Conceptos fundamentales de Java

2-10: Variables

Proyecto

Este proyecto avanzará con el usuario a lo largo del curso. Después de cada lección, habrá más contenido para agregar hasta que se cree una animación completa que puede cargar en YouTube o exportar como un archivo de animación local.

Si no ha completado la tarea 9, descargue el archivo Fish_9.a3p que encontrará en Oracle iLearning en la computadora.

Objetivos de la lección:

- Utilizar números aleatorios para definir aleatoriamente el movimiento
- Comprender las variables
- Comprender cómo se utilizan las variables en la programación
- Ver el código Alice como código Java en la parte lateral

Instrucciones:

1. Abra Alice 3 en la computadora.
2. Mediante el separador My Projects o el separador File System, busque y abra el archivo Fish_9.a3p.
3. Mediante el comando Save As del menú File, cambie el nombre del archivo a Fish_10.a3p.
4. Si no está ya en el editor de código, utilice el botón Edit Code para ir al editor de código.

   Para esta lección vamos a ver las variables y el comportamiento aleatorio en el código.

   Actualmente tiene un bloque de código en myFirstMethod que hace que todos los peces agiten la cabeza al mismo tiempo.

   Sin embargo, esta acción solo la hacen una vez y convendría que el pez agitara la cabeza un número de veces aleatorio cuando se ejecuta el código.

Las animaciones son mucho más llamativas si muestran un comportamiento aleatorio cuando corresponda.

5. Arrastre una sentencia count justo encima de esta sentencia do together y elija 3 como el valor de marcador.

6. Arrastre la sentencia do together al bucle de recuento.

7. De este modo, los peces agitan la cabeza tres veces cada vez que se ejecuta el código. Para hacer que el valor sea aleatorio, haga clic en la flecha junto al valor del marcador y, a continuación, seleccione random y elija la opción aleatoria que le permita seleccionar el valor inferior y superior inclusivo que desee utilizar como rango aleatorio. Seleccione 1 y 3 como los valores de argumento.

   Este código permitirá a los peces agitar la cabeza una vez como mínimo y tres veces como máximo.

8. Pruebe el código ejecutándolo varias veces. Cuente el número de veces que los peces agitan la cabeza.

9. A continuación, debe hacer que los procedimientos sean más útiles agregando variables al código. Ya ha utilizado este concepto anteriormente mediante los argumentos que ha empleado en sus procedimientos.

   Haga clic en el separador swim. Si no aparece, haga clic en el botón classes y, a continuación en FISH y haga clic en swim.

10. La clase swim está muy restringida actualmente porque solo permite al pez nadar a una longitud determinada (1.2).

    Para cambiarlo, puede agregar una variable en la parte superior del bloque de código.

11. Arrastre la sentencia variable desde la parte inferior de la pantalla y colóquela en la parte superior del código. Recuerde que la línea verde muestra el lugar donde se colocará al soltarla. Es decir, en la parte superior de la sentencia do in order.

12. Seleccione la variable, elija DecimalNumber como el tipo de datos y asígnele el nombre distance. Establezca el valor inicial en 0 a través de la opción del menú customDecimalNumber.

13. De este modo obtendrá una variable declarada e inicializada en el código denominada distance con un valor inicial de 0.0.

14. Guarde el programa.

15. Para asignar la variable al valor, arrastre el nombre de la variable (distance) sobre los valores del marcador actual para 1.2. Deberá realizar este paso dos veces para ambas sentencias forward move del pez payaso.

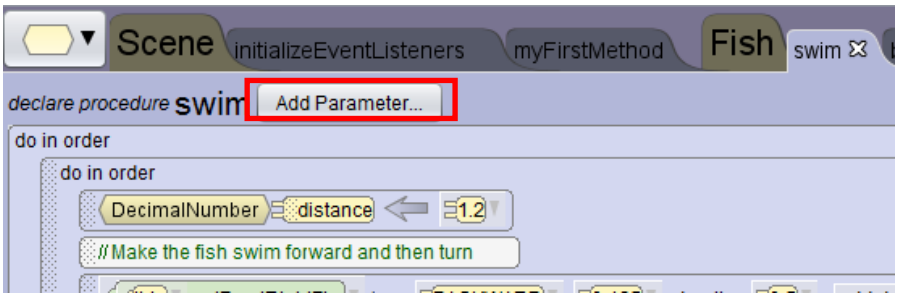
16. Ejecute el código y pruébelo.

17. El pez payaso no va a ninguna parte porque hemos establecido la distancia en 0. Cambie el valor a 1.2 y vuelva a probar el código.

