Programando con Python y Robots

Fernando López

LINTI
Facultad de Informática
Universidad Nacional de la Plata

29 de Agosto de 2013


LINTI
Laboratorio de Investigación en
Nuevas Tecnologías Informáticas

Indice

Los inicios del proyectos

- Inicialmente en 2008 basado en los materiales del IPRE
- Compras en baja escala
- Cursos en escuelas mediante contacto con los directivos
- Cursos en pasantías de alumnos de los colegios de UNLP

Los inicios del proyectos

- Inicialmente en 2008 basado en los materiales del IPRE
- Compras en baja escala
- Cursos en escuelas mediante contacto con los directivos
- Cursos en pasantías de alumnos de los colegios de UNLP

Los inicios del proyectos

- Inicialmente en 2008 basado en los materiales del IPRE
- Compras en baja escala
- Cursos en escuelas mediante contacto con los directivos
- Cursos en pasantías de alumnos de los colegios de UNLP

Los inicios del proyectos

- Inicialmente en 2008 basado en los materiales del IPRE
- Compras en baja escala
- Cursos en escuelas mediante contacto con los directivos
- Cursos en pasantías de alumnos de los colegios de UNLP

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Hardware

- Robot Scribbler 1 de Parallax
  - Sensores de línea IR
  - Sensores de obstáculos IR
  - Parlante
- Placa de expansión del IPRE:
  - Bluetooth
  - Cámara de 640x480 píxels
  - Sensores IR
  - Sensores de luminosidad

Problemas con el hardware

- Importados desde EEUU
- Problemas con la conexión Bluetooth
- Licencia de la API poco clara (una parte es shared source pero no libre)

- Alternativa desarrollada en Argentina por Robot Group
- Garantía cuando un robot se daña
- API de bajo nivel con licencia tipo MIT

Problemas con el hardware

- Importados desde EEUU
- Problemas con la conexión Bluetooth
- Licencia de la API poco clara (una parte es shared source pero no libre)

- Alternativa desarrollada en Argentina por Robot Group
- Garantía cuando un robot se daña
- API de bajo nivel con licencia tipo MIT

Problemas con el hardware

- Importados desde EEUU
- Problemas con la conexión Bluetooth
- Licencia de la API poco clara (una parte es shared source pero no libre)

- Alternativa desarrollada en Argentina por Robot Group
- Garantía cuando un robot se daña
- API de bajo nivel con licencia tipo MIT

Problemas con el hardware

- Importados desde EEUU
- Problemas con la conexión Bluetooth
- Licencia de la API poco clara (una parte es shared source pero no libre)

- Alternativa desarrollada en Argentina por Robot Group
- Garantía cuando un robot se daña
- API de bajo nivel con licencia tipo MIT

Problemas con el hardware

- Importados desde EEUU
- Problemas con la conexión Bluetooth
- Licencia de la API poco clara (una parte es shared source pero no libre)

- Alternativa desarrollada en Argentina por Robot Group
- Garantía cuando un robot se daña
- API de bajo nivel con licencia tipo MIT

Problemas con el hardware

- Importados desde EEUU
- Problemas con la conexión Bluetooth
- Licencia de la API poco clara (una parte es shared source pero no libre)

- Alternativa desarrollada en Argentina por Robot Group
- Garantía cuando un robot se daña
- API de bajo nivel con licencia tipo MIT

El proyecto en las escuelas 2012

- Proyecto financiado por Fundación YPF
- 20 robots por escuela
- 1 notebook con Ubuntu
- 1 cañón
- Alumnos y docentes de la UNLP hicieron de tutores

El proyecto en las escuelas 2012

- Proyecto financiado por Fundación YPF
- 20 robots por escuela
- 1 notebook con Ubuntu
- 1 cañón
- Alumnos y docentes de la UNLP hicieron de tutores

El proyecto en las escuelas 2012

- Proyecto financiado por Fundación YPF
- 20 robots por escuela
- 1 notebook con Ubuntu
- 1 cañón
- Alumnos y docentes de la UNLP hicieron de tutores

