



Creando un juego con Alice 3

¿Qué es Java?

Java es un lenguaje de programación y una plataforma desarrollada por Sun Microsystems (ahora Oracle) en 1995. Java se ejecuta en más de 850 millones de computadoras personales de todo el mundo y en miles de millones de dispositivos de todo el mundo, incluidos los dispositivos móviles y de televisión.

¿Qué se puede hacer con Java?

Java te permite:

- Reproducir juegos en línea
- Chatear con personas de todo el mundo
- Usar mapas interactivos
- Ver imágenes en 3D

¿Dónde puedo encontrar Java?

Desde portátiles hasta centros de datos, consolas de juegos hasta supercomputadoras científicas, teléfonos móviles a Internet, ¡Java está en todas partes!



¿Qué es Alice?

Alice es un entorno innovador de programación basado en bloques que facilita la creación de animaciones, la creación de narrativas interactivas o la programación de juegos sencillos en 3D

A diferencia de muchas de las aplicaciones de codificación basadas en rompecabezas ("puzzle-based"), Alice motiva el aprendizaje a través de la exploración creativa.

Alice está diseñada para enseñar habilidades de pensamiento lógico y computacional, principios fundamentales de la programación y para ser una primera exposición a la programación orientada a objetos.



Crear un juego con Alice 3 – Parte 1

Topic	Detalles
Visión general	En este tutorial, practicaré el uso de las estructuras de control IF y WHILE para controlar la ejecución de procedimientos y funciones en función de una condición verdadera o falsa.
Conceptos clave	<ul style="list-style-type: none">• Utilice la estructura de control IF para afectar a la ejecución de instrucciones• Utilizar la estructura de control WHILE para crear un bucle condicional para un comportamiento repetitivo• Uso de operadores lógicos en bucles
Dificultad	Intermedia: este tutorial es adecuado para alguien que ya haya utilizado Alice 3 para: <ul style="list-style-type: none">• Agregación y colocación de objetos en una escena• Sentencias de programación de códigos en el editor de códigos• Randomización de movimientos• Declaración de un procedimiento definido por el usuario• Crear una animación que utilice una sentencia de control para controlar la sincronización de la animación• Utilizar funciones para controlar el movimiento basado en un valor de retorno
Duración	60 minutos
Notas	Este tutorial se ha creado con Alice 3.4.0

Crear un juego con Alice 3 – Parte 1

Parte 1: Definir el escenario

Revise el escenario y complete la animación correspondiente.

En este tutorial, va a crear una animación a partir de este escenario


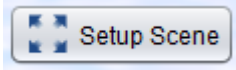
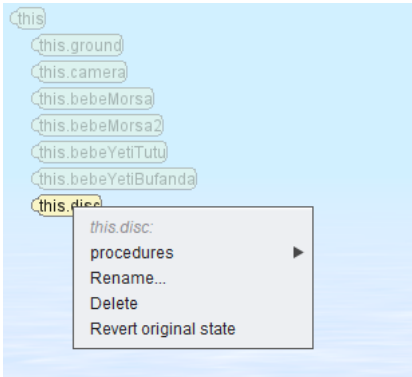
Escenario
Dos morsas corren hacia un estanque y el que llega al estanque primero se sumerge. Cada vez que se reproduce la animación, una morsa diferente puede ganar.


Diseño del programa

Para crear una animación para el escenario proporcionado, debe planificarla antes de iniciar cualquier codificación.

1. Debe pensar en los personajes que tendrá en la animación.
2. Planifique las posiciones de inicio de los personajes.
3. Planifique las acciones de los personajes.
4. Planifique el script que se utilizará para la conversación.

1. Personajes
2 morsas, 2 yeti bebé, 1 estanque.
2. Posiciones de inicio
El estanque está en el centro del mundo
Los dos yeti bebé están de pie a la derecha del estanque.
Las dos morsas están a la misma distancia del centro del estanque detrás del mismo mirando hacia él.
3. Acciones de los personajes
Uno de los yeti bebé comienza la carrera
Los dos yetis se giran
Ambos personajes de morsas corren hacia el estanque
La morsa que llegue allí primero salta en el estanque.
4. Script
Yeti bebe 1: "¿Estáis listas para correr?"
Las dos morsas: "Listas!"
Yeti bebe 1: "Preparadas, listas, ya"
Morsa ganadora: "¡Yujú! ¡Gané! Vamos a nadar".

Parte 2: Añadir Objetos al Escenario	
Completa las siguientes tareas:	
1	Crea una carpeta en tu ordenador o en un USB Pendrive con nombre Proyecto Alicia 3. Será donde guardes todos tus proyectos.
2	Inicia (doble-clic) Alice 3
	
3	Selecciona la plantilla Snow (Nieve) y pulsa clic en OK .
4	En el menú File (Archivo) selecciona Save As (Guardar Como) e introduce el nombre del Proyecto_Hielo_para_verte_Hielo_Parte_1 . Haz clic en Save (Guardar).
5	Haz clic en Setup Scene (Configurar Escena) para colocar a los personajes en sus posiciones iniciales.
	
6	Desde la pestaña Browse Gallery by Theme (Galería por temas), Snow theme (tema Nieve) añade los siguientes objetos a tu mundo: <ul style="list-style-type: none"> • 2 Bebé Morsas (Baby Walruses): llámalos bebeMorsa y bebeMorsa2 • 1 Bebé Yeti con Tutu (Baby Yeti with Tutu) • 1 Bebé Yeti con bufanda (Baby Yeti with Scarf) Nota: No te preocupes aún por la posición de los personajes porque los colocaremos después
7	En la pestaña Shapes/Text (Formas/Texto) añade el siguiente objeto a tu mundo: <ul style="list-style-type: none"> • Disc (Disco) Pon la propiedad del color de pintura a CYAN. Nota: No te preocupes aún por las posiciones de los objetos.
8	En el object tree (árbol de objetos), haz clic en el botón derecho en el disco y selecciona Rename (Renombrar) Pon el nombre de estanque al disco azul.
	