Puede ver la ventaja de utilizar variables aquí. Si desea cambiar la distancia a la que nada el pez, solo necesita cambiar el valor en un lugar (en el nivel variable) y los cambios se ejecutarán a lo largo de todo el procedimiento.

Puede ir un paso más allá si transfiere un parámetro al procedimiento. Un parámetro es la información transferida de un procedimiento a otro. Puede crear un código que permita a los peces especificar la distancia a la que desea nadar cada uno cuando llamen al procedimiento swim.

18. Haga clic en el botón Add Parameter.



19. Si lo va a utilizar en lugar de la variable distance. Es necesario que sea un DecimalNumber para que funcione en el bloque de código y requiere un nombre que no se haya utilizado aún en el código. Asigne un nombre al parámetro dist y, a continuación, marque el cuadro para indicar que entiende que es necesario actualizar las invocaciones (llamadas) a este procedimiento. Haga clic en OK.

20. Suprima la línea de código que creó la variable distance.

    Las instancias de la variable que existían en el código se muestran ahora en color rojo.

21. Para asignar el valor del parámetro para que funcione en el código, arrastre el nombre del parámetro (dist) a las ubicaciones en color rojo. En el bloque de código.

22. Haga clic en el separador myFirstMethod.

23. La llamada al método swim ahora tiene un valor rojo para el parámetro en el procedimiento.

24. Utilice el menú customDecimalNumber para cambiarlo a un valor de 1.2.

25. Ejecute el código para probarlo.

    Ahora tiene un procedimiento que le permitirá hacer posible que cualquier pez nade. El pez Blue Tang (Cirujano azul) actualmente nada hacia atrás y hacia adelante. Ahora puede utilizar el procedimiento swim para darle una apariencia más realista.

26. Arrastre un procedimiento swim de Blue Tang a la parte superior del código. Utilice la misma expresión para que el pez pueda nadar 3 veces su propia longitud.

27. Desactive el código existente que permite a Blue Tang moverse y ejecute y pruebe el código.

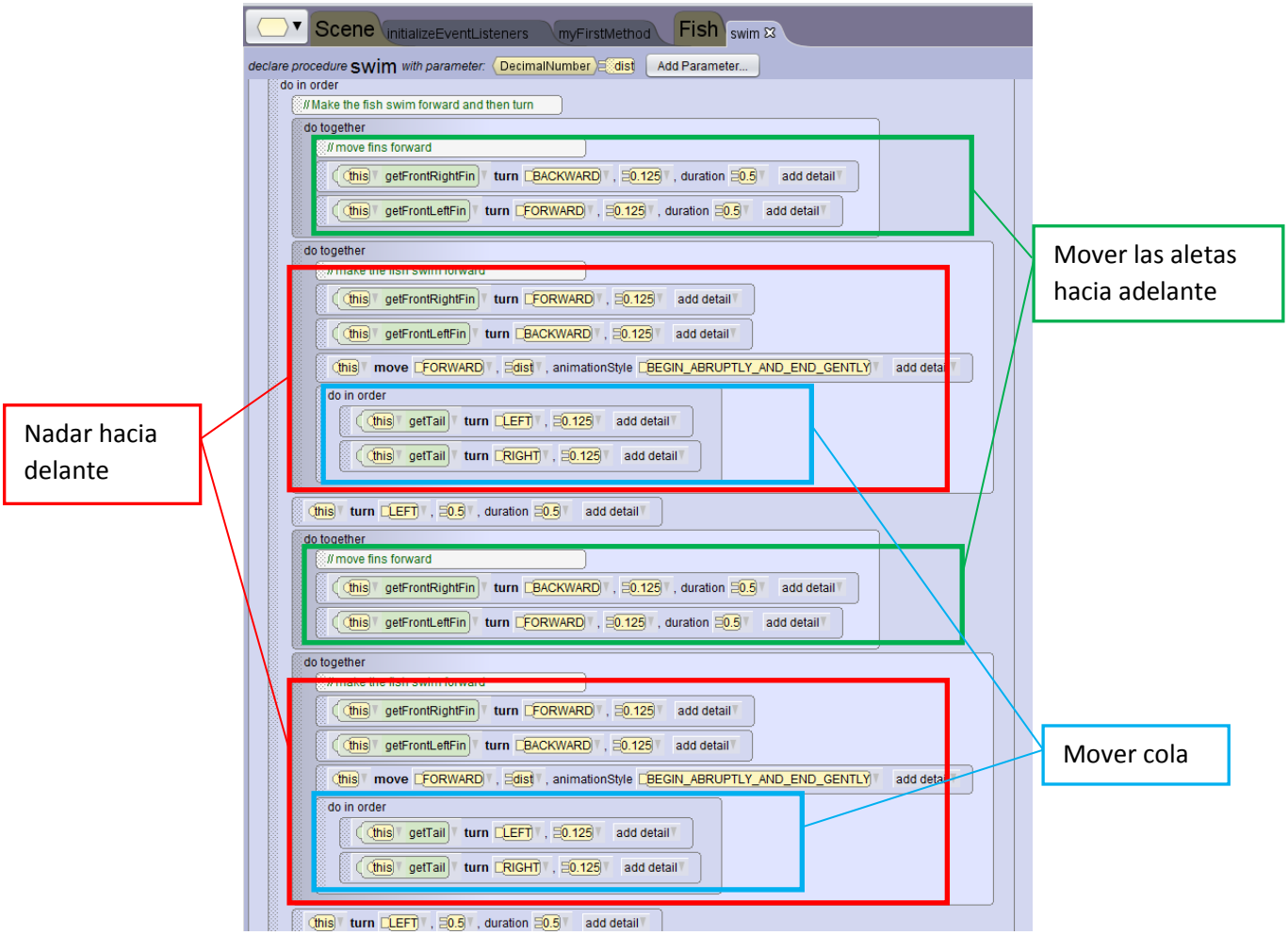
28. Suprima el código desactivado si está satisfecho con los resultados de la prueba.

