

## Conceptos fundamentales de Java

### 2-11: Controles de teclado

#### Proyecto

Este proyecto será el proyecto final que aborda los aspectos que faltan por conocer del curso de Alice.

Si no ha completado la tarea 10, descargue el archivo Fish\_10.a3p que encontrará en Oracle iLearning en la computadora. También necesitará el archivo underTheSea\_start.a3p que también se puede encontrar en Oracle iLearning.

#### Objetivos de la lección:

- Crear una secuencia de apertura
- Utilizar controles de teclado para manipular una animación
- Guardar el archivo de clase
- Utilizar el separador inicial
- Agregar un archivo de clase existente a una animación

#### 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\_10.a3p.

En esta lección vamos a aprender a guardar un archivo de clase desde un programa y a transferirlo a otro.

También vamos a aprender a incluir interactividad en las animaciones mediante el eventos del teclado y del mouse, así como a automatizar el proceso de detección de una colisión.

3. Ya ha creado varios procedimientos transferibles dentro de la clase Fish en esta animación. Estos procedimientos se pueden guardar en un archivo para que pueda transferirlos a otra aplicación.

Haga clic en el botón classes y seleccione Fish y, a continuación, Fish. Observará que junto a Fish en el menú principal, aparece el número 6 entre paréntesis. Este es el número de procedimientos almacenados en ese nivel.

Ésta es la plantilla de la clase que detalla los procedimientos, funciones o propiedades que existen para esa clase.

4. Hacer clic en el botón Save to Class File.
5. En la carpeta MyClasses, asigne el nombre myFish.a3c al archivo.
6. En el menú File, seleccione open y busque el archivo underTheSea\_start.a3p.
7. Haga clic en Save as y utilice el nombre underTheSea.a3p para el archivo.
8. El archivo underTheSea cuenta con una escena que muestra un entorno submarino. Al utilizar un archivo creado por una tercera persona, siempre resulta útil familiarizarse con el contenido del archivo. Mediante la lista de objetos en el editor de escena, complete el siguiente diagrama escribiendo los nombres de las clases en el cuadro correspondiente de la tabla.

**Lista de contenidos de la animación**



<i>Ex: scubaHelmet</i>


Ahora que ya está familiarizado con el contenido, puede volver a la interfaz del editor de código.

Puede ver que aún no existe código en esta animación. Va a agregar la clase myFish.a3c desde la animación anterior para conseguir que el pez nade.

9. Haga clic en el botón **clases** y seleccione **Fish** y, a continuación, **Fish**.

10. Desde la plantilla **Fish**, haga clic en el botón **Add from Class File**.

11. Elija el archivo **myFish.a3c** que ha guardado anteriormente en este ejercicio.

De este modo obtendrá una lista de procedimientos que pueden copiarse en la animación. Si no desea todos los procedimientos, puede anular la selección de los que no desee desactivando la casilla. Si hace clic en el botón **Next**, podrá ver el código para los procedimientos. Haga clic en **Finish** para agregar todos los procedimientos en la animación.

12. Ahora dispone de una lista de todos los procedimientos que ha creado anteriormente para la clase **Fish** en esta animación.

13. Ahora que dispone de procedimientos predefinidos, puede implantarlos en el código para esta animación.

En el editor de código, arrastre y suelte una sentencia **while** utilizando el valor de argumento **true**.

14. Si desea que sucedan varias cosas a la vez, arrastre una sentencia **do together** al bucle **while**.

Un guión gráfico textual de muestra para esta parte sería:

El pez payaso se desplazará hacia arriba y abajo dentro del casco.

El pez payaso debe mover la cola.

El pez **Blue Tang** (Cirujano azul) debe nadar 6 metros.

El pez **Pajama** (Pijama) debe nadar 4 metros.

El tiburón debe mover la cola.