9	Renombra el bebé yeti con bufanda a bebeYetiBufanda .

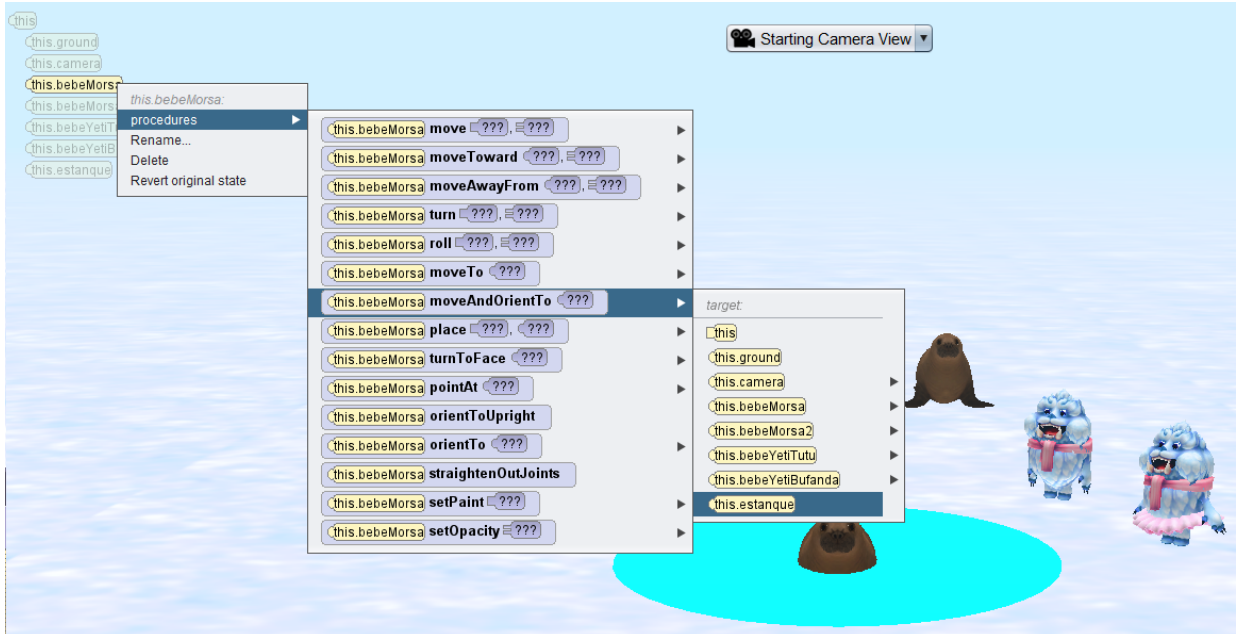
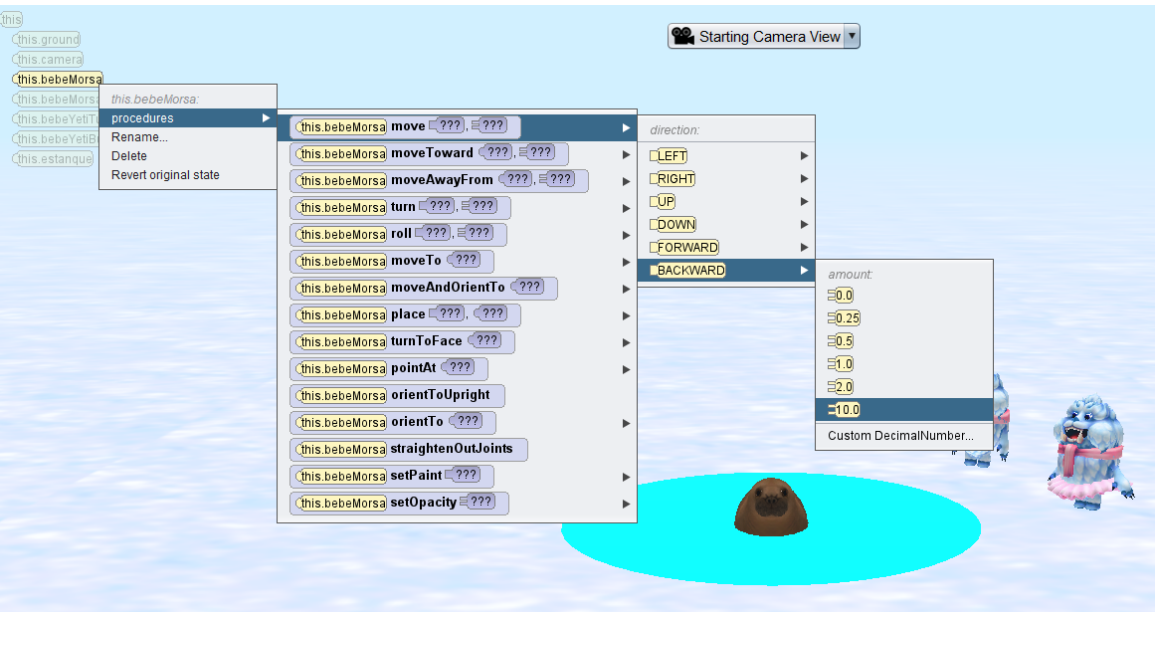
10	Renombra el bebé yeti con tutu a bebeYetiTutu .
11	<p>Utiliza el botón Resize handle (Redimensionar) para cambiar el tamaño del estanque y que los dos bebés morsa puedan zambullirse en él.</p> 





12	Utiliza las herramientas (handle style) de Default y Rotation para mover a los yetis a la derecha del estanque (tu derecha), un poco orientados hacia el estanque
----	--



13	En los siguientes pasos, colocarás con precisión las morsas para el inicio de la carrera, de modo que cada morsa esté a la distancia exacta del centro del estanque. Así, nos aseguramos de que las morsas tendrán una salida justa.
----	--

13a	Haz clic con el botón derecho en el objeto bebeMorsa y haz que se mueva hacia el estanque con moveAndOrientTo estanque .
	
13b	Esto moverá la morsa al centro del estanque. Puede parecer que la morsa se hunde en el estanque; lo solucionarás en un paso posterior.
13c	Haz clic con el botón derecho en el objeto bebeMorsa2 y haz que se mueva también al estanque con moveAndOrientTo estanque .
13d	Las dos crías de morsa deben estar una encima de la otra en el centro del estanque. Puede parecer que se hunden en el estanque. Arreglarás esto en un paso posterior, para que estén sentadas en el suelo.
13e	<p>Haz clic con el botón derecho del ratón sobre la bebeMorsa. Nota: Utiliza el árbol de objetos a la izquierda para seleccionar el bebé morsa correcto ya que están una encima de la otra en la escena.</p> <p>Selecciona Procedures (Procedimientos). Selecciona move (mover).</p> <p>Selecciona los argumentos: BACKWARD → 10.0.</p>
	

13f	Mueve también el bebeMorsa2 hacia atrás (backwards) 10 metros.
13g	Las dos morsas tienen que estar precisamente a 10 metros del estanque una sobre la otra.
	