El proyecto en las escuelas 2012

- Proyecto financiado por Fundación YPF
- 20 robots por escuela
- 1 notebook con Ubuntu
- 1 cañón
- Alumnos y docentes de la UNLP hicieron de tutores

El proyecto en las escuelas 2012

- Proyecto financiado por Fundación YPF
- 20 robots por escuela
- 1 notebook con Ubuntu
- 1 cañón
- Alumnos y docentes de la UNLP hicieron de tutores

Este año

- Nuevas pasantías
- Apoyo a las escuelas que siguen trabajando por su cuenta
- Búsqueda de alternativas económicas a ZigBee

Este año

- Nuevas pasantías
- Apoyo a las escuelas que siguen trabajando por su cuenta
- Búsqueda de alternativas económicas a ZigBee

Este año

- Nuevas pasantías
- Apoyo a las escuelas que siguen trabajando por su cuenta
- Búsqueda de alternativas económicas a ZigBee

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Software necesario

Para trabajar va a ser necesario:

- Alguna distribución GNU/Linux (de preferencia Lihuen o Debian)
- Python 2.6/2.7
- Algún intérprete como pycrust o idle (opcional)
- Algún editor de texto con resaltado de sintaxis (como Geany/Gedit)
- La API para controlar el robot:
  - En Lihuen: `apt-get install robot`
  - En cualquier distro[1]: `pip install duinobot`
  - Para desarrollo:

```
git clone git@github.com:Robots-Linti/duinobot.git
cd duinobot
python setup.py install
```

[1] Primero instalar pygame, pyserial y tkinter

Revisión rápida de la práctica 1

- Instalación: Ejercicios 1 a 9
- Realizar: Ejercicios 10 y 11

Revisión rápida de la práctica 1

- Instalación: Ejercicios 1 a 9
- Realizar: Ejercicios 10 y 11

Primeros pasos

- Descargar el manual:

  `http://robots.linti.unlp.edu.ar/material_disponible`

- Abrir PyCrust y conectarnos al robot:

```
from duinobot import *
board = Board()
robot = Robot(board, 0)
```

- Por defecto los robots vienen con id 0, hay que inicializarlos de a uno:

```
robot.setId(4)
robot = Robot(board, 4)
```

Primeros pasos

- Descargar el manual:

  `http://robots.linti.unlp.edu.ar/material_disponible`

- Abrir PyCrust y conectarnos al robot:

```
from duinobot import *
board = Board()
robot = Robot(board, 0)
```

- Por defecto los robots vienen con id 0, hay que inicializarlos de a uno:

```
robot.setId(4)
robot = Robot(board, 4)
```

Primeros pasos

- Descargar el manual:

  `http://robots.linti.unlp.edu.ar/material_disponible`

- Abrir PyCrust y conectarnos al robot:

```
from duinobot import *
board = Board()
robot = Robot(board, 0)
```

- Por defecto los robots vienen con id 0, hay que inicializarlos de a uno:

```
robot.setId(4)
robot = Robot(board, 4)
```

Primeros pasos

- Descargar el manual:

  `http://robots.linti.unlp.edu.ar/material_disponible`

- Abrir PyCrust y conectarnos al robot:

```
from duinobot import *
board = Board()
robot = Robot(board, 0)
```

- Por defecto los robots vienen con id 0, hay que inicializarlos de a uno:

```
robot.setId(4)
robot = Robot(board, 4)
```

Primeros pasos

- Descargar el manual:

  `http://robots.linti.unlp.edu.ar/material_disponible`

- Abrir PyCrust y conectarnos al robot:

```
from duinobot import *
board = Board()
robot = Robot(board, 0)
```

- Por defecto los robots vienen con id 0, hay que inicializarlos de a uno:

```
robot.setId(4)
robot = Robot(board, 4)
```

Movimientos básicos y beep

- `robot.forward(velocidad, tiempo)`
- `robot.backward(velocidad, tiempo)`
- `robot.turnLeft(velocidad, tiempo)`
- `robot.turnRight(velocidad, tiempo)`
- `robot.beep(frecuencia, tiempo)`

Con tiempo en segundos...