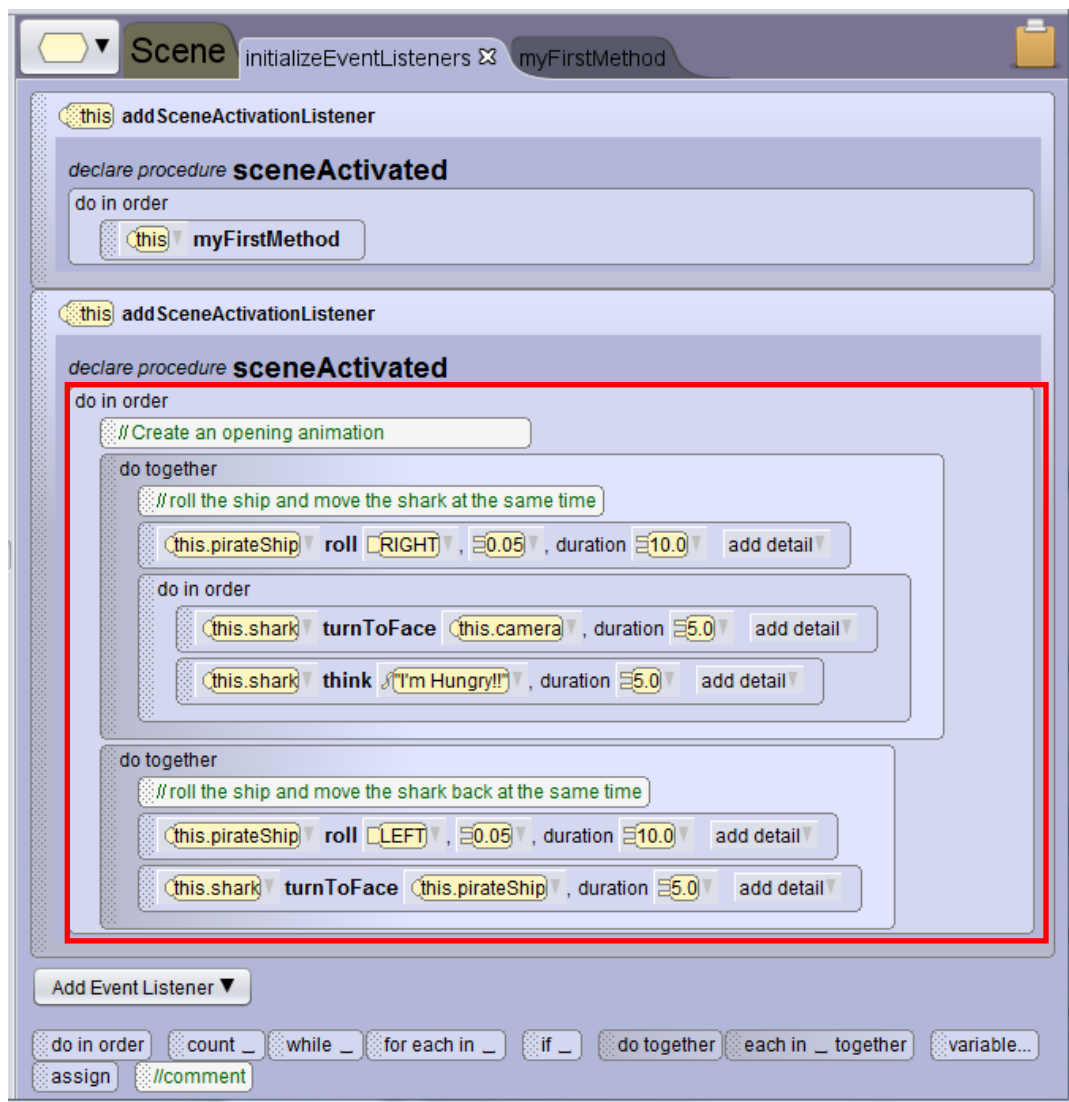
Todas estas acciones deben llevarse a cabo de forma conjunta.

15. Aplique los pasos del guión gráfico textual en Alice arrastrando los procedimientos adecuados en la sentencia **do together**.

16. Guarde el programa.

Veamos cómo se utiliza **EventListeners**. Un evento es algo que ocurre, como cuando se hace clic en el mouse, se pulsa una tecla, se inicia una aplicación, etc. El primer **EventListener** que va a utilizar es **sceneActivated**, que se ejecuta cuando iniciamos la animación por primera vez.

17. Para crear un manejador activado de la escena, haga clic en el separador initialize event listeners.
18. Haga clic en el botón add event listener y, en la lista desplegable, seleccione scene/activation/time y, a continuación, elija addSceneActivationListener.
19. A continuación, agregaremos los procedimientos que necesitamos para crear la animación.