29. El procedimiento swim se creó para el pez payaso, ya que solo se podía ver un lado a la vez. Si va a utilizarlo para otros peces, será necesario que mueva ambas aletas al mismo tiempo y también la cola. Adapte el procedimiento swim para que cumpla lo siguiente:

30. Agregue comentarios adicionales al código para explicar lo que hace cada sección.

    La siguiente sección de este ejercicio trata sobre la limpieza del código y la eliminación de cualquier duplicación de código en la medida de lo posible. Si nos fijamos en el procedimiento swim, puede ver que el mismo código aparece varias veces. Esto es algo que debe tratar de evitar como programador.

    Lo que debe hacer es identificar el código que se repite y eliminarlo con su propio procedimiento.

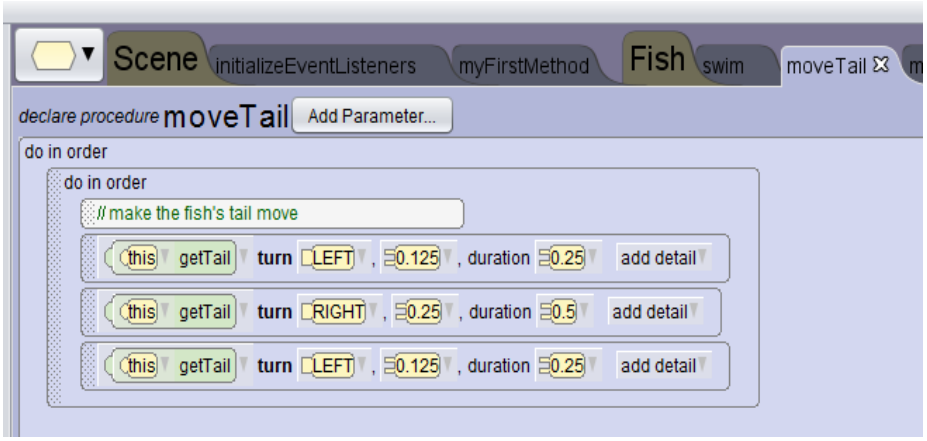


31. Puede ver que hay tres partes en este código; mover las aletas hacia adelante, mover la cola y nadar hacia adelante. Todas estas partes pueden eliminarse y colocarse en sus propios procedimientos.

32. Mediante el botón classes, cree un nuevo procedimiento en el nivel FISH llamado moveTail.

33. Arrastre al portapapeles una de las sentencias do in order que controle la cola.

34. Haga clic en el separador moveTail y arrastre allí el código. De este modo se realiza un movimiento simple que puede mejorar simplemente creando un flujo de eventos similar al procedimiento shakeHead. Modifique el código en el procedimiento moveTail de modo que coincida con el siguiente:



35. Vuelva a hacer clic en el separador swim.

36. Elimine el otro bloque de código que movía la cola y sustitúyalos por procedimientos moveTail.

37. Cree otro procedimiento FISH y asígnele el nombre moveFinsForward.

38. Arrastre al portapapeles uno de los bloques de código que mueve las aletas hacia adelante.

39. Haga clic en el separador moveFinsForward y arrastre el código del portapapeles al editor de código.

40. Haga clic en el separador swim y elimine el código para mover las aletas hacia adelante y sustitúyalo en cualquier lugar en el que se llame a las aletas para que se muevan hacia adelante con el nuevo procedimiento moveFinsForward.

