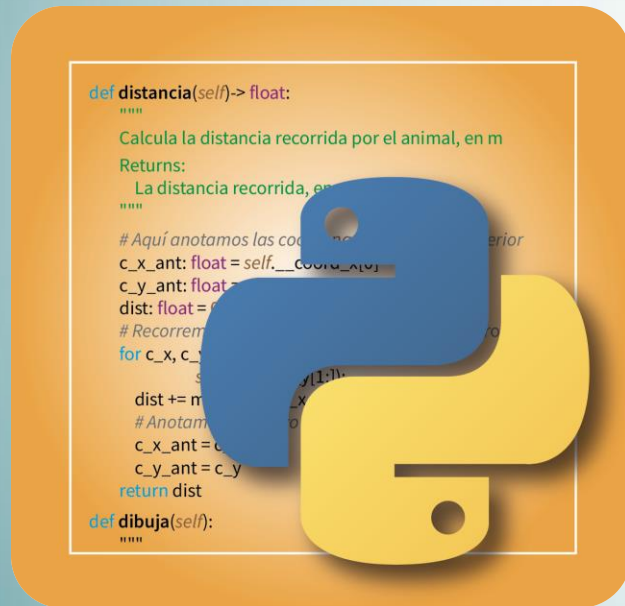


Práctica 7. Movimiento de un robot bidimensional



Michael González Harbour
José Javier Gutiérrez García
José Carlos Palencia Gutiérrez
José Ignacio Espeso Martínez
Adolfo Garandal Martín

Departamento de Ingeniería
Informática y Electrónica

Este material se publica con licencia:
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 7: Movimiento de un robot bidimensional

Objetivos. Practicar con los bucles

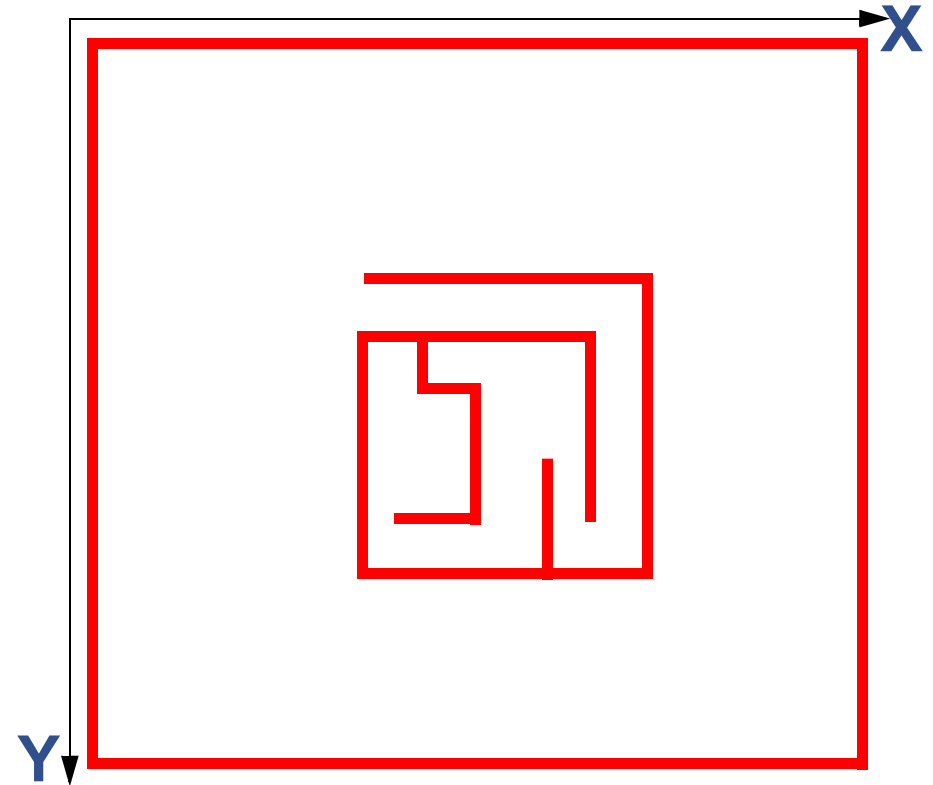
Simularemos el movimiento de un robot bidimensional en un recinto cerrado con obstáculos

El recinto tiene los ejes X e Y indicados en el dibujo y se divide en 120x120 celdas

El movimiento se produce según los ángulos 0, 90, 180, o 270 grados

- 0° =derecha, 90° =arriba, etc.