- Mover el robot
- Hacer un triángulo
- Hacer un cuadrado
- Hacer 5 cuadrados

Movimientos básicos y beep

- `robot.forward(velocidad, tiempo)`
- `robot.backward(velocidad, tiempo)`
- `robot.turnLeft(velocidad, tiempo)`
- `robot.turnRight(velocidad, tiempo)`
- `robot.beep(frecuencia, tiempo)`

Con tiempo en segundos...

- Mover el robot
- Hacer un triángulo
- Hacer un cuadrado
- Hacer 5 cuadrados

Movimientos básicos y beep

- `robot.forward(velocidad, tiempo)`
- `robot.backward(velocidad, tiempo)`
- `robot.turnLeft(velocidad, tiempo)`
- `robot.turnRight(velocidad, tiempo)`
- `robot.beep(frecuencia, tiempo)`

Con tiempo en segundos...

- Mover el robot
- Hacer un triángulo
- Hacer un cuadrado
- Hacer 5 cuadrados

Movimientos básicos y beep

- `robot.forward(velocidad, tiempo)`
- `robot.backward(velocidad, tiempo)`
- `robot.turnLeft(velocidad, tiempo)`
- `robot.turnRight(velocidad, tiempo)`
- `robot.beep(frecuencia, tiempo)`

Con tiempo en segundos...

- Mover el robot
- Hacer un triángulo
- Hacer un cuadrado
- Hacer 5 cuadrados

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón “start” del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón “start” del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón “start” del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón “start” del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón “start” del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón "start" del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón "start" del joystick

Con Joystick

- Principalmente usado para motivar
- Acciones predefinidas

```
joy = Joystick(robot, nroJoystick)
joy.play()
```

- Video TEC La Plata
- El número de Joystick va de 0 en adelante
- La API aún no tiene una forma directa de programar los botones
- Los acelerómetros de las Notebooks a veces son listados por el PyGame como Joysticks
- joy.play() es un loop que termina con el botón “start” del joystick

Métodos y funciones útiles

- board.report() ← lista de robots encendidos
- joysticks() ← lista de Joysticks disponibles
- robot.senses() ← panel con los valores de los sensores [2]

Ejemplo: Robot sorpresa

```
from random import random
from duinobot import *
board = Board()
robots = board.report()
eleccion = int(random() * len(robots))
robot = Robot(board, robots[eleccion])
robot.forward(50, 2)
board.exit()
```

[2] es un poco propenso a generar errores aún, en la versión 0.12 debería ser más seguro

Métodos y funciones útiles

- board.report() ← lista de robots encendidos
- joysticks() ← lista de Joysticks disponibles
- robot.senses() ← panel con los valores de los sensores [2]

Ejemplo: Robot sorpresa

```
from random import random
from duinobot import *
board = Board()
robots = board.report()
eleccion = int(random() * len(robots))
robot = Robot(board, robots[eleccion])
robot.forward(50, 2)
board.exit()
```

[2] es un poco propenso a generar errores aún, en la versión 0.12 debería ser más seguro

¿Cómo harían para que el robot vaya haciendo beeps a mientras avanza?

¿Sin detenerse en cada beep?

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
    robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
    robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
  robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
  robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
  robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
  robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Beeps mientras el robot se mueve

- Imposible con robot.forward(velocidad, tiempo)
- Decimos que el anterior es bloqueante ya que bloquea el interprete hasta terminar
- robot.forward(velocidad) es la versión no bloqueante

```
robot.forward(50)
for i in range(5):
  robot.beep(i * 100, 1)

robot.stop()
```

- Cuando usamos la anterior hay que recordar usar robot.stop() al final
- Todos los métodos de movimiento se pueden invocar de forma bloqueante y no bloqueante
- También el beep() (se detiene con beep(0))

Estructuras de control y raw_input()

- Se pueden plantear juegos de preguntas y respuestas
- Juevos de azar con random
- Control remoto con las teclas (con ENTER)
- Secuencia de control en un string

Estructuras de control y raw_input()

- Se pueden plantear juegos de preguntas y respuestas
- Juevos de azar con random
- Control remoto con las teclas (con ENTER)
- Secuencia de control en un string

Estructuras de control y raw_input()