41. Cree otro procedimiento FISH y asígnele el nombre swimForward.

42. Arrastre uno de los bloques de código do together al portapapeles que mueve el pez hacia adelante.

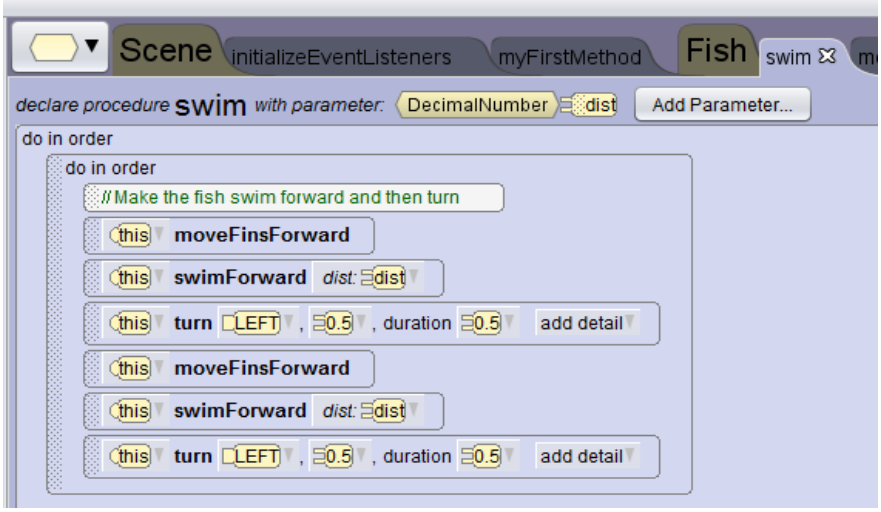
43. Haga clic en el separador swimForward y arrastre el código del portapapeles al editor de código.

44. Éste es algo distinto porque requiere un parámetro, así pues cree un parámetro dist aquí exactamente como lo hizo anteriormente. Arrastre el nombre del parámetro sobre la etiqueta roja para completar el procedimiento.

45. Haga clic en el separador swim y sustituya de nuevo cualquier instancia del bloque de código swim por el nuevo procedimiento swimForward.

    Al arrastrarlo al editor de código, se le pedirá que proporcione un valor para el parámetro. Elija dist en la lista.

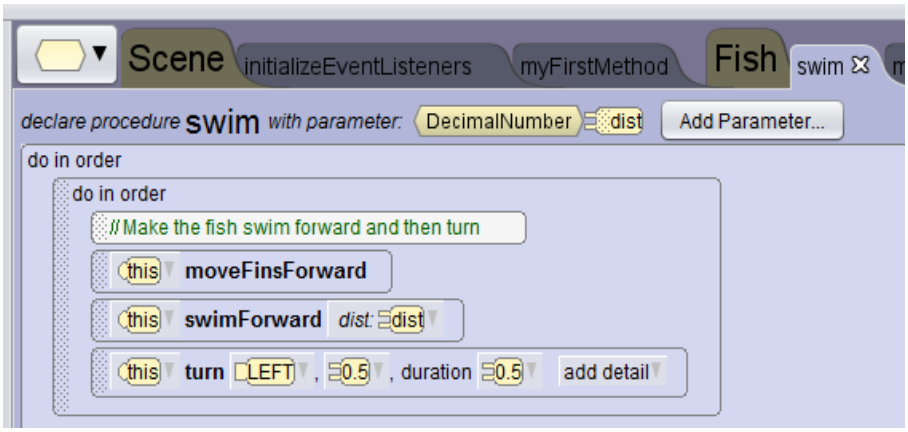
46. De este modo obtendrá una clase swim driver que coloca juntos todos los elementos de swimming desde los distintos procedimientos disponibles.



También significa que dispone de mucha más flexibilidad en cuanto a lo que puede hacer con un pez.

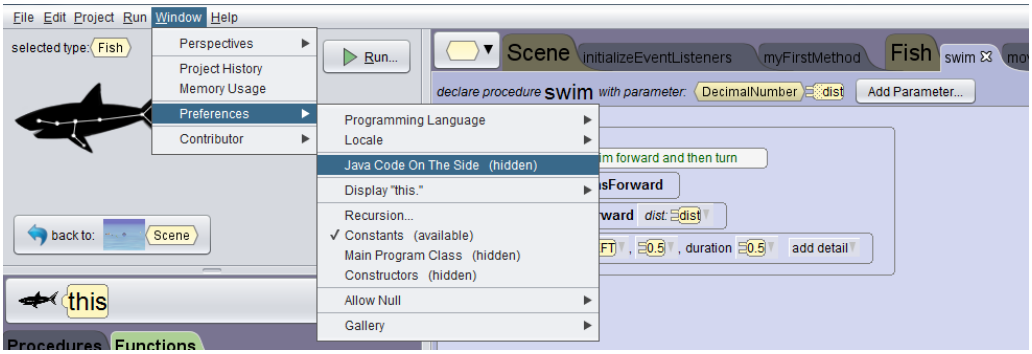
Un último punto a tener en cuenta aquí es que el procedimiento swim hace que el pez nade, gire y vuelva a nadar antes de realizar un giro final. Lo que puede hacer aquí es reducir aún más este código suprimiendo de nuevo el código que se repite.

47. Elimine las líneas de código hasta que el procedimiento swim cumpla lo siguiente:

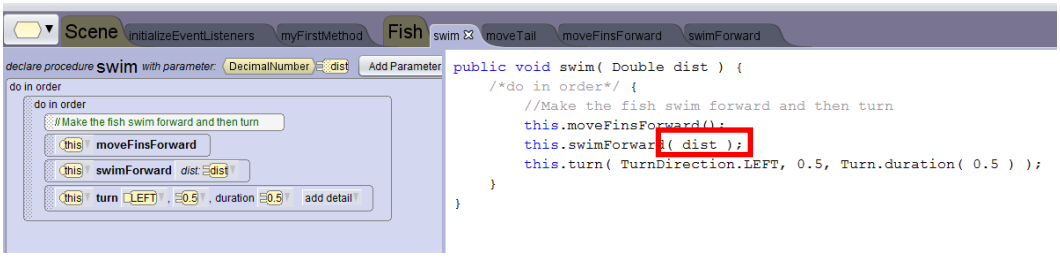


Mucho menos aquí que cuando comenzamos y mucho más fácil de leer y entender. La diferencia aquí es que el pez solo realizará una única operación swim.

48. Si desea ver el aspecto que tendría este código en Java, puede activar la opción Java Code on The Side en las opciones del menú Window.



Puede resultar útil consultar este código antes de continuar con Greenfoot y Eclipse, ya que se podrá hacer una idea del aspecto que tiene el código Java y de cómo funciona.



Puede ver que el parámetro dist se transfiere utilizando paréntesis. Es así como se transfieren los parámetros en Java. Si nos fijamos en el método turn, podemos ver que al transferir varios parámetros, estos se separan con comas. Puede obtener consejos útiles de codificación estudiando la opción Java Code on The Side.

49. Si desea desactivar la opción Java Code on The Side, siga los mismos pasos que utilizó para activarla.

50. Si el pez solo nada una vez y necesita que nade dos veces para que la animación funcione correctamente, haga clic en el separador myFirstMethod.

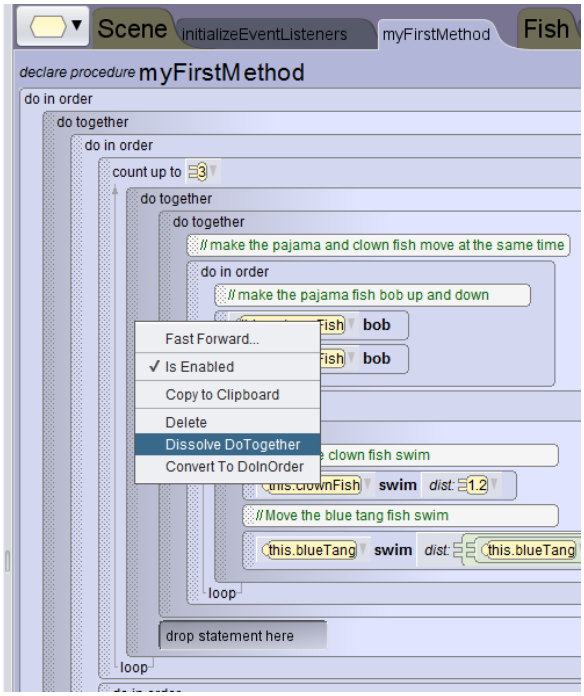
51. Ahora que en el procedimiento solo se nada una vez, puede hacer que el código lo llame dos veces utilizando una sentencia count y seleccionando 2 como argumento. Colóquelo encima de la llamada al procedimiento swim y, a continuación, arrastre allí la llamada swim.