La cronología es importante al programar las animaciones. Puede observar que todas las partes de la primera sentencia do together tardarán 10 segundos en ejecutarse, de este modo se completan las animaciones del tiburón y del barco al mismo tiempo. Para la segunda parte de la sentencia, el tiburón terminará antes que el barco.

20. Ejecute y pruebe que las animaciones y el listener activado de la escena funcionan correctamente.
21. Ahora podrá crear los manejadores de eventos para el teclado y el mouse.

A continuación, va a agregar el listener para las pulsaciones de tecla, por lo que **moveremos el submarino manualmente**.

22. Haga clic en Add Event Listener, seleccione keyboard y, a continuación, elija teclado y, a continuación, addObjectMoverFor. Seleccione Submarine (Submarino) para el objeto.

Se crea un listener de evento que le permite controlar el submarino con las teclas de flecha.

23. Ejecute y pruebe el programa moviendo el submarino. Recuerde que no se ha creado colisión del objeto en el submarino.

### **Mover el tiburón**

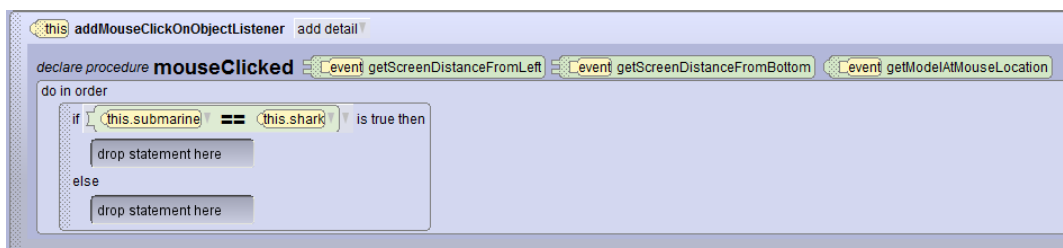
24. El usuario debe poder hacer clic en el tiburón y hacer que lleve a cabo una acción. Para ello, necesitamos utilizar un listener del mouse. Haga clic en AddEventListener, a continuación en mouse y en addMouseClickedOnObjectListener.

25. Cuando tenga el procedimiento listener, debe agregar una sentencia if para manejar el evento cuando se produzca. Seleccione el valor de marcador true.

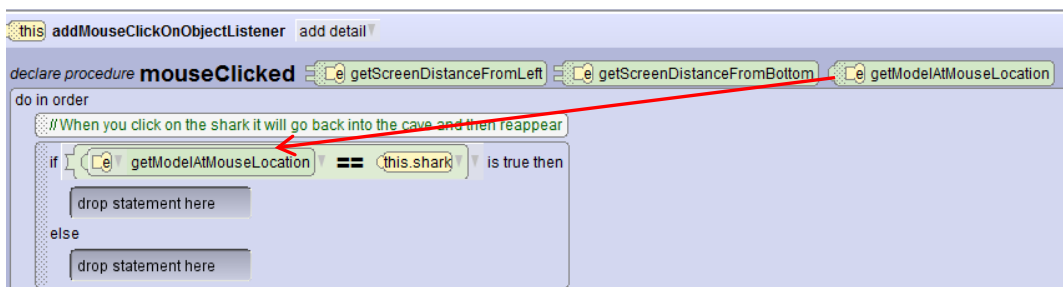
26. Haga clic en la variable true en la sentencia if y seleccione relational SThing (==) en el menú desplegable.

27. Vuelva a utilizar el valor aleatorio para un marcador para la primera parte del operador relacional, pero seleccione shark como el segundo operador.

28. De este modo obtendremos el siguiente procedimiento.



29. Ahora es necesario arrastrar el operador getModelAtMouseLocation a la variable submarine. Obtenemos lo siguiente:



30. Ahora nuestra expresión comprueba si se ha hecho clic con el mouse sobre el tiburón.
31. Agregue el código para mover el tiburón 2 metros hacia atrás durante 3 segundos. El tiburón esperará 10 segundos antes de volver a moverse hacia adelante en 2 segundos.
32. Ejecute y pruebe el programa haciendo clic en el tiburón.

### Cofre del tesoro

33. Para este objeto, debe tener en cuenta un juego de acciones más complicado.

Si hace clic en el cofre del tesoro una vez, se producirá una acción y si vuelve a hacer clic, se producirá una acción diferente. Si solo tiene un procedimiento de abrir la tapa, cada vez que haga clic en él, se abrirá aún más hasta que la tapa realice un círculo completo en torno al cofre.

34. En el caso del cofre del tesoro, la tapa se abrirá o se cerrará.

35. Cree otro objeto onMouseClickOnObjectListener.

36. Arrastre en este objeto una sentencia if.

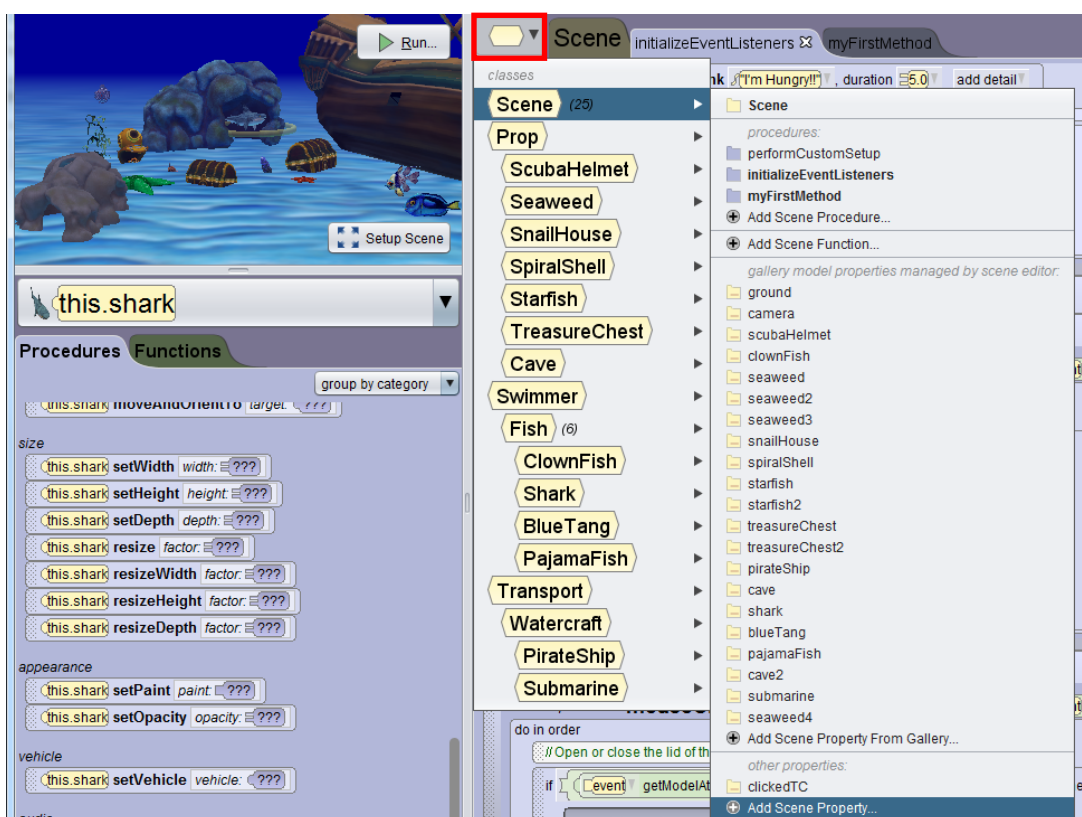
37. Elija la expresión relacional, tal y como lo hizo anteriormente con el tiburón.

38. Elija un objeto de forma aleatoria para la primera mitad de la expresión y treasureChest2 para la segunda parte.

39. Arrastre getModelAtMouseLocation al marcador.

40. Para realizar un seguimiento de si hemos hecho clic o no en el cofre del tesoro, es necesario que creamos una variable global a la que pueda acceder desde cualquier parte en el código.

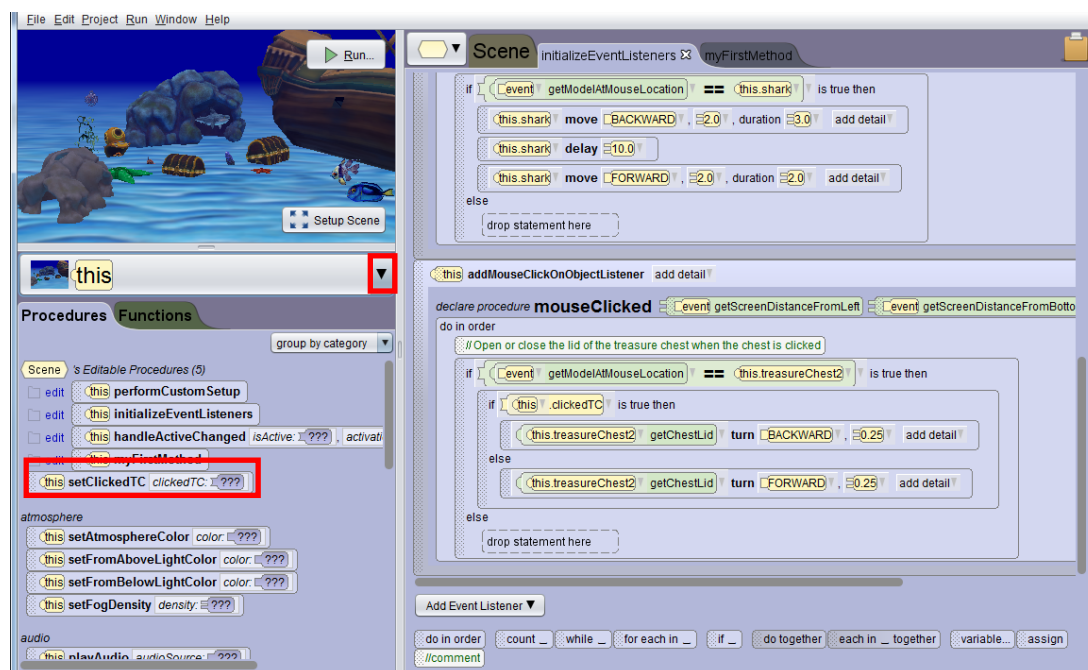
41. Para ello, haga clic en el menú desplegable de la escena, seleccione Scene y, a continuación, Add property.



42. A continuación, seleccione Variable, Boolean, clickedTC como el nombre y seleccione false como valor inicial. A continuación, haga clic en OK.

43. Vuelva a los manejadores de eventos.
44. En nuestro procedimiento del cofre del tesoro, suelte otra sentencia if.
45. Seleccione this.clickedTC como la sentencia Boolean.
46. La sentencia if no solo tiene que mover el objeto deseado, sino que también cambia el valor de la variable Boolean.
47. Deseamos abrir la tapa del cofre, así que seleccionamos esa acción en la lista de objetos (utilice el objeto adecuado y, a continuación, las flechas unidas para acceder al cofre del tesoro).
48. Agregue una sentencia turn utilizando backwards y 0.25 como valores de argumento.
49. Haga lo contrario en la sentencia else.
50. Ahora debe establecer el valor de la variable. Se inicializa como false para mostrar que no se ha hecho clic en el cofre. Cuando el usuario hace clic en el cofre, la variable debe establecerse en true y se debe abrir la tapa. Al volver a hacer clic, la variable debe establecerse en false y la tapa debe cerrarse.

Para acceder al procedimiento para cambiar el valor de nuestra variable global (clickedTC), elija **this** en la lista de objetos.



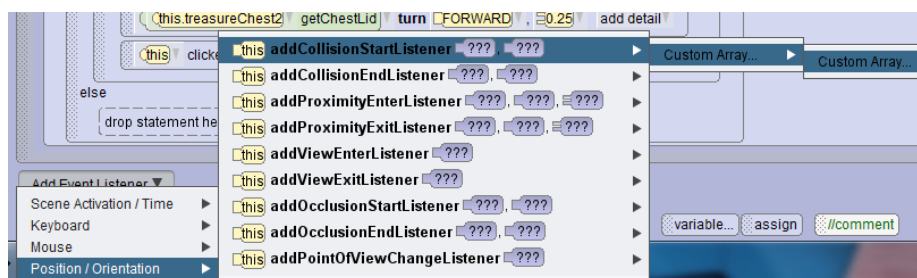
51. Arrastre el método setClicked bajo las sentencias turn de modo que el código finalizado tenga el aspecto siguiente.



