

Sesión

4

Aprendizajes esperados

Duración sugerida

Al final de esta sesión se espera que puedas:



Utilizar los disfraces de los objetos como estados.



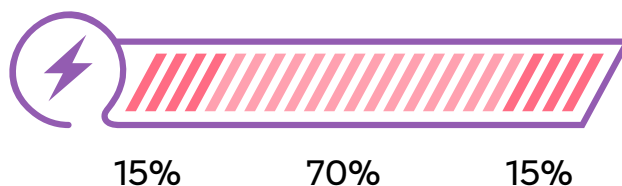
Asociar acciones a cada estado.



Cambiar el estado de los objetos al entrar en contacto con otro objeto.



Usar funciones con parámetros.



Material para la clase

- Anexo 4.1.
- Computador con acceso a internet.



Lo que sabemos, lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

En esta sesión vas a programar una versión simplificada del juego de la lleva congelada que es el reto central de esta guía.

Los requerimientos de este programa son:

- 1 Cada objeto solo puede tener un color a la vez.
- 2 5 objetos comienzan del mismo color y 1 de un color diferente (la lleva).
- 3 Los objetos se mueven aleatoriamente y rebotan en los bordes.
- 4 La lleva va más rápido que los demás.
- 5 Cuando la lleva toca a otro objeto, este se congela y cambia de color.

Después de leer los requerimientos, ¿conoces otros nombres para esta dinámica de juego?

Glosario



Clonar: crear una copia idéntica de un objeto, archivo, repositorio o estructura de datos.

Manos a la obra

Conectadas



Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos de 2 a 3 personas, siguiendo las recomendaciones de tu docente.

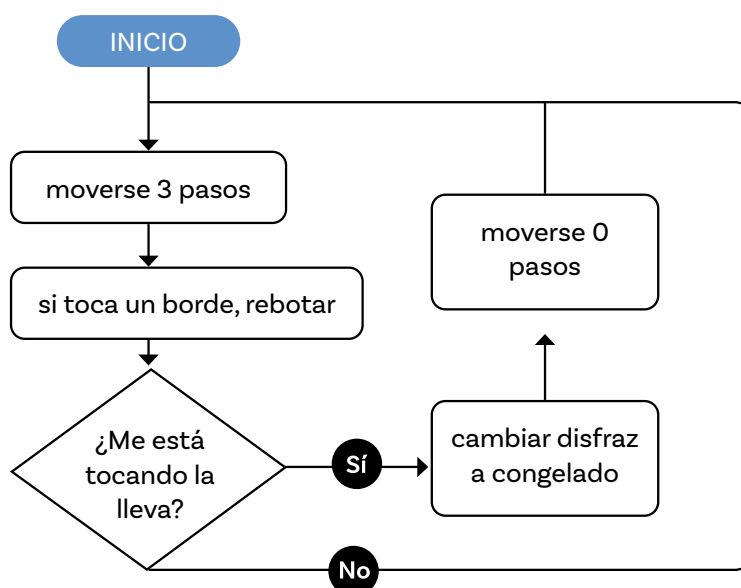
En la sesión anterior realizaron un programa de simulación que incluía requerimientos muy similares a los requerimientos 1, 2 y 3 del juego de lleva que van a programar. Así que ya saben cómo desarrollar estos tres puntos.

Ahora verán cómo responder al quinto requisito. Deben conservar la misma estructura, en la que hay un objeto que se **clona** y dos funciones. Una que indica la posición inicial y otra que determina el movimiento y cambio de disfraz.

En la *Figura 1* verán el diagrama de flujo de una primera aproximación a la función que controla el movimiento de los objetos para esta actividad.

Analicen este diagrama de flujo.

Figura 1. Diagrama de flujo: controlar el movimiento de los objetos



Recuerden que esta función le debe servir a todos los objetos, incluyendo al que tiene el rol de ser la lleva.



¿Le faltará algo a este algoritmo?

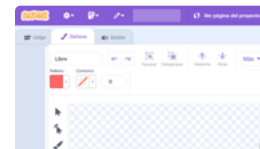
Tengan en cuenta cuándo se termina el algoritmo.

Nota

En esta actividad los objetos se comportan diferente dependiendo del rol que tengan. Algunos roles son para toda la actividad, como el de la lleva, que no cambia, y otros van cambiando pues pueden estar libres o congelados. En programación estos roles se suelen llamar estados.

En *Scratch*, una forma de implementar estados es con los disfraces.

Estos tienen 3 características que se pueden asociar a cada estado: un número, un nombre y colores. El nombre lo escribes:



Al cambiar el orden de los disfraces cambia el número y los colores los cambias en el lienzo. Para obtener algo así:



Anexo

Anexo 4.1

Nombre: _____

Con respecto al algoritmo inicial, en dicho, lee el condicional que está en blanco al principio, responde:

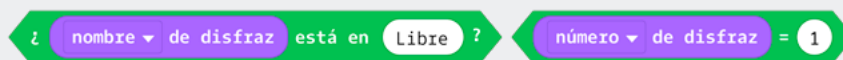


Sigan las instrucciones del Anexo 4.1 para encontrar lo que falta.