13h	Mueve al bebeMorsa a la izquierda (left) 1.0 metro.
13i	Mueve al bebeMorsa2 a la derecha (right) 1.0 metro.
	
13j	Ahora subiremos las dos bebés morsas para que se posen en el suelo en lugar de estar en parte enterradas.

Haz clic el handle style **Move**.

Haz clic en **bebeMorsa**. Utiliza las **flechas de arriba** que aparecen sobre la cabeza del bebé morsa para moverlo hacia arriba, para que se sienta en el suelo y puedas ver sus aletas. Para ello selecciona la flecha de arriba y luego arrastra hacia arriba.


Haz lo mismo con **bebeMorsa2**



13k Con el **bebeMorsa** seleccionado hazlo girar y encarar (**turnToFace**) el estanque.

13L Con el **bebeMorsa2** seleccionado hazlo girar y encarar (**turnToFace**) el estanque.

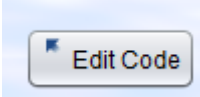
13m Cuando termines, tu escena debería verse así:



14 Guarda la animación dándole a **Save**.

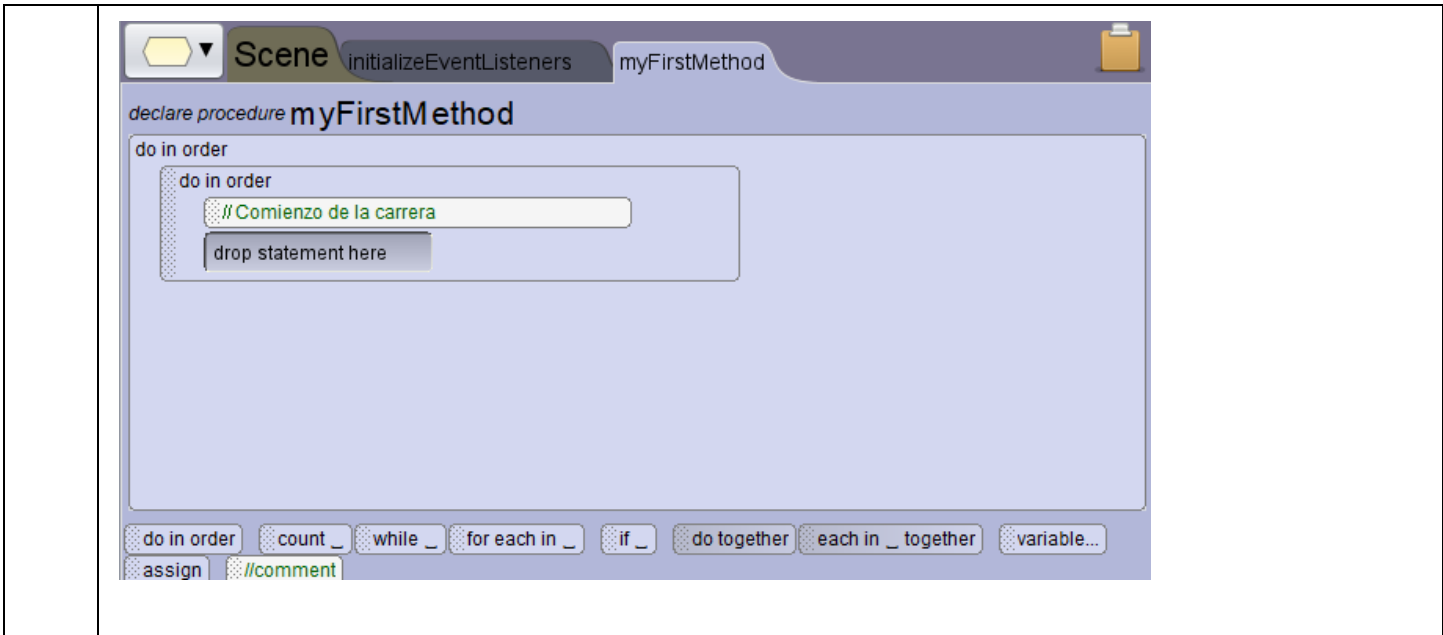
Parte 3: Programar el comienzo de la carrera

15 Ve al editor de Código hacienda click en el botón **Edit Code**.



16 Arrastra un **do in order** desde la barra de abajo a la derecha en **myFirstMethod**.

17 Arrastra un comentario (**comment**) dentro de la sentencia **do in order** que diga, “**Comienzo de la carrera**”.



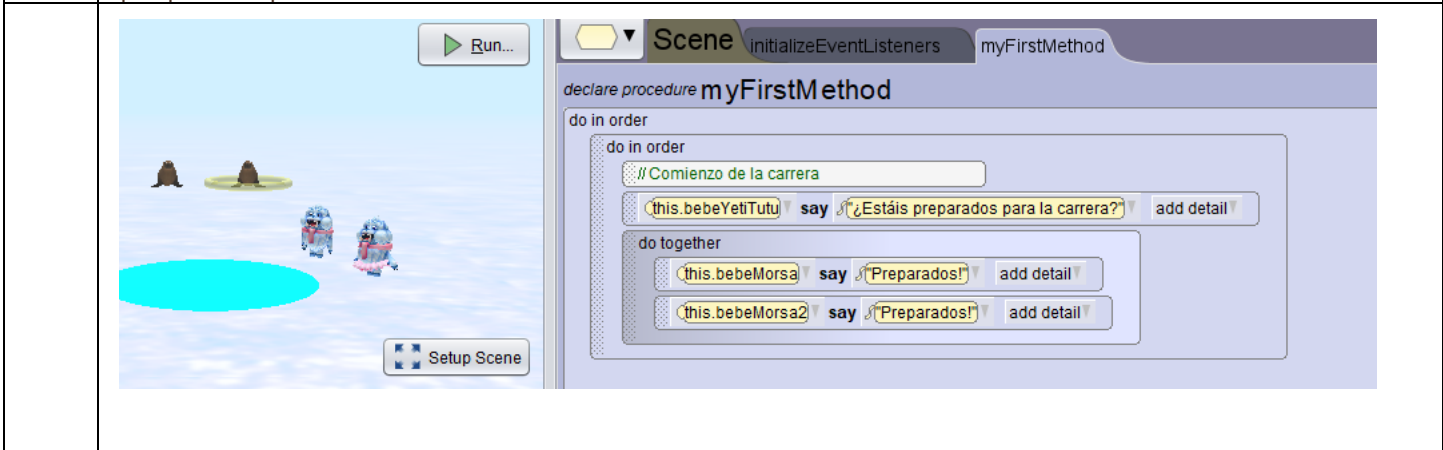
18 Haz que el **bebeYetiTutu** diga “¿Estáis preparados para la carrera?” en la sentencia **do in order**.
Nota: tendrás que seleccionar Custom TextString

