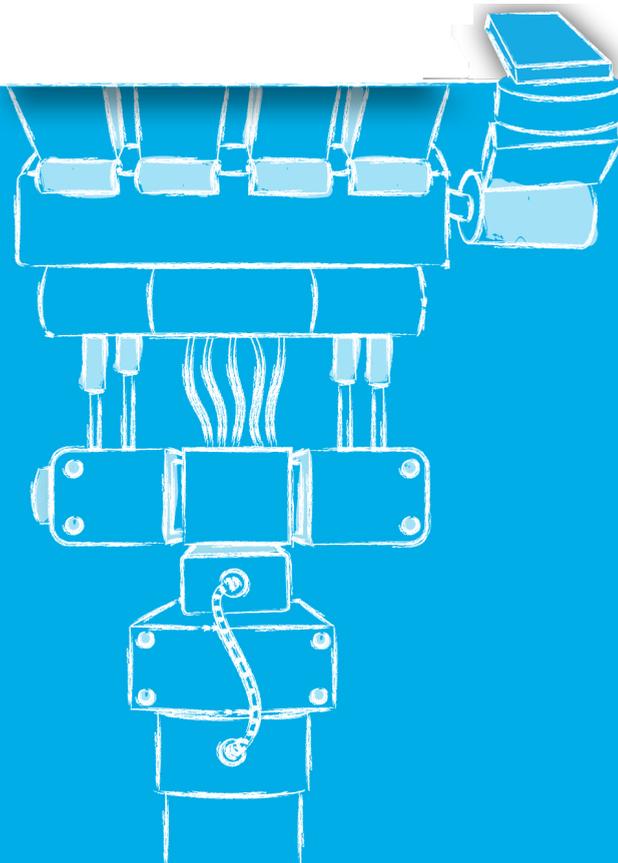


**ANEXO 5. Proyecto: Curvas en el
parque (movimientos circulares)**



ANEXO 5.

Proyecto: Curvas en el parque (movimientos circulares)

1. GIRAR CON PEQUEÑOS DESPLAZAMIENTOS Y PEQUEÑOS GIROS

El giro “puro” de un robot (virtual y real) se consigue haciendo pivotar el robot en torno a un punto, el centro de giro del robot¹. Ello tiene la gran ventaja de separar el movimiento lineal del movimiento angular del objeto móvil, y así hemos girado en el capítulo 7, apartado 6.

Pero, en la realidad, los coches no giran pivotando: giran simultaneando pequeños desplazamientos con pequeños giros. De ello resultan las trayectorias curvas de los coches cuando giran en una curva de la carretera...

Vamos a plantearnos aprender a escribir tareas que hagan girar al robot de esta manera: a base de pequeños desplazamientos y pequeños giros.

Una forma simple de describir una trayectoria curva en BYOB es repetir múltiples veces (por ejemplo 100) un pequeño movimiento rectilíneo de avance del robot, por ejemplo, 1 paso, seguido de un pequeño giro “puro”, por ejemplo de 1 grado.

Colocando el robot horizontalmente en el punto inicial $x = 0$; $y = 60$, y programando la tarea anterior, el resultado en BYOB es el siguiente:²

¹ El punto “rojo” marcado en nuestros coches.

² Es importante respetar la selección “-R” de las instrucciones “mover” y “girar” en estos programas de repetición, ello hace que los desplazamientos se realicen correctamente. El uso de la instrucción “mover-L” puede dar resultados erróneos con parámetros de desplazamiento inferiores a la unidad.

```

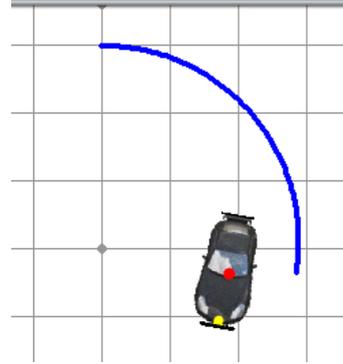
Ir a posición x= 0 y= 60
orientarse en dirección 90 grados
fijar tamaño de lápiz a 4
bajar lápiz

```

```

repetir 100
  mover-R 1 pasos
  girar a la derecha-R 1 grados

```



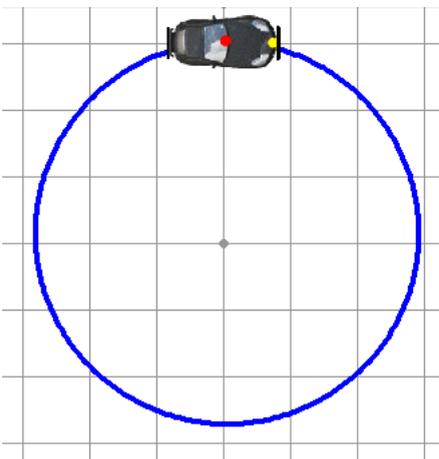
Observamos que, efectivamente, la trayectoria del robot es una curva (hemos desplazado la posición final del robot para poder ver el final de la trayectoria). Como el programa es una repetición de iguales ángulos y desplazamientos, podemos prever que la trayectoria es próxima a una circunferencia (sería una circunferencia perfecta en la situación límite de infinitos giros con desplazamientos infinitesimales...)

