



ORACLE

Academy

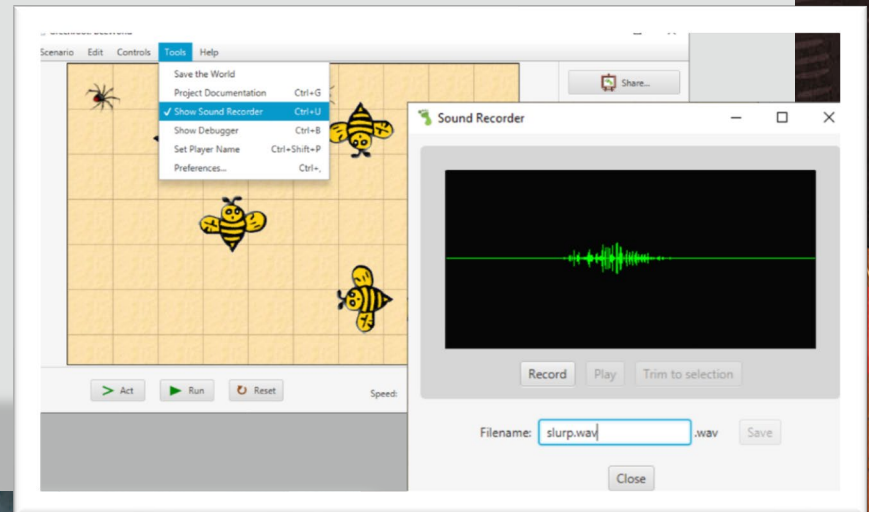


Creación de Programas Java con Greenfoot

Lección 8

Control de teclado y sonido

ORACLE
Academy



Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Escribir sentencias de programación para incluir sonido en un programa
 - Escribir sentencias de programación para incluir movimientos de teclado en un programa
 - Escribir sentencias de programación para incluir la interacción del ratón en un programa
 - Escribir sentencias de programación para recuperar información del usuario

Controles de teclado

- Los juegos están controlados por una persona o jugador de equipo mediante el control remoto o los controles del teclado
- Para que un escenario se comporte como un auténtico juego, programe sentencias que incluyan controles de teclado para que el jugador pueda controlar uno o más objetos en el juego





El Método isKeyDown()

- El método isKeyDown() comprueba si se ha pulsado una tecla en el teclado
 - Ubicado en la clase Greenfoot
 - Es un método static (asociado a una clase)
 - Devuelve el valor true o false
 - Espera un argumento String en la lista de parámetros
 - Se puede utilizar como una condición en una sentencia IF
- Firma del método:

```
public static boolean isKeyDown(String key)
```

Parámetro String(Cadena) en el Método isKeyDown()

- Un String es un trozo de texto (palabra o frase) escrito entre comillas

- Por ejemplo:

- "This is a String"
- "A"
- "name"



left

right

up

down

- El parámetro String en el método isKeyDown() espera el nombre de la tecla que se va a pulsar en el teclado
- Busque el nombre de la tecla observando el teclado
- A veces, el nombre no es evidente (la tecla de cursor a la derecha se denomina "right")

Uso del ejemplo del método isKeyDown()

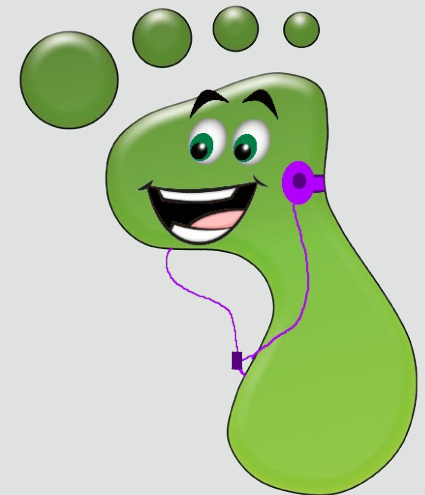
- Este código del método act() utiliza las teclas izquierda y derecha del teclado para permitir que el jugador controle la dirección del objeto "Bee" mientras se mueve

```
/**
 * Act - do whatever the Bee wants to do. This method is called whenever
 * the 'Act' or 'Run' button gets pressed in the environment.
 */
public void act()
{
    move(3);
    if(Greenfoot.isKeyDown("left")){
        turn(-2);
    }else if(Greenfoot.isKeyDown("right")){
        turn(2);
    }//endif
} //end method act
```




Incluir sonido en el Juego

- Los sonidos pueden mejorar el juego
 - Proporcione sonidos con comentarios para el jugador cuando gane, pierda o logre victorias menores durante todo el juego
 - Incluya sonidos de fondo en un juego
- El método `playSound()` se utiliza para reproducir sonidos en un juego
 - El método se encuentra en la clase `Greenfoot`
 - La lista de parámetros espera el nombre de un archivo de sonido (como `String`) a modo de argumento
 - El método no devuelve los datos