19 Arrastra un **do together** dentro del **do in order** debajo del paso **say**.

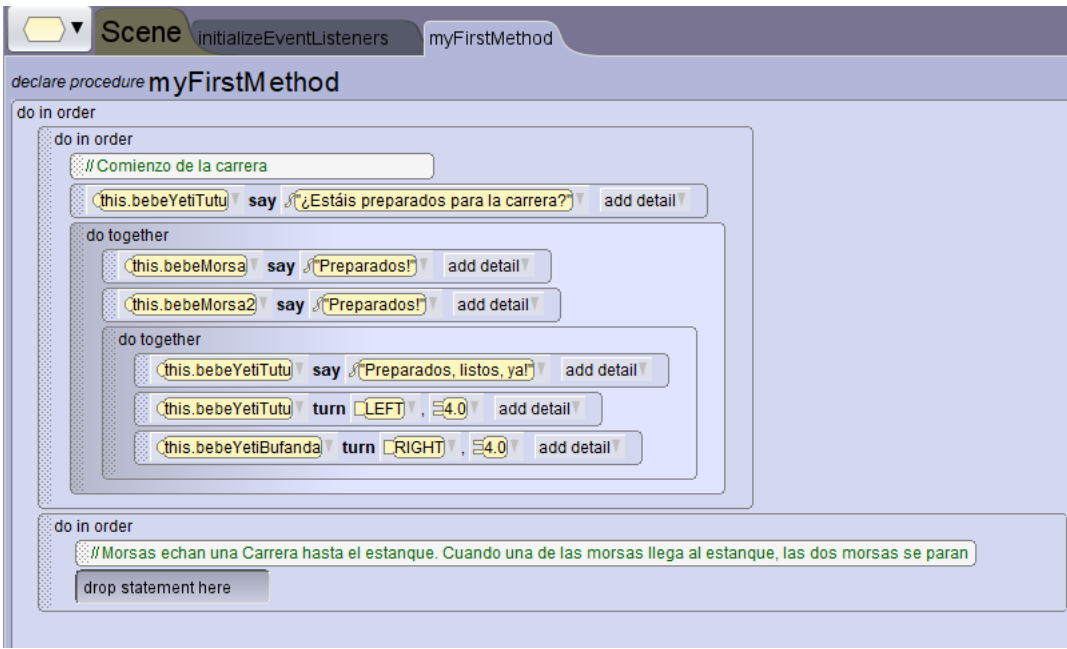


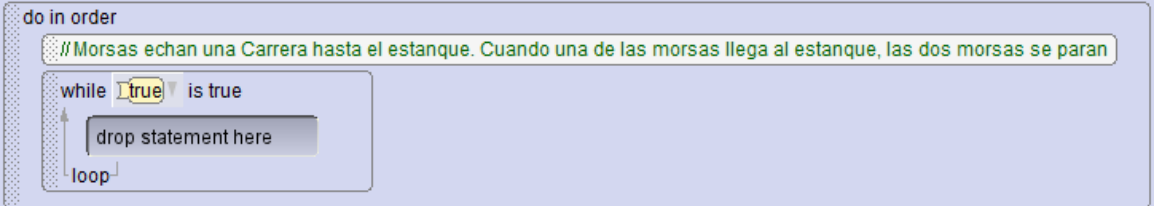
20 Haz que el **bebeMorsa** diga “Preparados!” en la sentencia **do together**.

21 Copia el procedimiento **say**. Pégalo directamente debajo del existente y cambia la referencia de **walrusBaby** a **walrusBaby2**.
CONSEJO: Recuerda que la manera más rápida de hacer esto es presionando la Tecla **CTRL** y arrastrando la sentencia que quieres copiar con el ratón.

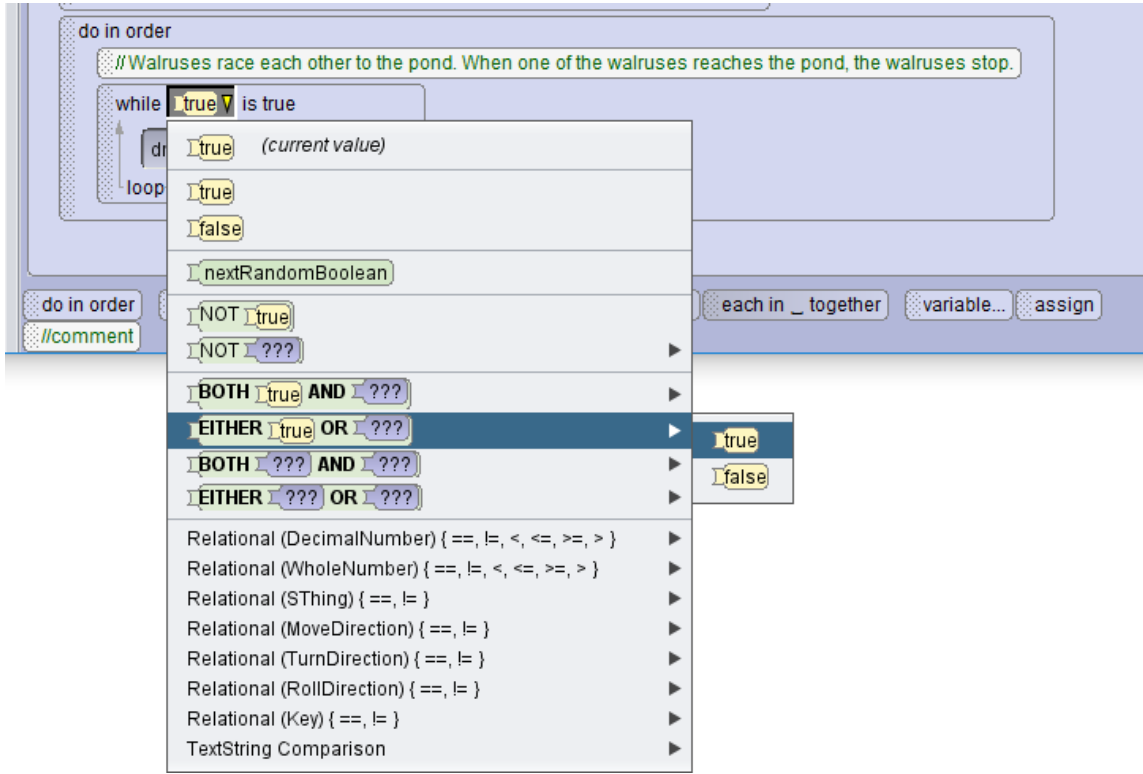


22	Arrastra otra sentencia do together bajo la existente dentro del mismo do in order .
23	Haz que el bebeYetiTutu diga “ Preparados, listos, ya! ” en la nueva sentencia do together .
24	Con el bebeYetiTutu seleccionado haz que se gire 4.0 con turn left by 4.0 en la sentencia do together .
25	Copia el procedimiento turn . Pegalo directamente bajo el existente y cambia el objeto de referencia de bebeYetiTutu a bebeYetiBufanda . Cambia la dirección de izquierda (LEFT) a derecha (RIGHT)
26	Guarda y ejecuta la animación.

Parte 4: Codificando la Carrera.	
27	Arrastra otra sentencia do in order bajo la que ya tienes en myFirstMethod .
28	Arrastra un comentario al do in order que diga “Morsas echan una Carrera hasta el estanque. Cuando una de las morsas llega al estanque, las dos morsas se paran”
	
29	Arrastra una sentencia de control while en el do in order . Selecciona la condición true como un valor marcador (esto se cambiará en el siguiente paso).

	 <p>The screenshot shows a code editor with a light blue background. At the top, it says "do in order". Below that is a comment in Spanish: "//Morsas echan una Carrera hasta el estanque. Cuando una de las morsas llega al estanque, las dos morsas se paran". Underneath is a "while true is true" loop. Inside the loop is a box labeled "drop statement here". A vertical arrow on the left side of the loop indicates the flow, and the word "loop" is written at the bottom left of the loop structure.</p>
30	<p>Ahora debes especificar la condición bajo la cual el bucle while se ejecuta (mientras que las morsas no lleguen al estanque) continuarán moviéndose hacia delante de forma aleatoria.</p> <p>Una vez que la condición deja de ser verdad (true), el bucle while dejará de ejecutarse.</p>

30a Selecciona la **flecha hacia abajo** cerca de la condición true.
 Selecciona **EITHER true OR ???**
 Selecciona **true**.



30b Ahora cambiarás la condición para usar el operador lógico NOT de manera que la declaración se ejecute si la condición es falsa.
 Seleccione la **flecha hacia abajo más exterior** cerca de la condición.
 Selecciona **NOT EITHER true OR true**.

```

do in order
  // Walruses race each other to the pond. When one of the walruses reaches the pond, the walruses stop.
  while EITHER true OR true is true
    // do in order
    //comment
  loop
  
```

EITHER true OR true (current value)
 BOTH true AND true (replace operator)
 true (reduce to left operand only)
 true (reduce to right operand only)
 true
 false
 nextRandomBoolean

NOT EITHER true OR true
 NOT ???
 BOTH EITHER true OR true AND ???
 EITHER EITHER true OR true OR ???
 BOTH ??? AND ???
 EITHER ??? OR ???

Relational (DecimalNumber) { ==, !=, <, <=, >=, > }
 Relational (WholeNumber) { ==, !=, <, <=, >=, > }
 Relational (SThing) { ==, != }
 Relational (MoveDirection) { ==, != }
 Relational (TurnDirection) { ==, != }
 Relational (RollDirection) { ==, != }
 Relational (Key) { ==, != }
 TextString Comparison

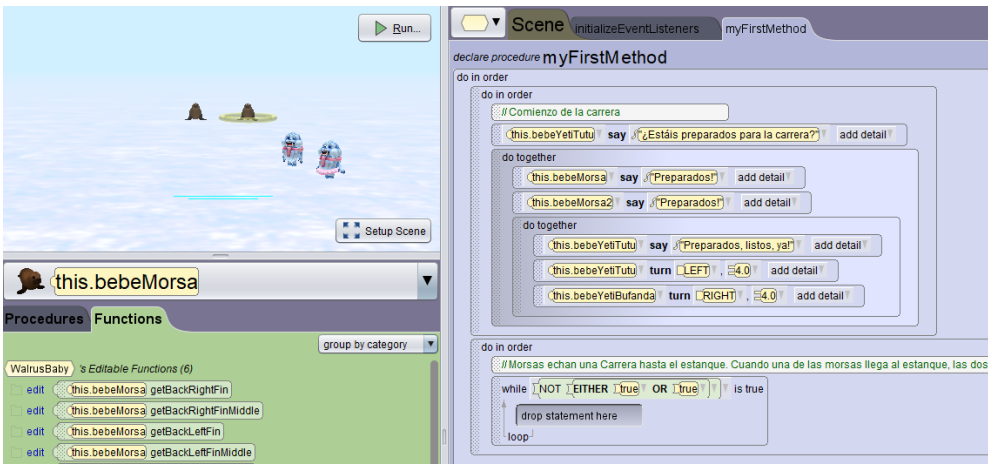
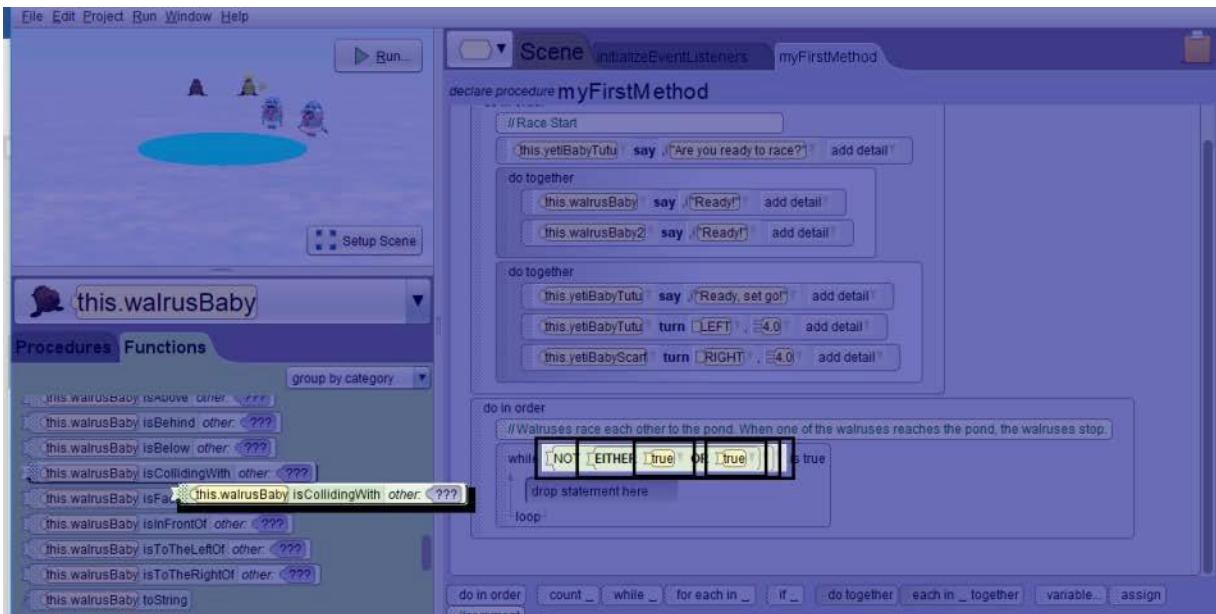
each in _ together variable... assign