El avance es siempre de una celda cada vez



Clases disponibles

Se dispone de las clases siguientes *ya realizadas*, basadas en el paquete fundamentos v1.1 (*Nota*: descargar esta nueva versión)

- `recinto.Recinto`: representa el recinto con obstáculos
 - esta clase no debe usarse directamente
- `robot.Robot`: representa el robot que se mueve, con estos métodos:
 - constructor, permite situar el robot en una posición y dirección inicial
 - avanzar el robot una celda (si no hay obstáculo)
 - girar 90 grados
 - saber si hay un obstáculo delante, detrás, a la izquierda, o a la derecha
 - pintar el robot y el recinto
 - saber si el robot está dentro del laberinto
 - destruir la ventana con el dibujo del laberinto y el robot

Consultar la documentación de la clase `Robot` para más detalles

Realización

Observaciones:

- Pintar el robot cada vez que se avance o se gire

Escribir un módulo `mueve_robot.py` que tenga estas funciones:

- `avanza_recto()`:
 - crea el robot en la posición (10, 10) y ángulo 0°
 - avanza en línea recta hasta que encuentra un obstáculo delante
 - destruye el robot
- `avanza_en_escalera()`: usando bucles para no repetir instrucciones:
 - crea el robot en la posición (70, 100) y ángulo 0°
 - repite 8 veces:
 - avanza 5 pasos y gira 90°
 - avanza 5 pasos y gira 270°
 - destruye el robot

Realización (cont.)

- `avanza_evitando()`:
 - crea el robot en la posición (35, 65) y ángulo 0°
 - hace 40 veces lo siguiente:
 - avanza en línea recta hasta encontrar un obstáculo
 - gira 90 grados
 - destruye el robot
- `avanza_y_gira_aleatorio()`:
 - crea el robot en la posición (35, 65) y ángulo 0°
 - hace 20 veces lo siguiente:
 - avanza en línea recta hasta encontrar un obstáculo
 - gira un ángulo aleatorio de 90, 180, o 270 grados
 - destruye el robot

Nota: Para obtener un número entero aleatorio entre `start` y `stop` (incluidos) se puede usar `random.randint(start, stop)`

Parte avanzada: estrategias para salir del laberinto

Añadir al módulo `mueve_robot.py` estas nuevas funciones, que deberán retornar el número de pasos que ha dado el robot para salir del laberinto para evaluar el algoritmo:

- `movimiento_aleatorio()`:
 - crea el robot en la posición (65, 45) y ángulo 0°
 - mientras el robot no esté fuera del laberinto:
 - gira un ángulo aleatorio de 0, 90, 180, o 270 grados
 - avanza un número de pasos aleatorio entre 0 y 10
 - destruye el robot

Parte avanzada (cont.)

- `avanza_con_mano_derecha()`:
 - crea el robot en la posición (65, 45) y ángulo 0°
 - Avanza en línea recta hasta encontrar un obstáculo delante y gira 90°
 - Mientras el robot no esté fuera del laberinto simula el avance con la mano derecha pegada a la pared, repitiendo estos pasos:
 - sabemos si la mano derecha está pegada a la pared mirando a ver si hay un obstáculo a la derecha; en ese caso, si no hay obstáculo delante, avanza, y si no, hemos llegado a una esquina izquierda y hay que girar 90°
 - si la mano no está pegada a la derecha entonces hemos llegado a una esquina derecha: gira 270 grados y avanza
 - destruye el robot

Entregar 2 ficheros

- El código del módulo `mueve_robot.py`
- Un informe en pdf con:
 - *parte básica*: una captura de pantalla del resultado de una cualquiera de las funciones
 - *parte avanzada*: una breve conclusión sobre cuántos pasos dan las dos funciones realizadas para salir del laberinto