DATA15001

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

THE FINAL EPISODE (11): ROBOTICS
TODAY’S MENU

1. "GRAND CHALLENGE"
2. LEGO MINDSTORMS
3. ROBO WORKSHOPS
ROBOTICS AS A "GRAND CHALLENGE OF AI"

actuators:
+ movement (precision, control)

sensors:
+ video & audio
+ touch (feedback loop)
+ balance
+ sense of movement
+ taste & smell

"AI proper":
+ computer vision
+ sound
+ (speech recognition)
+ speech
+ NLP
+ information retrieval
+ reasoning (logic, probability)
+ machine learning
+ search
+ games
+ affective computing
ROBOTS IN THE CULTURE

- A prominent topic in fiction
  - good robots:
    + Tik-Tok (Oz)
    + Wall-E
    + Data (Star Trek), ...
  - bad (or 'unhappy') robots:
    + 'Robot' (Karel Čapek, 1923),
    + Terminator (Part 1),
    + Roy (Blade Runner)

- Very far from reality
REALITY CHECK
REALITY CHECK

Endpoint Control

Introducing Handle
7,321,110 views
MINDSTORMS

- Touch sensor
- 3 motors
- Main "brick" (computer)
- Sound sensor
- Light sensor
- Ultrasonic sensor
MINDSTORMS: TRIBOT

- 2 motors connected to wheels
- 1 motor connected to "claws"
- ultrasound and touch: front
- light sensor: down
MINDSTORMS: "CAR"

Motor.A
light sensor
Motor.B

ultrasound
(this third wheel)

this way forward
leJOS

- leJOS NXT: An operating system that can be run on Mindstorms
- Has a Java Virtual Machine: you can control the robot by writing Java code
- Tools (on your/our computer):
  - Java compiler with package lejos
  - Uplink to robot (USB or Bluetooth)
- Well documented API for accessing motors and sensors
leJOS: PROGRAMMING

- Integrated development environment in Eclipse
  - writing code
  - compiling
  - uploading to robot

- We'll provide the environment so that you don't have to install it on your computer (but you can if you like)
Normal Java

```java
public class RobotTest {
    public static void main (String[] args) {
        System.out.println("Hei mualima.");
    }
}
```

After compilation, bytecode uploaded to robot

Choose program from robot menu and execute

You can abort the program on the robot by pressing the orange and dark gray buttons at the same time
MOTORS

• 3 motors, 3 motor ports

• Static objects

• API Class NXTRegulatedMotor

  Motor.A.setSpeed(400);  // set speed
  Motor.A.forward();      // start rotating
  Motor.A.stop();         // stop
  Motor.A.backward();     // start rotating back
  Motor.A.rotate(45);     // rotate 45 degrees
PILOT

- A higher-level interface to the motors in vehicle-type robots
- API class `DifferentialPilot`

Need to specify the diameter (wD) and axis span (aD) of wheels:

```java
DifferentialPilot pilot =
    new DifferentialPilot(wD, aD, Motor.A, Motor.B);

pilot.travel(50); // drive 50 cm ahead
pilot.rotate(-90); // turn 90° counter-clockwise.
```
TOUCH SENSOR

• Returns a Boolean indicating whether pressed or not

• API class TouchSensor

    TouchSensor touch =
    new TouchSensor(SensorPort.S1);

• touch.isPressed() // true if pressed
LIGHT SENSOR

- Returns an integer that says how bright the target is.

- API class `LightSensor`

```java
LightSensor light = new LightSensor(SensorPort.S2);
```

- `int l = light.readValue()` // 0-100

- Note: Lightness depends on the environment, so you can't know what, e.g., 48 means.

- The sensor can be calibrated (see API) but in practice, it's easiest to simply print out the numbers and adjust your code.
ULTRASONIC SENSOR

- Returns an integer that says far the target is (in cm)

- API class UltraSonicSensor

  ```
  UltraSonicSensor sonic = new UltraSonicSensor(SensorPort.S3);
  ```

- `int d = sonic.getDistance()`  // 0–255 (cm)

- Works best with flat, hard surfaces

- Accuracy a couple cm, max 200 cm

- Return value 255 means that no object was detected (no echo)
ABOUT ROBO WORKSHOPS

• You have to register on Doodle
  – not (necessarily) your regular exercise group!
  – see the link in the github material (Exercise 6.1)

• You should prepare a solution on your own before the workshop
  – otherwise you will probably not have enough time to complete both tasks
  – however, you will not be able to do any testing without a robot (and the leJOS environment): "blind coding"

• In the workshops, you'll do the tasks in pairs
  – tasks are quite straightforward
  – but hard because of the sensing constraints
  – have to think differently
ABOUT ROBO WORKSHOPS

• One robot + laptop with leJOS per pair

• Pre-built "cars"
  – check the motor and sensor ports (follow the cables)