- Se pueden plantear juegos de preguntas y respuestas
- Juevos de azar con random
- Control remoto con las teclas (con ENTER)
- Secuencia de control en un string

Estructuras de control y raw_input()

- Se pueden plantear juegos de preguntas y respuestas
- Juevos de azar con random
- Control remoto con las teclas (con ENTER)
- Secuencia de control en un string

Ejercicio

Escribir un script que decodifique un string donde cada letra representa un movimiento:

- f $\rightarrow$ avanzar 1 segundo
- b $\rightarrow$ retroceder 1 segundo
- l $\rightarrow$ girar a izquierda medio segundo
- r $\rightarrow$ girar a derecha medio segundo

Por ejemplo: "flfrb" avanza, gira a izquierda, avanza, gira a derecha y retrocede.

Ejercicio

Escribir un script que decodifique un string donde cada letra representa un movimiento:

- f $\rightarrow$ avanzar 1 segundo
- b $\rightarrow$ retroceder 1 segundo
- l $\rightarrow$ girar a izquierda medio segundo
- r $\rightarrow$ girar a derecha medio segundo

Por ejemplo: "flfrb" avanza, gira a izquierda, avanza, gira a derecha y retrocede.

¡Un slide en blanco!

Una posible solución


```
movimientos = "flfbrfbl"
for movimiento in movimientos:
    if movimiento == "f":
        robot.forward(50, 1)
    elif movimiento == "b":
        robot.backward(50, 1)
    elif movimiento == "l":
        robot.turnLeft(50, 1)
    elif movimiento == "r":
        robot.turnRight(50, 1)
    else:
        print "Movimiento no válido"
        robot.beep(500, 1)
```

Revisión rápida de la práctica 2

Vistazo de los ejercicios

Sensores y condicionales de nuevo

- `robot.getWheels()`
- `robot.getLine()`
- `robot.ping()`
- `robot.getObstacle(distance)`
- `robot.getIR()` → no implementado
- `robot.senses()` → en proceso de ser reimplementado usando un event loop

Sensores y condicionales de nuevo

- `robot.getWheels()`
- `robot.getLine()`
- `robot.ping()`
- `robot.getObstacle(distance)`
- `robot.getIR()` → no implementado
- `robot.senses()` → en proceso de ser reimplementado usando un event loop

Sensores y condicionales de nuevo

- `robot.getWheels()`
- `robot.getLine()`
- `robot.ping()`
- `robot.getObstacle(distance)`
- `robot.getIR()` → no implementado
- `robot.senses()` → en proceso de ser reimplementado usando un event loop

Sensores y condicionales de nuevo

- `robot.getWheels()`
- `robot.getLine()`
- `robot.ping()`
- `robot.getObstacle(distance)`
- `robot.getIR()` → no implementado
- `robot.senses()` → en proceso de ser reimplementado usando un event loop

Sensores y condicionales de nuevo

- `robot.getWheels()`
- `robot.getLine()`
- `robot.ping()`
- `robot.getObstacle(distance)`
- `robot.getIR()` $\rightarrow$ no implementado
- `robot.senses()` $\rightarrow$ en proceso de ser reimplementado usando un event loop

Sensores y condicionales de nuevo

- `robot.getWheels()`
- `robot.getLine()`
- `robot.ping()`
- `robot.getObstacle(distance)`
- `robot.getIR()` $\rightarrow$ no implementado
- `robot.senses()` $\rightarrow$ en proceso de ser reimplementado usando un event loop

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Actividades

- Leer los sensores desde el intérprete interativo
- Hacer un script que haga que el robot avance en superficies claras y se detenga en superficies oscuras
- Hacer un script que haga que el robot se detenga al encontrar un obstáculo
- Hacer que el robot esquive el obstáculo en lugar de detenerse
- Hacer que el robot se escape marcha atrás cuando detecte que algo se acerca
- Hacer distintos tipos de alarmas con los sensores usando el beep:
  - Al ser levantado del suelo
  - Al pasar sobre un objeto más oscuro o claro que el suelo
  - Al detectar un obstáculo

Direcciones

Direcciones de contacto

robots@linti.unlp.edu.ar
soportelihuen@linti.unlp.edu.ar