Podemos verificar lo anterior programando una repetición de 360 veces para las dos instrucciones anteriores, de modo que, en total, el robot gire 360 grados.

```

repetir 360
  mover-R 1 pasos
  girar a la derecha-R 1 grados

```



Obtenemos, así, una circunferencia con gran precisión (aunque en realidad sea un polígono de 360 lados, que no podemos distinguir de una circunferencia debido a la insuficiente densidad de píxeles en la pantalla).

2. EL RADIO DE LA CIRCUNFERENCIA

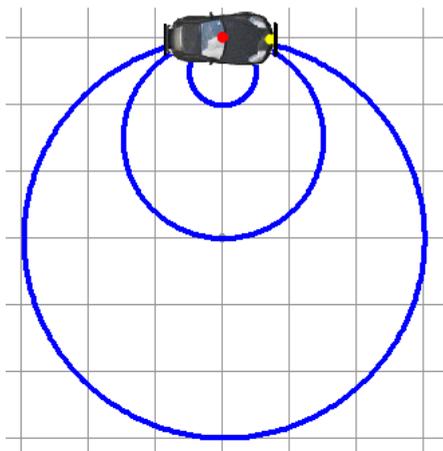
El programa que hemos escrito supone una definición “constructiva” de una circunferencia, que no utiliza la noción de radio³. Por ello no sabemos a priori el valor del radio de la circunferencia que ha descrito el robot. Pero podemos calcularlo por un método exploratorio, como a continuación se describe.

Haciendo medidas sobre la trayectoria del coche, vemos que con 360 desplazamientos (avances) de 1 paso, el diámetro es aproximadamente de 115 pasos y, en consecuencia, el radio es de aproximadamente 57,5 pasos.

Si hacemos la hipótesis de que *el radio de la circunferencia y, por tanto, su tamaño, es proporcional al tamaño del desplazamiento*, podemos establecer la siguiente proporción: si un desplazamiento unitario produce una circunferencia de radio = 57,5 pasos, un desplazamiento de $1/57,5$ producirá una circunferencia de radio = 1 paso.

Así: *Una circunferencia cualquiera de radio = R se obtiene con un desplazamiento de valor $R/57,5$.*

Podemos verificar la ley anterior construyendo circunferencias de radios 10, 30, 60 pasos (véase figura).



³ Otras definiciones sí lo usan, por ejemplo la conocida descripción de la circunferencia como “El lugar geométrico de todos los puntos que equidistan de uno dado, que es su centro”. Aquí, la equidistancia es precisamente el valor del radio.

Con esta ley ya podemos predecir a dónde irá el robot cuando gire ejecutando trayectorias circulares:



- El número N de repeticiones indicará el sector de circunferencia girado: $N = 360$ corresponde a una circunferencia completa, $N = 180$ a media circunferencia, $N = 90$ a un cuarto, etc.
- El valor D del desplazamiento en cada instrucción “mover” predecirá el radio R de la trayectoria, según la función anterior: $D = R/57,5$
- El valor G del giro en cada instrucción “girar a la derecha” siempre será una constante, de valor 1 grado.

Recordemos, finalmente, que la constante hallada (el valor 57,5) depende de la escala del escenario, en nuestro caso de que cada cuadrado es de 20 x 20 pasos. Con otra escala hallaremos un valor diferente.

NOTA: Por cuestiones de programación, estos movimientos pequeños, generalmente inferiores a un paso, se deben programar con los procedimientos “mover-R” y “retroceder-R” (los procedimientos “mover-L” y retroceder-L” no funcionan).

3. UN PROCEDIMIENTO PRIMITIVO PARA DIBUJAR ARCOS DE CIRCUNFERENCIA



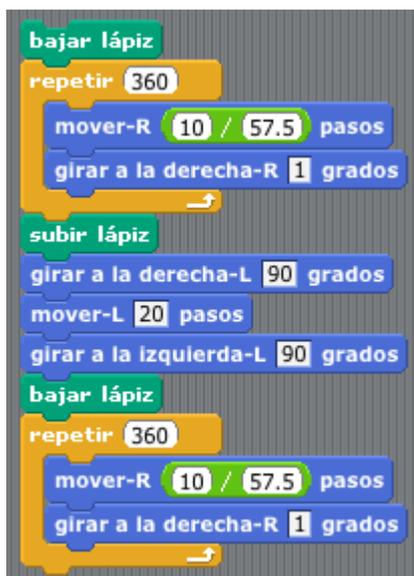
La ley anterior nos permite fácilmente escribir un procedimiento (un bloque de usuario) para dibujar un arco de circunferencia de ALFA grados (eventualmente una circunferencia entera) con un radio R determinado.