52. Haga que el pez Blue Tang se mueva del mismo modo arrastrando el procedimiento swim de Blue Tang en la sentencia del bucle count junto al procedimiento swim del pez payaso.

53. Cambie el valor del pez Blue Tang de 3 a 2 para que deje de nadar fuera de la pantalla.

54. Agregue una sentencia do together para que ambos peces naden al mismo tiempo.

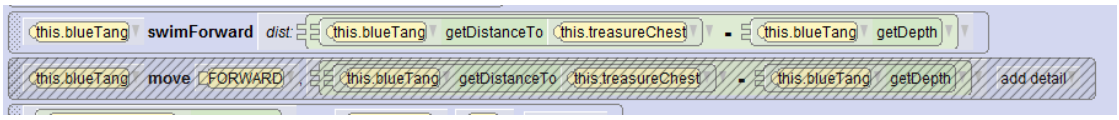
55. Puede ver que ahora hay dos sentencias do together aquí, una de las cuales ahora es redundante. Puede disolver esta acción haciendo clic con el botón derecho fuera de la sentencia do together y seleccionando dissolve en el menú.



56. El pez Blue Tang agita la cabeza y, a continuación, dice: "no more swimming today" (no sigue nadando). Sería mejor si estas acciones se llevasen a cabo de manera conjunta. En Alice, puede realizar fácilmente la conversión entre sentencias. Haga clic con el botón derecho en esta sentencia do in order y seleccione Convert to Do Together.

57. Se crean varios procedimientos que controlan el modo en el que el pez se mueve en el agua. Ahora puede utilizarlos para que el movimiento del pez sea más realista.

    Ahora, cualquier procedimiento move forward del pez puede sustituirse por un procedimiento swimForward para que se muevan las aletas y la cola mientras el pez nada. Desactive todos los procedimientos move forward del pez y sustitúyalos por procedimientos swimForward utilizando los mismos argumentos.

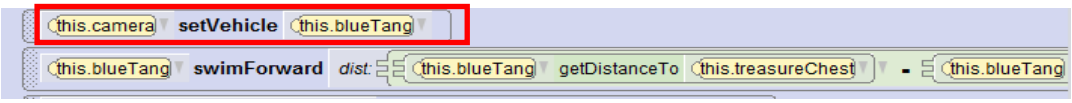


58. Elimine todos los procedimientos desactivados cuando haya probado sus sustituciones de swimForward.

    Y ya solo faltan unos cuantos retoques cinemáticos antes de que finalice el proyecto y estará listo para subirlo a YouTube.

59. Cuando el pez Blue Tang se mueva hacia el cofre del tesoro, deseo que la cámara se aleje al mismo tiempo. Para ello, solo tiene que configurar el vehículo de la cámara del mismo modo que Blue Tang.

    Arrastre un procedimiento setVehicle de la clase Camera al editor de código antes de que el pez Blue Tang se mueva hacia el cofre del tesoro.

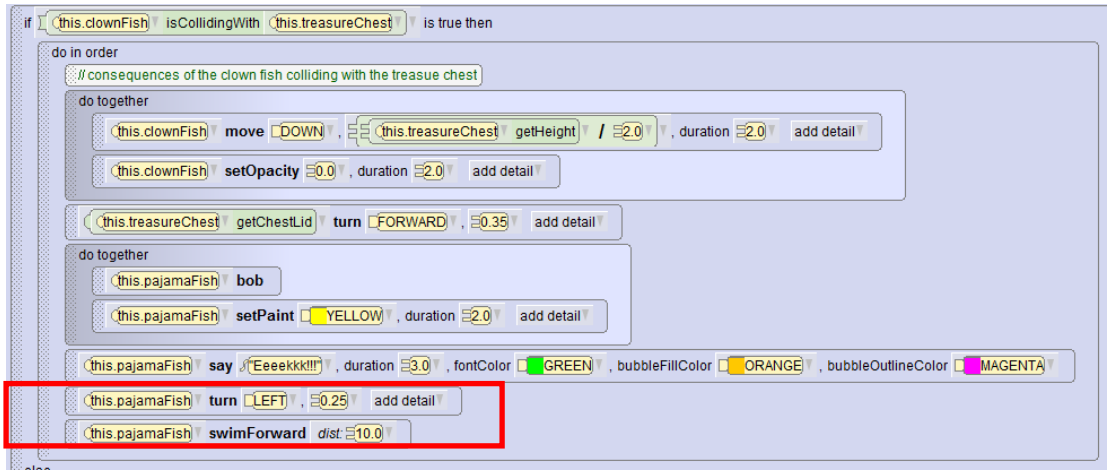


60. La cámara debe permanecer en su posición antes de que el pez Blue Tang se aleje nadando.

    Pulse CTRL y arrastre el mouse para copiar el código set vehicle de la cámara y soltarlo encima del código turn. Cambie el valor del argumento por este. De este modo se suelta la cámara desde Blue Tang.

61. Deseo que la cámara se mueva próxima al cofre del tesoro mientras los peces payaso y pajama (Pijama) nadan hacia él. Arrastre una sentencia moveToward al código do together utilizando el cofre del tesoro y el 2 como argumentos.

62. En lugar de que el pez pajama solo flote en el fondo, agregue un procedimiento turn left y un procedimiento swimForward utilizando el valor 10 como el argumento al final de la sentencia if.



    Correcto el segundo y último elemento de este proyecto. El cofre del tesoro debe esperar unos segundos antes de regresar al fondo del mar. Mientras el cofre desciende, la cámara debe apartarse de la escena.

63. Agregue una sentencia do in order después de la sentencia if.

64. Coloque una sentencia delay desde el cofre del tesoro hasta la sentencia do in order. Utilice el valor 4 como argumento.

65. Agregue una sentencia do together bajo la sentencia delay.

66. Arrastre un procedimiento move desde el cofre del tesoro que está descendiendo a la altura del mismo (utilice una función) y especifique el valor 5 como duración.

    Arrastre un procedimiento moveAwayFrom desde la cámara y seleccione el cofre del tesoro y 5 como argumentos. Establezca la duración en 7.

67. Ejecute y pruebe la animación.

68. Una última cosa. Agregue algún sonido cuando el pez payaso desaparezca. Seleccione el pez payaso, arrastre un procedimiento play audio a la sentencia do together donde desaparezca el pez payaso.

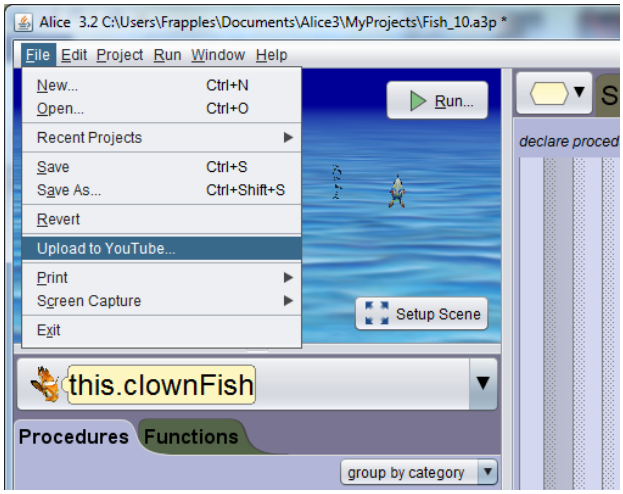
Seleccione import audio, musical cues y, a continuación, intense como audio.

69. Pruebe la animación.

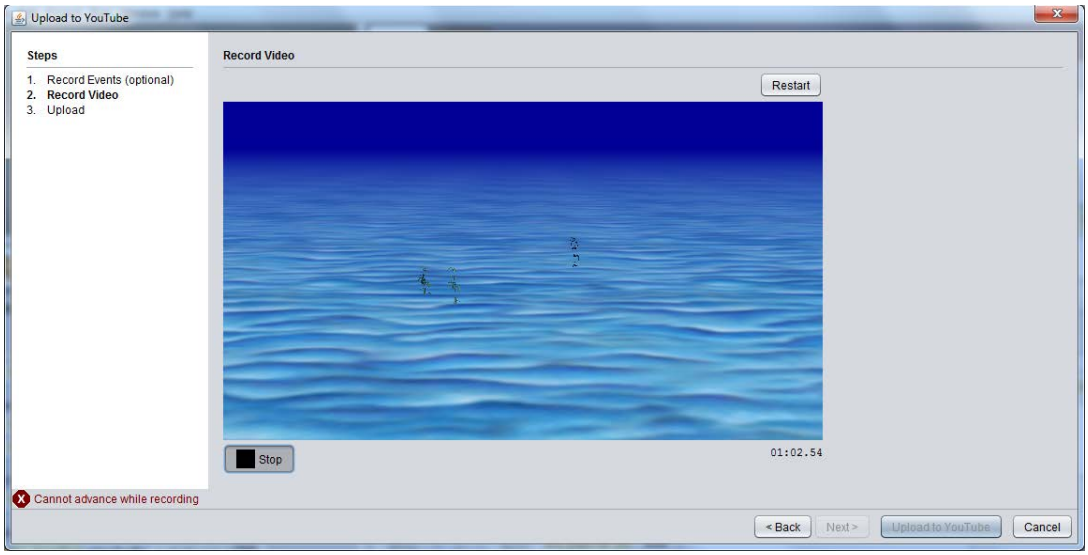
70. Guarde el programa.

71. Ahora que ya dispone de una animación completa, puede subir el producto finalizado a YouTube o exportar el archivo localmente a la máquina.

72. Haga clic en File y, a continuación, Upload to YouTube.

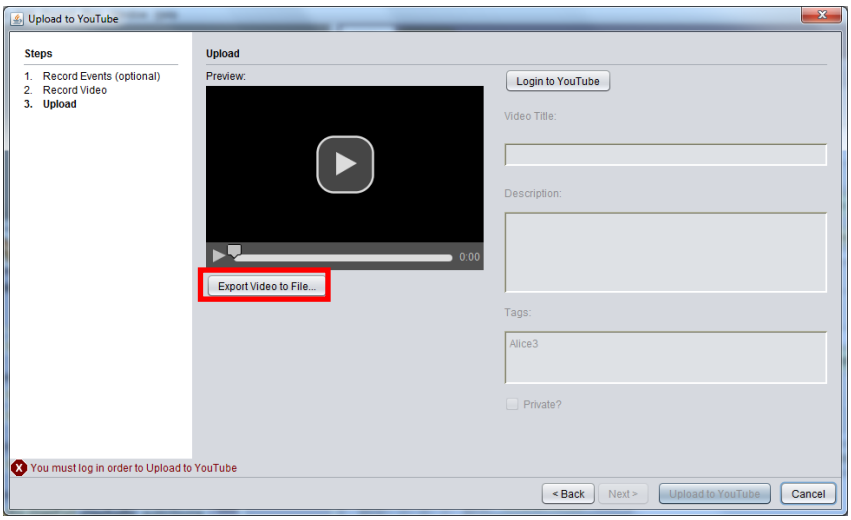


73. Pulse el botón record para comenzar a grabar la animación. Haga clic en Stop una vez que la cámara haya dejado de moverse al final.

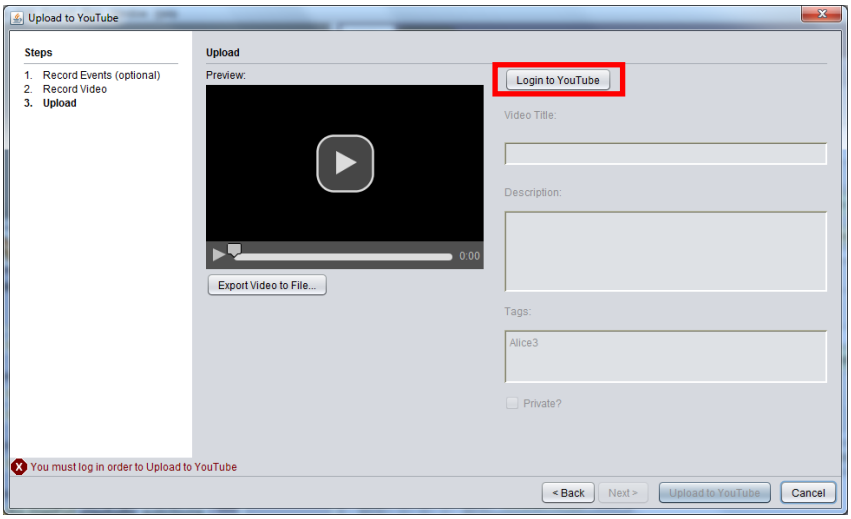


74. Haga clic en Next.

75. Si hace clic en Export Video to File, puede crear una copia local de la animación que puede reproducir en una aplicación multimedia.



76. Si hace clic en Log into YouTube, puede introducir los detalles de YouTube, proporcionar una descripción al archivo y algunas etiquetas para ayudar a encontrarlo en Internet y, a continuación, cargar la animación siguiendo unos pasos sencillos.



77. Guarde el programa.

Ya ha finalizado su primera animación completa y la ha cargado en Alice 3.

78. Salga de Alice 3.