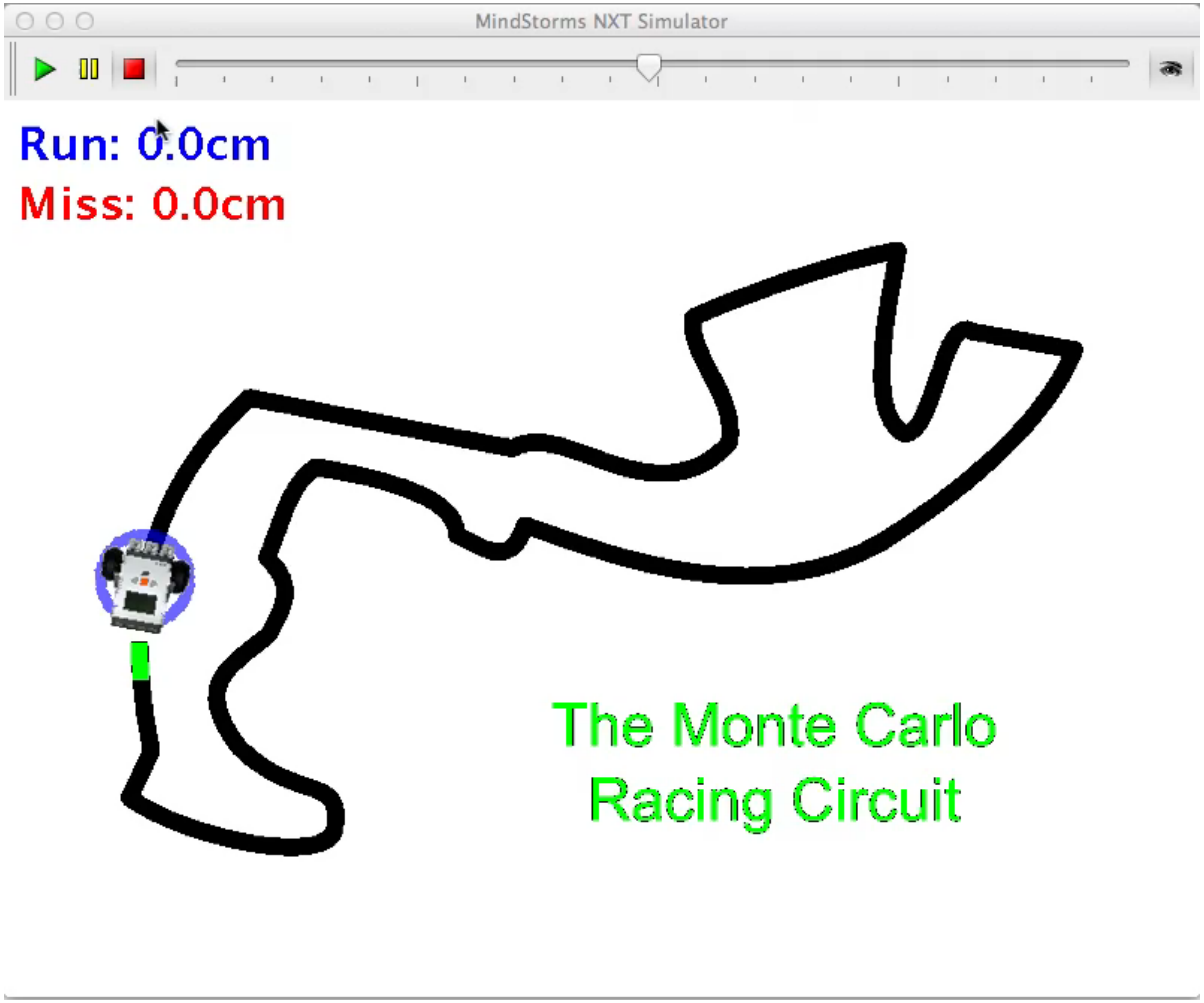
ライントレーサーのシミュレーター



- 黒: ライン
- 緑: ゴール
- センサー: 3つ
  - 光センサーA
  - 光センサーB
  - 光センサーC
- 行動:
  - 前進、後進
  - 左回転、右回転