La figura muestra el procedimiento para un “arco a derecha”. Con la instrucción “girar a la izquierda-R” obtendríamos de un modo semejante el procedimiento “arco a izquierda...”.

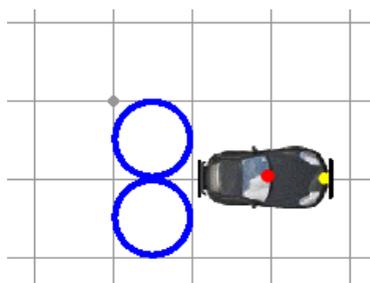
Es obvio que en los cursos de nivel elemental los alumnos deben utilizar estos procedimientos como primitivos para dibujar circunferencias y arcos de circunferencia cuando el radio de la circunferencia sea un dato numérico del problema. Ello no impide que inicialmente hayan aprendido a construir circunferencias por el método “continuo” de “avanzar un paso y girar un grado...” y hayan explorado este programa modificando sus parámetros para tener una noción cualitativa del diseño de curvas.

Ejemplos

Ejemplo 1. Dibujar “un ocho” recorriendo dos circunferencias inscritas en cuadrados contiguos de 20 x 20 pasos (dos cuadrados de la rejilla)



Los radios deben ser de 10 pasos y, en consecuencia, el desplazamiento debe ser de $0,0166 \times 10 = 0,166$



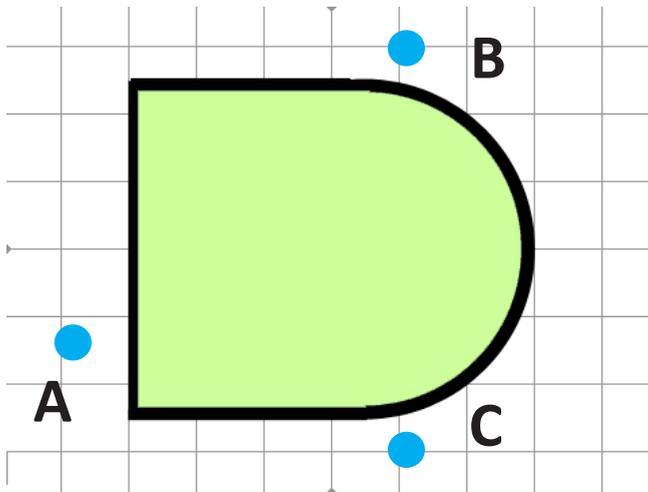
Al finalizar la ejecución del programa hemos desplazado el coche a la derecha para poder ver dibujo completo del recorrido.

Usando la instrucción “arco a derecha...” como primitiva, podemos igualmente realizar la misma tarea:

```

bajar lápiz
arco a derecha de 360 grados con radio de 10 pasos
subir lápiz
girar a la derecha-L 90 grados
mover-L 20 pasos
girar a la izquierda-L 90 grados
bajar lápiz
arco a derecha de 360 grados con radio de 10 pasos
  
```

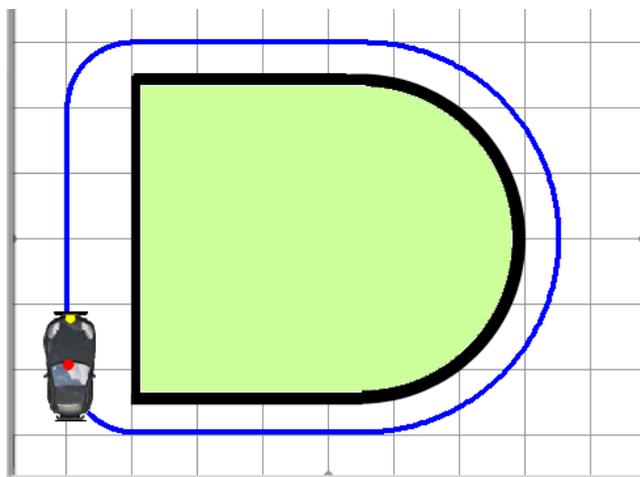
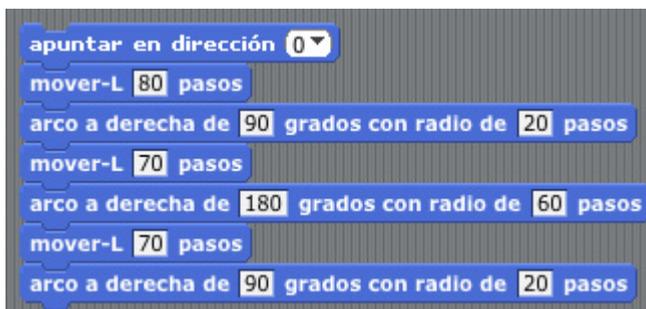
Ejemplo 2. Dar una vuelta al circuito de la plaza con el coche