### Colisiones

52. El evento de listener final para este proyecto trata de colisiones entre matrices (varias) de objetos.
53. Conseguiré que la caracola se mueva por el fondo del océano y si colisiona con otro objeto, se dará la vuelta y se alejará en la nueva dirección.

Para ello, cree un collisionStartListener.



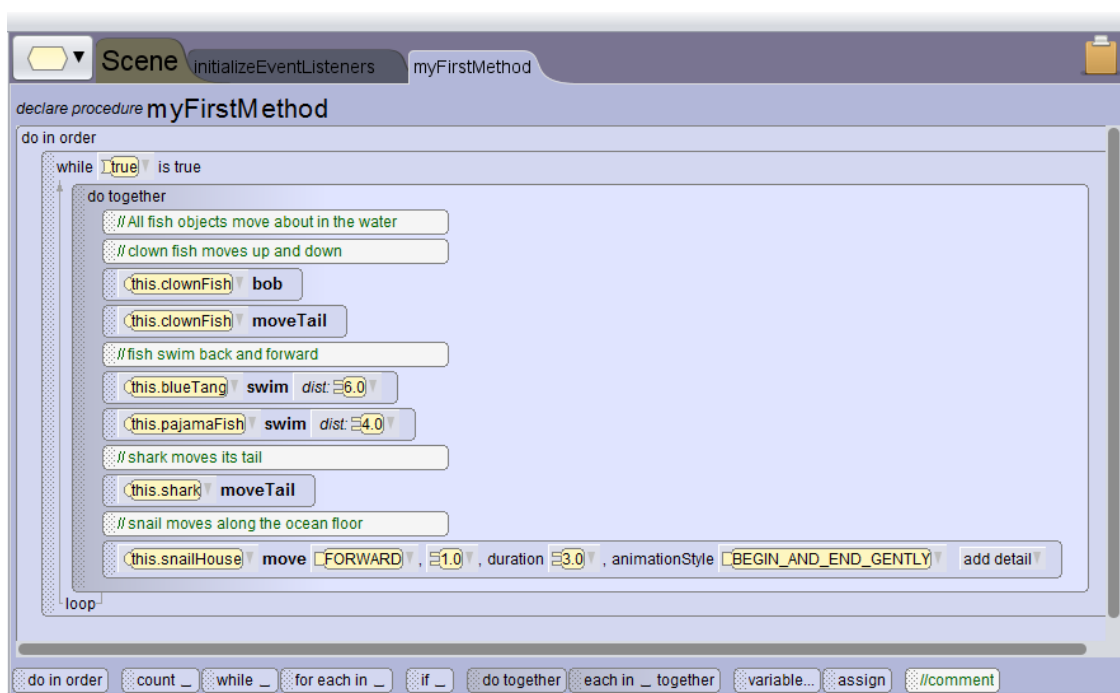
54. Para el primer elemento de la matriz personalizada, agregue la caracola, ya que ese es el objeto que va a colisionar.
55. Haga clic en OK.
56. Para la segunda matriz, agregue todos los objetos que pueden colisionar con la caracola y, a continuación, haga clic en OK.
57. Agregue una sentencia turn que haga que la caracola gire a la derecha con un valor de 0.3.
58. Agregue un comentario al bloque de código explicando su propósito.
59. Por último, vuelva a myFirstMethod y agregue el código para que se mueva la caracola.

El caracol debe:

- moverse hacia adelante 1,
- la duración debe ser 3
- El estilo de la animación debe comenzar y finalizar con suavidad.



myFirstMethod finalizado debe ser similar al siguiente:



Puede ver que la clase driver (myFirstMethod) casi no contiene ningún código. Solo llama a otros procedimientos para que se encarguen de todo.

Todo esto es lo que debe tener en cuenta en su programación.

60. Guarde el programa.

### Ejercicios opcionales

Intente agregar opciones adicionales a la animación underTheSea.

- Cuando haga clic en el casco de buceo, se inclinará hacia adelante y hacia atrás.
- Cuando haga clic en la estrella de mar roja, haga que se sumerja en el fondo del océano durante 5 segundos.
- Agregue la detección de colisión para el submarino.
- Agregue cualquier otro objeto que desee para que la animación sea más atractiva.

61. Salga de Alice 3.