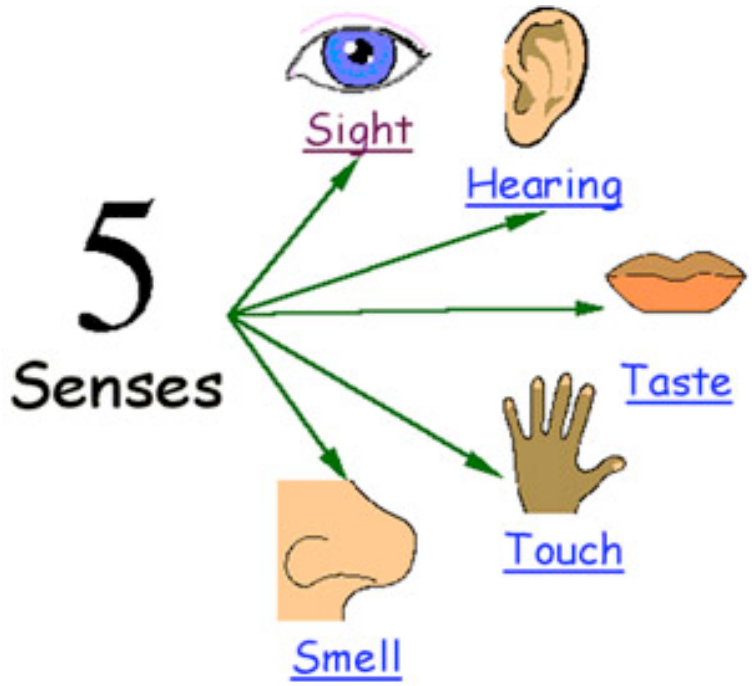
Sensing From Within

Week #4

Prof. Ryan Kastner

Five Senses for Humans?

- Our bodies have special sensory receptors



- *Example:* Taste receptors are concentrated mostly on tongue

Exteroceptors and Interoceptors

- The perceptual system of any organism includes a set of **external sensors (exteroceptors)** and **internal sensing mechanism** (interoceptors or proprioception)

- *Can you touch your belly button in the dark? How?*

Because of Proprioception

Senses for Robots?

- Essential for Robots!
- Robots have both **internal** and **external** sensors.
- *These sensor are capable of*
  - *Sensing light*
  - *Temperature*
  - *Touch*
  - *Distance to another object, etc…*

Proprioception in the Scribbler?

- There are **three** very useful internal mechanism!

1) Stall

Why: It could be stuck against a wall!!

2) Time

Why: Knowing the time is important to have more complex robot behaviors!!

3) Battery Level

Why: So you can detect when to change the batteries!!

Sensing Stall

- getStall()
  - Returns True if the robot has stalled
  - Returns False Otherwise

How would you use this commad

as a control behavior?

while not getStall():

<do something>

Keep doing

<do something> until

the robot has stalled

Sensing Stall: Example

- Write this program:

"Go forward unless you bump into something

```
while not getStall():
    forward(1.0)
stop()
speak ("Ouch! I think I bumped into something!")
```

Sensing Battery Power Levels

- Scribbler runs on 6AA batteries

- When the battery levels go down, you will get lower and lower voltages causing erratic behavior.

- getBattery()
  - Returns the current voltage being supplied by the battery
  - The battery voltage levels will vary between 0 and 9 volts (0 is being totally drained)

Sensing Battery Power Levels

- Scribbler runs on 6AA batteries

- When the battery levels go down, you will get lower and lower voltages causing erratic behavior.

- getBattery()
  - Returns the current voltage being supplied by the battery
  - The battery voltage levels will vary between 0 and 9 volts (0 is being totally drained)

Sensing Battery Power Levels: Example

- The red LED on the robot
  - remains lit when the power levels are high;
  - starts to flash when the battery levels are low

How would you use this commad

as a control behavior?

```
while (getBattery() >=5):
    <do something>
```

Time

- You used this sensor while writing *timeRemaining* and *wait* commands

- While writing commands like *forward(1, 2.5)*, this is where Scribbler gets time information

- Try

currentTime()

Time

- The value returned by *currentTime()* is a number that represents the seconds elapsed since some earlier time, whatever that is

- Issue the command several times and notice the difference

forward (1.0, 3.0)

forward (1.0, 3.0)
wait(3.0)

while timeRemaining (3.0)
forward(1.0)

Time

- Remember these commands?

forward (1.0, 3.0)

forward (1.0, 3.0)
wait(3.0)

while timeRemaining (3.0)
forward(1.0)

- Can you write the code to perform the same behavior using time sensor?

Remember These

```
# do something N times
for step in range(N):
    do something
```

```
# do something forever
while True:
    do something
```

```
# do something for some duration
while timeRemaining(duration):
    do something
```

```
# do something for some duration
duration = <some time in seconds>
startTime = currentTime()
while timeRemaining(duration):
    do something
```

Writing Conditionals

- Conditions result in either of two values:

  True (1) or **False** (0)

- Simple conditions can be written using comparison operations:

  < (less than), <= (less than or equal to)

  > (greater than), >= (greater than or equal to)

  == (equal to), != (not equal to)

Writing Conditionals

- Try these:

42 > 23

a, b, c = 10, 20, 10

a== c

a == a

True == 1

False == 1

- Try these too:

“Hello” == “Good Bye”

“Elmore” < “Elvis”

“New York” < “Paris”

“A” < “B”

“a” < “A”

Writing Conditionals

"Hello" == "Good Bye" → False

"Elmore" < "Elvis" → True

"New York" < "Paris" → True

"A" < "B" → True

"a" < "A" → False

- Strings are compared using alphabetical order;
- Uppercase letters are less than their equivalent lowercase counterparts

Writing Conditionals

- You can build more complex conditional expressions **using logical operations** (also called Boolean operations)
  - and
  - or
  - not
- *Try some examples*
  - *(5 > 7) and (8 > 3)*
  - *not ((5 > 7) and (8 > 3))*

Random Works

- One way to do interesting things with robots to inject some randomness in their behaviors
- Python provide a library for generating random numbers
- In order to access the random number library, you have the import random library:

*from random import**

Random Works

- There are many different features available in this library;

- Such as try

random()

Returns a random number between 0.0 and 1.0

random(A, B)

Returns a random number in the range [A…B]

Asking Questions

- Scribbler can also ask questions:

askQuestion ("Are you Ready?")

- You can have more control on these questions and answers:

askQuestion ("Change my pen to a different color and press 'OK' when ready")

- You can even specify lists:

askQuestion ("What is your favorite ice cream flavor?", ["Vanilla", "Chocolate", "Mango", "Hazelnut" , "Other"])

How to Import Your File?

- Save your file to the Network Drive. Then:

```
import sys
sys.path.append("X:/")
import your_file.py
```