El recorrido sale del punto A. El recorrido entre B y C es una semicircunferencia de un radio de 60 pasos...

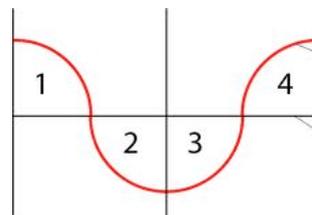
Según la técnica anterior, el desplazamiento del bucle de repetición debe ser de 1 paso (o algo mayor si se desea buena precisión).

La programación en BYOB y el resultado serán los de las figuras.



Ejemplo 3. Dibujar una ondulación como la de la figura, recorriendo semicircunferencias sucesivas

Recordemos que nuestro robot “productor” de circunferencias que hemos programado realiza trayectorias circulares “a derechas”. Es inmediato programar un productor de circunferencias “a izquierdas”...). Si el robot recorre la trayectoria de izquierda a derecha, el cuadrante 1 es “a derechas”, la semicircunferencia 2/3 es “a izquierdas” y el cuadrante 4 es “a derechas” de nuevo.

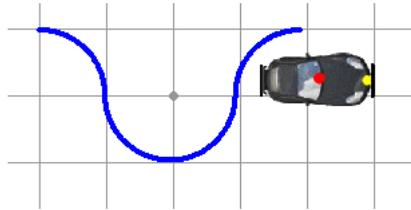


Haremos un recorrido que ocupe un rectángulo de 80 x 40 pasos (4 x 2 cuadrados), usando un desplazamiento a coordenadas cartesianas para situar el robot en la posición inicial.

```

subir lápiz
ir a posición x= -40 y= 20
orientarse en dirección 90 grados
bajar lápiz
arco a derecha de 90 grados con radio de 20 pasos
arco a izquierda de 180 grados con radio de 20 pasos
arco a derecha de 90 grados con radio de 20 pasos

```



Ejemplo 4. Aparcamientos

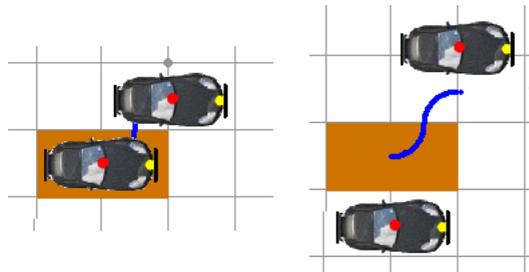
Nuestros procedimientos “arco a derecha... y a izquierda...” son siempre arcos “hacia delante”. Para aparcar debemos programar recorridos circulares de retroceso. Para ello volveremos a la programación inicial que explicita el avance/retroceso y el giro.

Ejemplo de un aparcamiento a la derecha y hacia atrás:

```

orientarse en dirección 90 grados
bajar lápiz
repetir 90
  retroceder-R 10 / 57.5 pasos
  girar a la izquierda-L 1 grados
  repetir 90
    retroceder-R 10 / 57.5 pasos
    girar a la derecha-L 1 grados

```



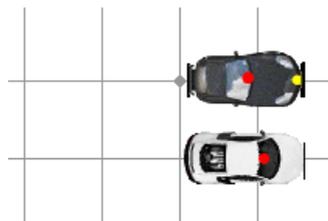
En el recorrido de aparcamiento anterior, el coche llega a girar 90 grados (poniéndose vertical respecto a la acera), lo que en la práctica es excesivo...

Se puede programar un aparcamiento más “suave” de modo que el coche gire menos grados en su retroceso. Por ejemplo, que gire 60 grados... Además iniciar y acabar el aparcamiento con un pequeño movimiento rectilíneo...

Todo ello se tiene en cuenta en el siguiente programa:

```
orientarse en direccion 90 grados
bajar lápiz
retroceder-L 10 pasos
repetir 60
  retroceder-R 20 / 57.5 pasos
  girar a la izquierda-L 1 grados
  →
repetir 60
  retroceder-R 20 / 57.5 pasos
  girar a la derecha-L 1 grados
  →
mover-L 10 pasos
```

Inicio:



Final:

