

Taller # 5 (Robomind)

INSTRUCCIONES BÁSICAS

adelante(1)

atras(1)

Estas dos primeras tienen un número entre paréntesis. El número indica el número de cuadros por el escenario que se moverá el robot.

izquierda()

derecha()

Pues bien, fácil el roboMind girará ala izquierda o la derecha.

pintarBlanco()

pintarNegro()

detenerPintar()

Saca la brocha y pinta de negro o de blanco. Si quieres que recoja la brocha y deje de pintar ya sabes detenerPintar()

Ejercicio 1

Abre el mapa ApenArea.Map y cree un programa que escriba la letra “A”

```
pintarBlanco()  
adelante(2)  
derecha()  
adelante(1)  
derecha()  
adelante(2)  
atras(1)  
derecha()  
adelante(1)  
detenerPintar()
```



Ejercicio 2

Abre el mapa



Haz un programa que pinte de color negro el borde los dos cuadros con dos piscinas que están enfrente del robot al inicio y que al acabar de pintar el borde vuelva a la posición inicial, pero sin pintar.

Ejercicio 3

Muy bien ahora vamos a usar los sensores de Robomind. Para eso necesitamos la condicional si (if en ingles). Lo que hará será comprobar que cumple la condición que le digamos y si la cumple realizará las ordenes que le demos y si no la cumple pues nada sigue su camino sin hacer las órdenes dadas. La estructura es esta:

```
si (condición)
{
    Lo que hace si se cumple la condición
}
```

Abrir el mapa default.map y haz que el robot se mueva hasta la baliza y al detectarla con la condicional si la recoja. Después tiene que llevar y dejar la baliza en el punto de inicio. Aquí tienes las instrucciones

```
derecha()  
adelante(3)  
derecha()  
adelante(1)  
si(frenteEsBaliza() )  
{  
    tomar()  
}  
derecha()  
adelante(3)  
derecha()  
poner()  
fin
```



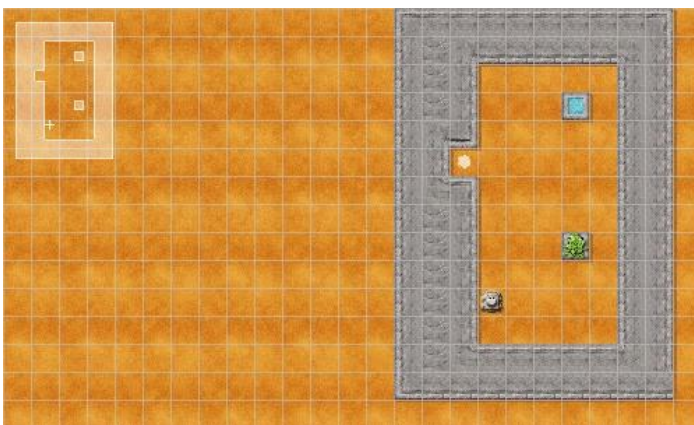
Observa que ponemos fin cuando queremos que se para y finaliza el programa.

Ahora te toca a ti solito.

Ejercicio 4

En el mapa default.map, crea un programa que recoja la baliza más lejana del robot y la deje al lado de la otra baliza.

Abre el mapa findSpot1.map.



Ejercicio 5

Cree un Programa para que el robot se ubique en el círculo blanco. Tiene que detectar el punto blanco el sensor de mirar a la izquierda.

Otra función muy útil en Robomind es la de repetir. Podemos decirle que repita algo un número de veces determinado o que lo repita siempre.

```
repetir(3)
```

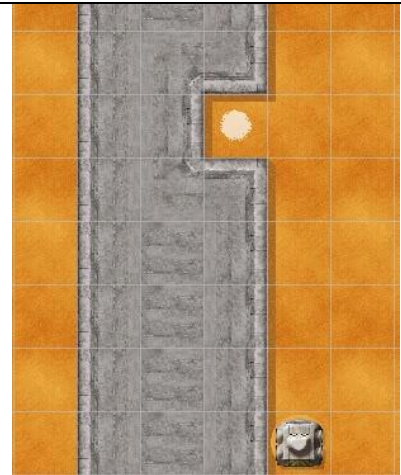
```
{  
    Repite 3 veces lo que hay aquí  
}
```

```
repetir()
```

```
{  
    Repite de forma indefinida lo que hay aquí  
}
```

Cree un Programa para que el robot se ubique en el círculo blanco Utilizando un ciclo.

```
repetir()  
{  
    si(izquierdaEsBlanco())  
    {  
        # There's a white spot on your  
        # izquierda  
        izquierda()  
        adelante(1)  
        fin  
    }  
    Otro  
    {  
        # There's no white spot yet  
        adelante(1)  
    }  
}
```

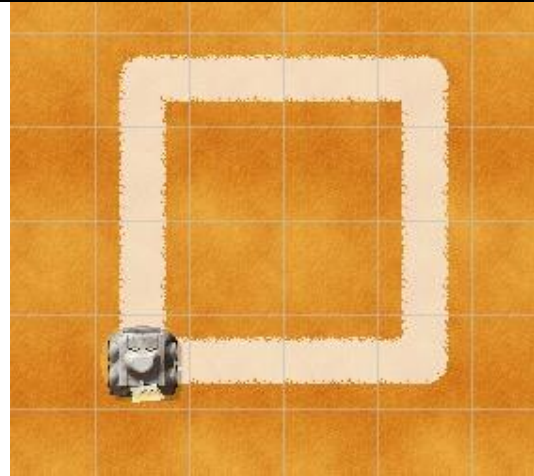


Ejercicio 6

Fíjate en este ejemplo vamos hacer que pinte un cuadrado de forma muy sencilla y con pocas instrucciones:

Abre el mapa OpenArea.map y crea un programa que haga un cuadro de lado 4 repitiendo 3 instrucciones:

```
repetir(4)
{
  pintarBlanco()
  adelante(3)
  derecha()
}
```



Ejercicio 7

Elaborar un programa que escriba en el mapa la palabra “ROBOT.”

Ejercicio 8

Abra el mapa default.map y cree un programa donde el robot se mueva sin parar por el mapa de forma que cuando se encuentre un obstáculo gire a la derecha para poder avanzar:

```
repetir()
{
  adelante(1)
  si(frenteEsObstaculo() )
  {
    derecha()
  }
}
```

Ejercicio 9

Ahora vamos hacer que el robot sea capaz de seguir una línea blanca pintada en un escenario:

Abre el mapa default.map y que el coche llegue a la línea blanca y la siga



Forma 1	Forma 2
<pre>derecha() adelante(8) adelante(2) izquierda() adelante(2) derecha() adelante(2) izquierda() adelante(6) izquierda() adelante(8) derecha() adelante(2)</pre>	<pre># Line follower # This script make the robot # automatically trace the white line. # map: default.map # Go to the start derecha() adelante(8) # look in which direction is # white paint and take a step repetir() { si(frenteEsBlanco()) { adelante(1) } Otro si(derechaEsBlanco()) { derecha()</pre>

	<pre>} otro si(izquierdaEsBlanco()) { izquierda() } otro si(frenteEsObstaculo()) { Fin } }</pre>
--	--

¿Te das cuenta?

Robomind en su avance analiza si hay una línea blanca a su izquierda a su derecha o de frente y actúa en consecuencia.

Ejercicio 10

¿Sabrías hacer el mismo programa pero que pinte encima de la línea blanca una negra?

Ejercicio 11

Abre el mapa Mapa GoRightAtWhite1.map

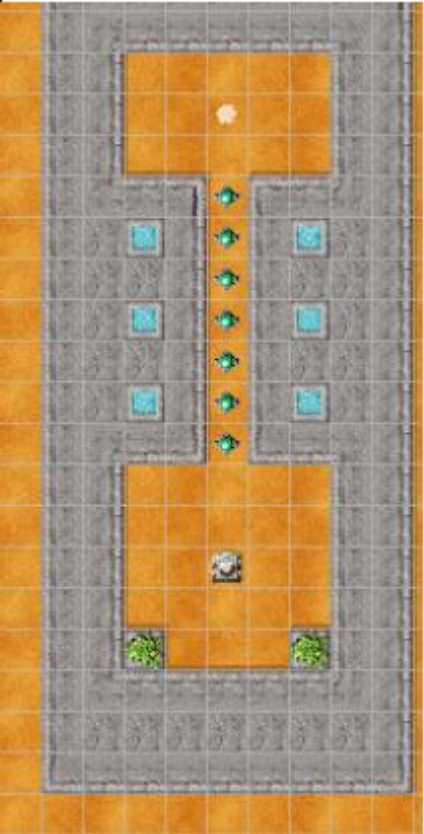


Escribe un programa para que el robot vaya del primer punto blanco al segundo y después vaya a recoger la baliza.

Ejercicio 12

Vamos hacer que el robot haga cosas independientemente.

Abre el mapa passBeacons1.map y cree un programa para que el robot sea capaz de llegar al punto blanco y pararse encima del.

	<pre>repetir() { adelante(1) si(frenteEsBaliza()) { tomar() derecha() derecha() adelante(3) poner() derecha() derecha() } si(frenteEsBlanco()) { adelante(1) fin } }</pre>
--	--

Ejercicio 13

¡¡¡Muy Bien!!! Ahora vamos a ver otra instrucción que vale para que Robomind tome decisiones aleatorias. Nosotros no sabemos la decisión que va a tomar, por eso es aleatoria, esta instrucción es lanzar moneda. Es como si lanzase una moneda y unas veces saldrá cara o cruz, para el programa será verdadero o falso. Para hacer esto la sentencia es la siguiente:


```

si ( lanzarMoneda())
{
    hace esto
}
otro
{
    Hace esto otro
}

```

Hará lo primero si es verdadero (moneda cara) y hará lo segundo si sale cruz o falso para el programa.

Veamos un ejemplo:

Abre el mapa Huguemap.map y crea un programa para que el robot avance por el mapa independientemente de forma que cuando se encuentre un obstáculo vaya a la derecha o a la izquierda (aleatoriamente).

```

repetir()
{
    si(frenteEsObstaculo() )
    {
        si( lanzarMoneda())
        {
            derecha()
        }
        Otro
        {
            izquierda()
        }
    }
    Otro
    {
        adelante(1)
    }
}

```



Bueno ahora te toca a ti los dos últimos ejercicios:

Ejercicio 14

Con cualquier mapa, ahora haz que con el programa anterior el robot marque con un punto negro los puntos donde haya un obstáculo.

Ejercicio 15

Abre el maze1.map o el maze2.map y crea un programa que sea capaz de recorrer el laberinto y recoger