

Adaptación de la actividad descrita en: <https://todoscratch.com/tutorial-de-programacion-laberinto/>

Vamos a crear un juego en el que tendremos que escapar de un laberinto. Sigue los siguientes pasos:

1 Creación del fondo

Debemos definir la forma del nuestro laberinto. Podemos crearlo utilizando programas como Paint o GIMP, o bien podéis utilizar Google y buscar plantillas de laberintos.

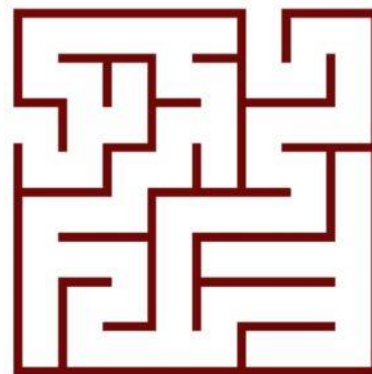
En el ejemplo utilizamos una plantilla encontrada con Google, modificada para que se vea mejor. El color de las paredes influirá en nuestro código, así que interesa que las paredes estén bien definidas y que tengan un color uniforme:

Este es el laberinto que tendremos que resolver:

Puedes descargar esta imagen desde el sitio web del centro o diseñar un laberinto diferente por tu cuenta.

Comenzaremos la programación de nuestro juego cargando esta imagen como escenario de nuestro juego. Puedes eliminar el escenario en blanco original.

Elimina también el personaje del gato pues no lo vamos a utilizar.



2 Creación y configuración del personaje

Añade al programa el objeto “Ball”. Reduce su tamaño de tal forma que se pueda mover sin problemas por los pasillos del laberinto. Una buena referencia sería un tamaño de 40. Selecciona el disfraz verde (se ve mejor sobre el fondo blanco).

Ahora ya estamos listos para la parte más importante, programar el código.

3 El punto de partida

Cada vez que iniciemos la partida (al darle a la bandera verde), lo primero que debe pasar es que el personaje se coloque en la casilla de salida. Utilizamos el bloque de movimiento «Ir a x... y...». Las coordenadas iniciales deben ser (-160,55).

4 El movimiento

Utilizaremos las flechas del teclado para mover a nuestro personaje. Para ello utilizaremos el condicional “Si... entonces”. Cada vez que el programa detecte que una de las flechas de posición está presionada, la posición del personaje debe cambiar 5 puntos en la dirección definida dicha (por ejemplo: 1 clic en la tecla ← debe disminuir en 5 unidades el valor de la coordenada X).

Este proceso no debe de funcionar hasta que iniciemos el juego y una vez que el juego se ponga en marcha debe repetirse por siempre hasta que termine la partida.

Ahora bien, ¿qué pasa si me choco contra una pared?

5 Detección de choques contra la pared

Tal como está ahora el código, atravesamos las paredes, no es la idea del juego. Hay que detectarlas. Para eso utilizaremos su color.

Utilizaremos otro condicional: Si la pelota está tocando el color de la pared, entonces vuelve hacia atrás. Este bloque lo añadiremos dentro de los bloques de movimiento creados en el punto anterior.

Para escoger el color correcto pinchamos sobre la opción de “copiar color”.

6 Finalización del programa

Ya tenemos nuestro laberinto operativo y podemos jugarlo. Pero vamos a pulir algún detalle.

6.1 Mensaje al inicio:

Cuando le damos a la bandera verde e iniciamos nuestro juego, vamos a hacer que salga un mensaje de bienvenida. El mensaje se debe mostrar durante tres segundos en la pantalla.

6.2 Reiniciar el programa:

Si nos hemos perdido por el laberinto y queremos volver a la posición de salida, haremos que pulsando la tecla «R» nos devuelva a la posición inicial.

6.3 Fin de la partida:

Vamos a hacer que cuando nuestro personaje llegue al final del laberinto y consiga salir, que el programa nos felicite y la partida termine.

Para ello añadimos un objeto tipo botón, y lo colocamos al final del laberinto. De manera que nos tape la salida:



Ahora le añadimos a la pelota el código para que cuando toque el nuevo objeto, nos entregue un mensaje de felicitación y se acabe el juego.

6.4 Tiempo empleado

Añade una variable que nos indique el número de segundos que hemos tardado en resolver el laberinto.