30c Selecciona el **bebeMorsa** del menú **instance**

The screenshot shows a game engine interface. On the left, a scene view displays a blue sky and water with several walrus characters. Below the scene view is a panel for the selected object, 'this.bebeMorsa', showing a list of procedures and functions. On the right, a code editor window is open, showing the implementation of a procedure named 'myFirstMethod'. The code includes comments and logic for starting a camera and controlling the camera's movement based on the state of the 'bebeMorsa' object.

```

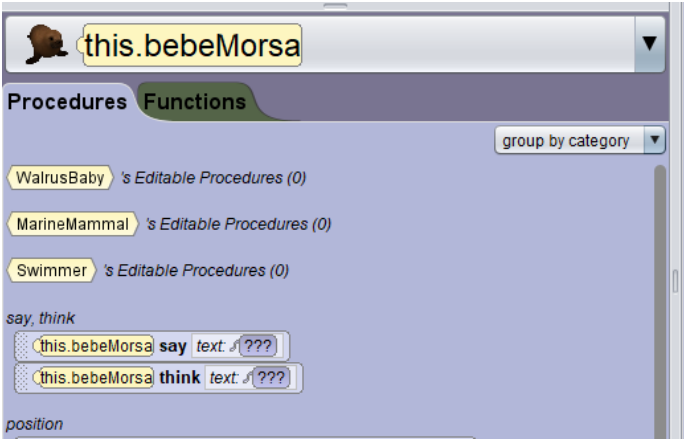
declare procedure myFirstMethod
do in order
  // Comienzo de la camera
  this bebeMorsa say "¿Estás preparado para la carrera?" add detail!
  do together
    this bebeMorsa say "¡Preparados!" add detail!
    this bebeMorsa say "¡Preparados!" add detail!
  do together
    this bebeMorsa say "¡Preparados, listos, ya!" add detail!
    this bebeMorsa turn LEFT 54.0 add detail!
    this bebeMorsa turn RIGHT -54.0 add detail!
  do in order
    // Morsas echan una Camera hasta el estanque. Cuando una de las morsas llega al estanque.
    while NOT (EITHER true OR true) is true
      // stop statement here
    loop
  
```

30d	<p>Haz clic en la pestaña Functions</p> 
30e	<p>Ve hacia abajo hasta que encuentres la función isCollidingWith (está chocando con) y arrastra hacia la primera condición true en la condición de while. Selecciona this.estanque.</p>
30f	 <p>Selecciona bebeMorsa2 del menu instance. Arrastra la función isCollidingWith hacia la segunda condición true en while. Selecciona de nuevo el this.estanque.</p>
30g	<p>Esta condición completa significa “Que ni el bebeMorsa ni el bebeMorsa2 estén chocando con el estanque”.</p> <p>Mientras ninguna morsa entre a la piscina, el bucle (ciclo o loop) continuaría ejecutándose.</p>

```

do in order
  // Morsas echan una Carrera hasta el estanque. Cuando una de las morsas llega al estanque, las dos morsas se paran
  while NOT EITHER (this.bebeMorsa isCollidingWith (this.estanque) OR (this.bebeMorsa2 isCollidingWith (this.estanque)) is true
    drop statement here
  loop
  
```

31 Ve a la pestaña **Procedures**.



32 Arrastra un **do together** (desde la barra de abajo) dentro del bucle **while**.

```

do in order
  // Morsas echan una Carrera hasta el estanque. Cuando una de las morsas llega al estanque, las dos morsas se paran
  while NOT EITHER (this.bebeMorsa isCollidingWith (this.estanque) OR (this.bebeMorsa2 isCollidingWith (this.estanque)) is true
    do together
      drop statement here
    loop
  
```

33 Mueve la **bebeMorsa** hacia adelante (**forward**) **1.0** metro.
NOTA: 1.0 es un valor provisional y será cambiado en el siguiente paso.

34 Ahora, vas a **modificar** el **argumento distance** en el procedimiento **move** de tal forma que sea un valor **aleatorio** entre **0.25** y **2.0** sea seleccionado **cada vez** que el procedimiento **move** es **ejecutado**.

- 34a Haz clic en la **flecha** que se encuentra junto al argumento de distance **1.0** en el procedimiento **move**.
- Selecciona **Random**.
 - Selecciona **nextRandomRealNumberInRange ??? ???**
 - Selecciona **0.25** para el valor bajo.
 - Selecciona **2.0** para el valor alto.

do in order

```
//Morsas echan una Carrera hasta el estanco. Cuando una de las morsas llega al estanco, las dos morsas se paran
```

while NOT EITHER (this.bebeMorsa isCollidingWith (this.estanco) OR (this.bebeMorsa2 isCollidingWith (this.estanco))) is true

do together

- (this.bebeMorsa) move FORWARD -1.0 add detail
- drop statement here

loop

amount: 1.0 (current value)

- 0.0
- 0.25
- 0.5
- 1.0
- 2.0
- 10.0

Random

- nextRandomDecimalNumberInRange 0.0, 1.0
- Whole to Decimal Number
- Math
- Custom DecimalNumber...

- 0.0
- 0.25
- 0.5
- 1.0
- 2.0
- 10.0
- Custom DecimalNumber...

Custom DecimalNumber...

Custom DecimalNumber...

34b	Quedaría así:
	
34c	<p>Copia el procedimiento move pegándolo directamente debajo del que ya existe.</p> <p>NOTA: Recuerda hacerlo presionando la tecla CTRL mientras se arrastra el procedimiento abajo</p>
	
34d	Cambia el objeto referencia de bebeMorsa a bebeMorsa2
	
34e	Quedaría así:

	<pre> do in order // Morsas echan una Carrera hasta el estanque. Cuando una de las morsas llega al estanque, las dos morsas se paran while NOT EITHER (this.bebeMorsa isCollidingWith this.estanque) OR (this.bebeMorsa2 isCollidingWith this.estanque) is true do together this.bebeMorsa move FORWARD, nextRandomDecimalNumberInRange 0.25, 2.0 add detail this.bebeMorsa2 move FORWARD, nextRandomDecimalNumberInRange 0.25, 2.0 add detail loop </pre>
35	<p>Guarda (Save) y ejecuta (Run) la animación.</p> <p><i>Vas a ver que las focas se paran cuando una de ellas alcanza el estanque. También vas a ver que ellas se mueven hacia adelante una distancia aleatoria, y esto va a cambiar cada vez que ejecutes la animación.</i></p> <p><i>Si las focas no se paran, puede ser que tengas que bajarlas un poco hacia el suelo, ya que puede ser que las hayas colocado demasiado alto y estén pasando por encima del estanque y, por tanto, no están colisionando con ella.</i></p>

Parte 5: Celebración de la carrera

36 Coloca una instrucción “do in order” al final de “myFirstMethod” con un comentario que diga: “Las morsas celebran el final de la carrera. Entonces, se tiran al agua.”

```

declare procedure myFirstMethod
do in order
  do in order
    // Comienzo de la carrera
    this.bebeYetiTutu say ¿Estáis preparados para la carrera? add detail
    do together
      this.bebeMorsa say ¡Preparados! add detail
      this.bebeMorsa2 say ¡Preparados! add detail
    do together
      this.bebeYetiTutu say ¡Preparados, listos, ya! add detail
      this.bebeYetiTutu turn LEFT, 4.0 add detail
      this.bebeYetiBufanda turn RIGHT, 4.0 add detail
  do in order
    // Morsas echan una Carrera hasta el estanque. Cuando una de las morsas llega al estanque, las dos morsas se paran
    while NOT EITHER (this.bebeMorsa isCollidingWith this.estanque) OR (this.bebeMorsa2 isCollidingWith this.estanque) is true
      do together
        this.bebeMorsa move FORWARD, nextRandomDecimalNumberInRange 0.25, 2.0 add detail
        this.bebeMorsa2 move FORWARD, nextRandomDecimalNumberInRange 0.25, 2.0 add detail
      loop
  do in order
    // Las morsas celebran el final de la carrera. Entonces, se tiran al agua
    drop statement here
  
```

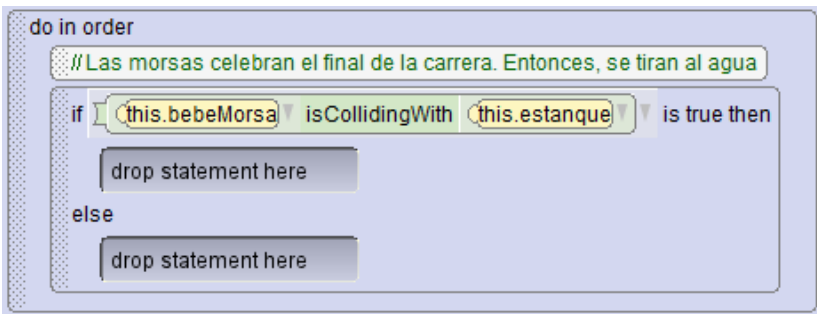
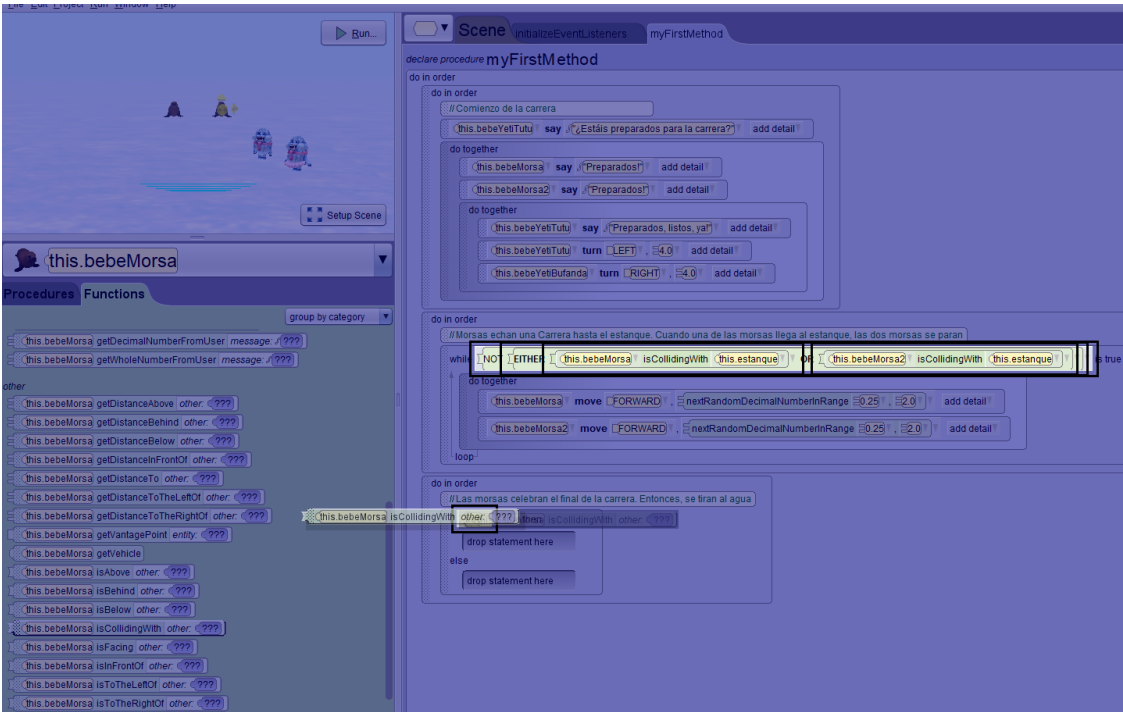
37 Arrastra un “if” dentro de la instrucción “do in order”. Selecciona “true” como valor por el momento momento.

```

do in order
  // Las morsas celebran el final de la carrera. Entonces, se tiran al agua
  if true is true then
    drop statement here
  else
    drop statement here
  
```

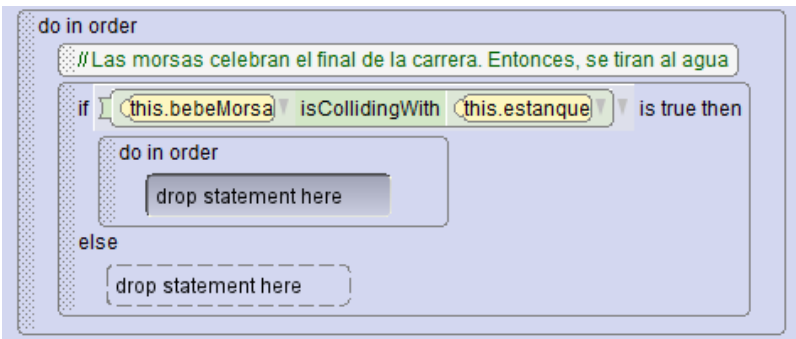
38 Selecciona el **bebeMorsa** del menu de la instancia y haz clic en la tab “Functions”.

39 Arrastra la funcion “isCollidingWith” dentro de la condicion “true” seleccionando el “estanque” como el otro valor.



40 Haz click en la pestaña “**Procedure**”.

41 Arrastra una instrucción “**do in order**” a la primera seccion “**drop statement here**” dentro de la instrucción “**if**”.



42 Ahora vas a programar lo que le pasa al objeto “**bebeMorsa**” después de la carrera.

42a Haz que el “**bebeMorsa**” diga “**Genial, he ganado! Vamos a darnos un chapuzon!**” dentro de la instrucción “**do in order**”. Haz que el texto aparezca durante **2 segundos**.

NOTA: para que dure 2 segundos hay que darle a la flecha junto a **add detail** y elegir **duration** y **2**

42b Mueve el “**bebeMorsa**” hacia **delante (forward) 1.0** metro.
NOTA: 1.0 es el valor por defecto, se cambiara en el siguiente paso

42c De la pestaña “**Functions**”, arrastra una funcion **getDistanceTo** encima del valor 1.0 para la instruccion “**move**”. Elige el “**estanque**” como valor “**target**”.
NOTA: Esto asegura que la morsa siempre ira al centro del estanque, no importa de donde acabe la carrera.

42d Selecciona la pestaña “**Procedures**” y arrastra de la barra de abajo una instruccion “**do together**” dentro del “**do in order**”.

42e Con **bebeMorsa** seleccionado hazlo girar (**turn**) hacia delante (**forward**) 0.25 dentro del “**do together**”.

42f Mueve (**move**) el **bebeMorsa** hacia delante (**forward**) **2.0** metros. Haz que la accion tarde **2** segundos en ejecutarse

```

do in order
  //Las morsas celebran el final de la carrera. Entonces, se tiran al agua
  if (this.bebeMorsa isCollidingWith this.estanque) is true then
    do in order
      this.bebeMorsa say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
      this.bebeMorsa move FORWARD , 1.0 add detail
      do together
        this.bebeMorsa turn FORWARD , 0.25 add detail
        this.bebeMorsa move FORWARD , 2.0 , duration 2.0 add detail
    else
      drop statement here
  
```

43 Ahora vas a programar el mismo movimiento exacto para **bebeMorsa2**. Para eso no hace falta repetir todos estos pasos!

43a **Copia y pega** la instrucción "do in order" presionando la tecla (CTRL) y arrastrando hacia abajo en la seccion "drop statement here" de la instruccion else.

43b El Código se debe ser así:

```

do in order
  //Las morsas celebran el final de la carrera. Entonces, se tiran al agua
  if (this.bebeMorsa isCollidingWith this.estanque) is true then
    do in order
      this.bebeMorsa say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
      this.bebeMorsa move FORWARD , 1.0 add detail
      do together
        this.bebeMorsa turn FORWARD , 0.25 add detail
        this.bebeMorsa move FORWARD , 2.0 , duration 2.0 add detail
    else
      do in order
        this.bebeMorsa say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
        this.bebeMorsa move FORWARD , 1.0 add detail
        do together
          this.bebeMorsa turn FORWARD , 0.25 add detail
          this.bebeMorsa move FORWARD , 2.0 , duration 2.0 add detail
      
```

43c **Cambia** las referencias de objeto en cada sentencia de la condición **else** de **bebeMorsa** a **bebeMorsa2**.

```

do in order
  //Las morsas celebran el final de la carrera. Entonces, se tiran al agua
  if (this.bebeMorsa isCollidingWith this.estanque) is true then
    do in order
      this.bebeMorsa say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
      this.bebeMorsa move FORWARD , 1.0 add detail
      do together
        this.bebeMorsa turn FORWARD , 0.25 add detail
        this.bebeMorsa move FORWARD , 2.0 , duration 2.0 add detail
    else
      do in order
        this.bebeMorsa say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
        this.bebeMorsa (current value) , 1.0 add detail
        this.bebeMorsa
        this.bebeMorsa2 FORWARD , 0.25 add detail
        this.bebeYetiTutu FORWARD , 2.0 , duration 2.0 add detail
        this.bebeYetiBufanda
        this.estanque
  
```

```

do in order
  //Las morsas celebran el final de la carrera. Entonces, se tiran al agua
  if (this.bebeMorsa isCollidingWith this.estanque) is true then
    do in order
      this.bebeMorsa say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
      this.bebeMorsa move FORWARD , 1.0 add detail
      do together
        this.bebeMorsa turn FORWARD , 0.25 add detail
        this.bebeMorsa move FORWARD , 2.0 , duration 2.0 add detail
    else
      do in order
        this.bebeMorsa2 say "Genial, he ganado! Vamos a darnos un chapuzon!" , duration 2.0 add detail
        this.bebeMorsa2 move FORWARD , 1.0 add detail
        do together
          this.bebeMorsa2 turn FORWARD , 0.25 add detail
          this.bebeMorsa2 move FORWARD , 2.0 , duration 2.0 add detail
  
```

44 **Guarda** y ejecuta (**Run**) varias veces la animación.

NOTA: Observa lo que sucede cuando cada Morsa toca primero el estanque. La morsa que toca el estanque primero habla y luego se sumerge en el estanque. A continuación, finaliza la animación