Ahora que ya conocen el algoritmo. Vean qué bloques en Scratch pueden ser útiles para implementarlo.

- 1 Lean la nota para entender el concepto de estado y ver cómo se implementa en Scratch.
- 2 Para identificar el estado de un objeto en su propio código se puede utilizar el número o el nombre. Siguiendo el ejemplo de la nota, para determinar si el estado del objeto es Libre se pueden utilizar cualquiera de las expresiones equivalentes que aparecen en la Figura 2.

Figura 2. Bloques de operadores en Scratch para identificar el estado de un objeto según su disfraz



Observen que no se puede usar la igualdad para comparar dos textos, sino que se debe preguntar si un texto está incluido en otro texto.

- 3 En Scratch solo hay una forma para determinar el estado de un bloque al que se está tocando: el color. Así que se debe utilizar el bloque sensor que aparece en la Figura 3.

Figura 3. Bloque de la categoría sensores ¿tocando el color__?



Con lo anterior ya pueden responder el quinto requisito.

- 4 Ahora aprenderán cómo responder al cuarto requisito. Recuerden que una única función determina el movimiento de todos los objetos incluido la lleva. Es precisamente esa la función para la cual programaron el algoritmo anterior.



¿Qué parte de ese algoritmo determina la velocidad a la que se mueven los objetos?

¿Para modificar la velocidad necesitarían cambiar bloques completos o solo un valor dentro de estos?

Sería deseable tener la misma función, pero usando un valor diferente en el código en su interior cada una de las dos veces que se usa la función. Es decir que, cuando se llame la función para la lleva, se le pasa un valor que haga que se mueva más rápido y, cuando se llame para los otros personajes, se le pasa un valor diferente que haga que se muevan más despacio.

La función que usaron en la sesión anterior no permite ninguna entrada, pero esto es posible. En el momento de crear la función se le pueden agregar entradas, permitiendo que la función sea más versátil. Las entradas también se conocen como **parámetros**.

Al momento de crear el bloque de la función, hagan clic en **Añadir una entrada**. Al hacerlo les aparecerá un recuadro dentro del bloque. En este deben escribir el nombre que adquirirá el parámetro dentro de la función, como se muestra en la *Figura 4*.

Figura 4. Creación de la función bucle movimiento, incluyendo el parámetro rapidez



Figura 5. Función bucle movimiento, desde la cual se hace llamado al parámetro rapidez

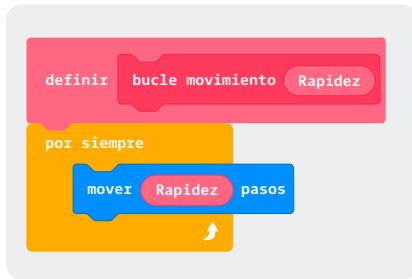


Figura 6. Bloque de llamado a la función bucle movimiento y parámetro rapidez (3)



Una función con parámetros de todas formas correrá el mismo código, con la diferencia de que, en alguna parte, el código usará una variable que tomará el valor de la entrada. En este ejemplo, cada vez que se quiera usar la entrada dentro del bloque se usará el parámetro llamado rapidez. Para usarlo, arrástrenlo desde la primera línea, como se muestra en la *Figura 5*.

Por lo anterior, cuando se usa la función, no basta con llamarla, sino que además se le debe incluir el valor de la entrada, ver *Figura 6*.

Ya tienen todos los elementos para responder a las especificaciones del simulador de juego de lleva que les han pedido realizar en esta sesión.

Glosario



Parámetro: variable especial que se usa dentro de una función para recibir información del exterior. De esta forma al llamar una función se le pueden pasar variables diferentes en cada uso de la función.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1

¿Puedes utilizar los disfraces de los objetos como estados?



Sí



Parcialmente



Aún no

2 ¿Puedes asociar acciones a cada estado?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

3 ¿Puedes cambiar el estado de los objetos al entrar en contacto con otro objeto?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

4 ¿Puedes usar funciones con parámetros?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas en Scratch. Luego, revisa con tus compañeras y compañeros de grupo lo que hicieron al crear la simulación del juego de lleva, en su versión simplificada. Si todavía te quedan dudas, consúltale a tu docente.

Trabaja ahora con tu grupo. Creen, al menos, 5 preguntas de repaso con al menos 3 opciones de respuesta, pero solo una respuesta correcta. Estas preguntas deben destacar lo que han aprendido o recordado con las actividades desarrolladas a lo largo de esta guía.

Marquen la respuesta correcta de cada una de sus preguntas. Cuando las tengan listas, entrégueñaselas a su docente. Algunas de sus propuestas podrán ser posteriormente utilizadas por su docente para hacer un juego de revisión de lo aprendido con toda la clase.