状況を簡単にするため、
まずは色センサーの代わりに
光センサーを使ってシミュレートする

デモ



スタートからゴールまでの**Runの距離が短いほど良い**
(無駄が少ない)

ラインから外れると、
Missが加算される。
Missが少ないほど良い
(線を辿っているので)

雛形プログラム

ソースファイル

Simulator.java	起動用のクラス
Model.java	シミュレーションデータを管理するクラス
View.java	描画を担当するクラス
ControlToolBar.java	実行制御用のツールバー
Robot.java	抽象ロボットクラス
MyRobot.java	**サンプルロボットクラス**

基本的にはこのクラスのみ編集すれば十分

※ その他のクラスも自由に編集して良い

マップデータ

- map1-rect.png ～ map6-monte.pngまでの6種類

その他

- ロボットやアイコンなどの画像データ6つ

雛形プログラムのダウンロード

1. Moodleから、linetracer-template.tar.gz をダウンロードする
2. 適当なディレクトリに保存する
3. 圧縮ファイルを解凍する

   % tar xvzf linetracer-template.tar.gz

雛形プログラムのコンパイルと実行

- コンパイル

  % javac *.java ← すべての.javaファイルから、.classファイルが生成されます

- 実行

  % java Simulator **MyRobot** map1-rect.png

  起動用のメインclass

  ユーザが作成したロボットclassを指定する

  マップの名前

```
public class MyRobot extends Robot
{
  public void run() throws InterruptedException
  {
    while (true) {

      // 右センサの色に応じて分岐
      switch (getColor(LIGHT_A)) {

      case BLACK:                  // 黒を検知 ⇒ 右回転 ⇒ 前進
        rotateRight(10);
        forward(1);
        break;

      case WHITE:                  // 白を検知 ⇒ 左回転 ⇒ 前進
        rotateLeft(10);
        forward(1);
        break;
      }

      delay();                     // 速度調整＆画面描画

      if (isOnGoal()) return;      // ゴールに到達すれば終了

    }
  }
}
```

必ず Robot クラスを継承すること

ここをがんばって作る

センサーの位置

進む方向

センサーB: 光

センサーC: 光

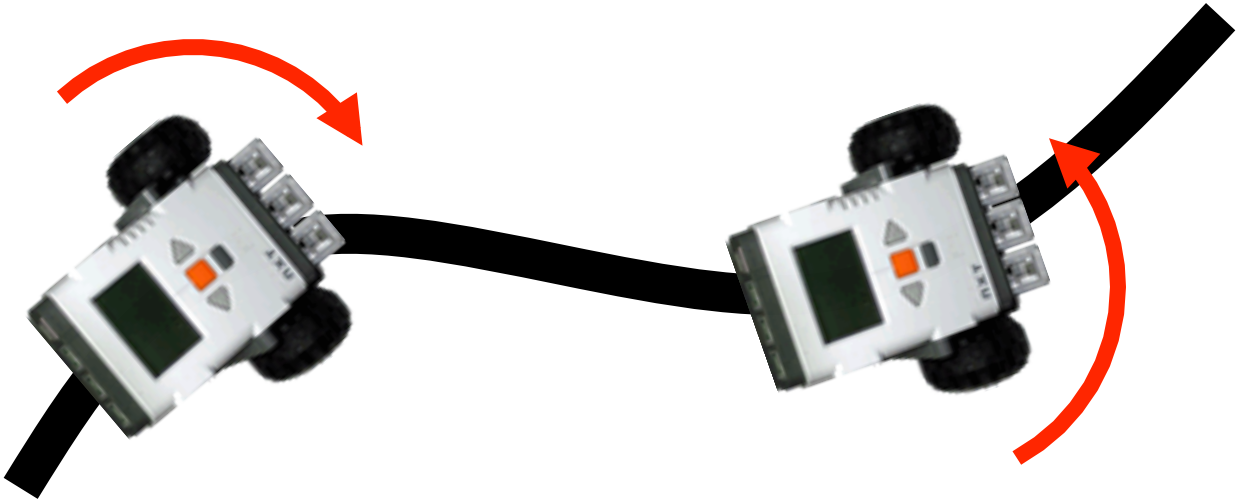
センサーA: 光

今回は、
3つのセンサーすべてが
緑を検知した時に
ロボットが自動停止する

超音波センサー

後方

雛形 MyRobot のアルゴリズム



	センサー	プログラム内では
A	光センサー右	BLACK
B	光センサー中	WHITE
C	光センサー左	WHITE

① センサーAが黒を検知したら、
右回転＆少し前進

	センサー	プログラム内では
A	光センサー右	WHITE
B	光センサー中	BLACK
C	光センサー左	WHITE

② センサーAが白を検知したら、
左回転＆少し前進

非常にシンプルだが、無駄な動き（回転）が多い

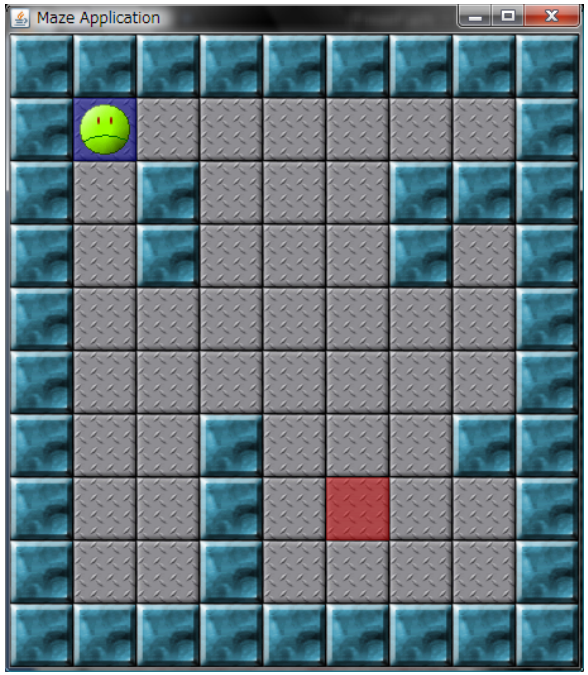
クラスRobotの概要

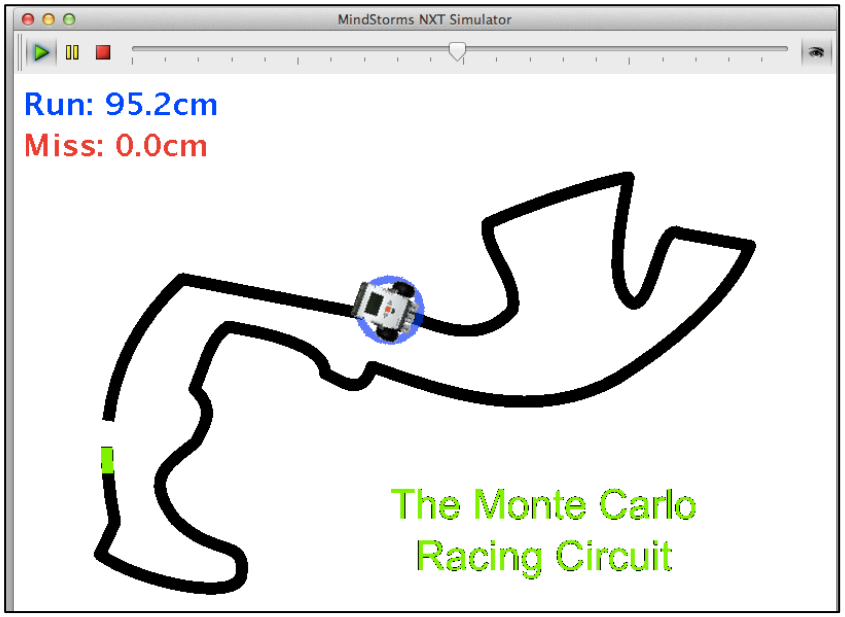
init()	ロボットを開始位置に戻す
delay()	速度調整＆描画用のメソッド
forward(double cm)	ロボットを指定距離だけ前進させる
backward(double cm)	ロボットを指定距離だけ後進させる
rotate(double angle)	ロボットを指定角度だけ回転させる （正の角度は右回転、負は左回転）
rotateRight(double angle)	ロボットを指定角度だけ右回転させる （正の角度を指定する）
rotateLeft(double angle)	ロボットを指定角度だけ左回転させる （正の角度を指定する）
getColor(int lightNo)	指定センサーから色を読み取る （センサーは LIGHT_A、LIGHT_B、LIGHT_C。 色は WHITEとBLACKの2色）
isOnGoal()	ゴールに到達すると true を返す

できることとできないこと

センサー入力: (1) ラインに乗っているかどうか
(2) ゴール位置にあるかどうか

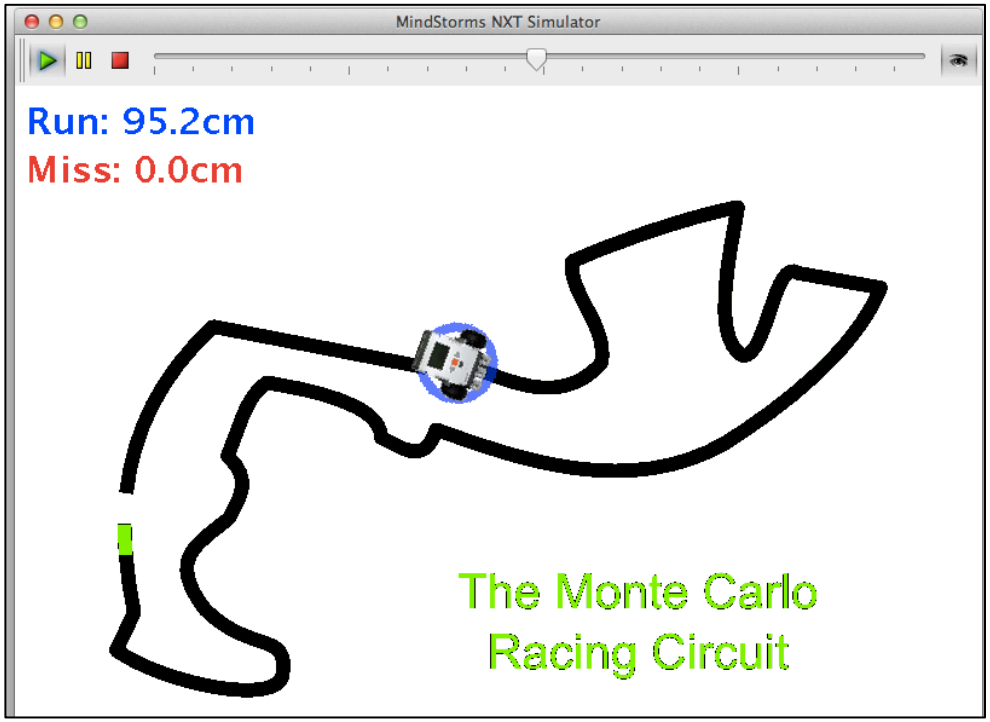
現在位置 (座標) の情報を取得できない





演習4

- 6つのmapにおいて、できるだけ短い距離でゴールするライントレーサーのアルゴリズムを考案し、実装せよ



第3回の出席確認

- 実装できたところまでで良いので、
  演習4に対するプログラムをMoodle上で
  提出する
  - tar.gz形式のファイルを提出する
  - 最終評定には直接関係しません

- Special thanks:
  - 山本 泰生先生
  - 鍋島 英知先生