Ejemplo de sonido:

- El método se denomina `playSound()` y utiliza una anotación de puntos en el cuerpo del método `catchFly()`
- Siempre que el objeto "Bee" caza una mosca, se emite un sonido

```
/**
 * catchFly - if the Bee touches a fly the fly is removed
 * A sound is played
 */
private void catchFly(){
    if(isTouching(Fly.class)){
        removeTouching(Fly.class);
        Greenfoot.playSound("slurp.wav");
    } //endif
}
```

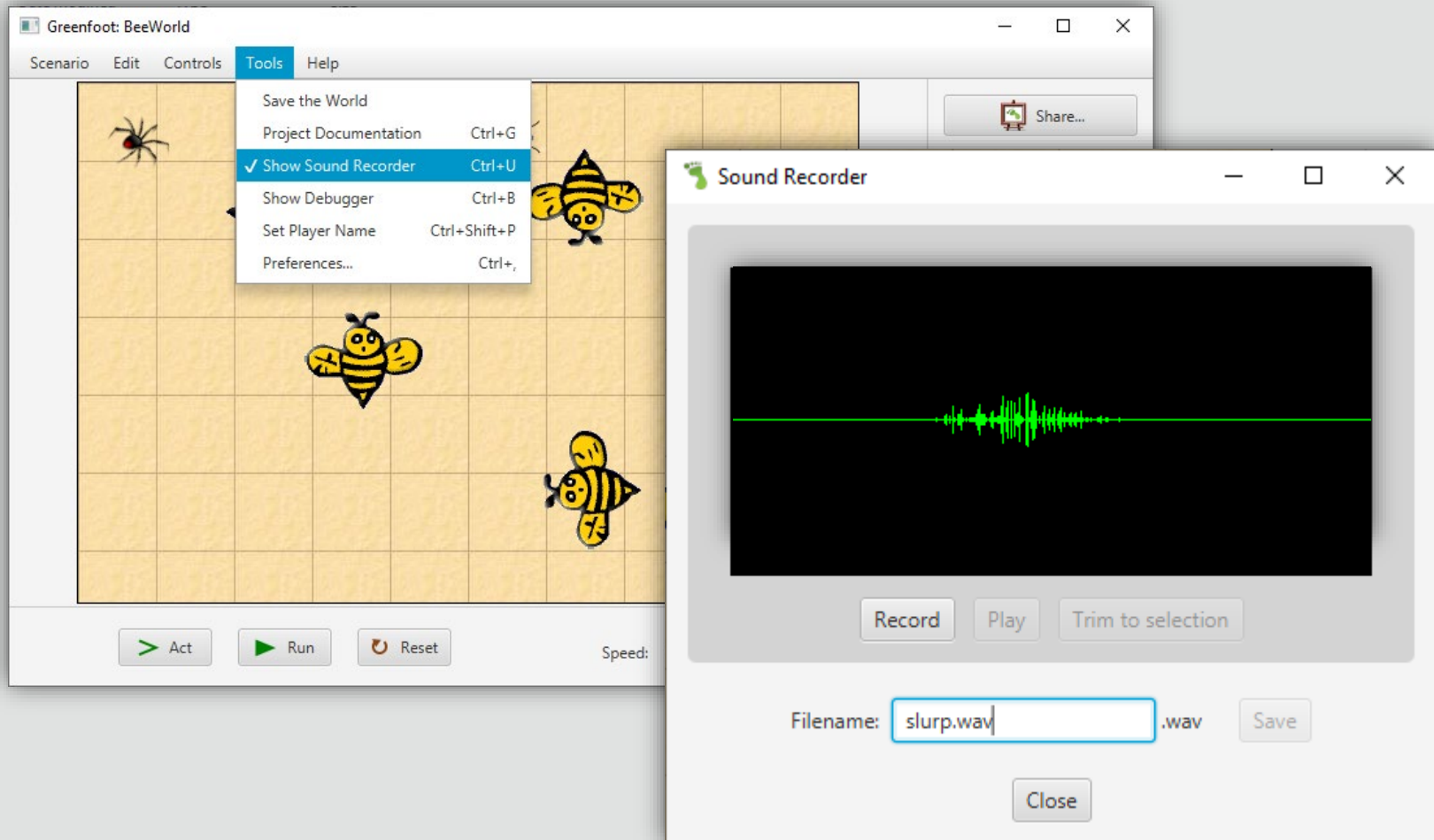


Pasos para grabar sonidos originales

- En el menú Tools del entorno, seleccione Show Sound Recorder
- Pulse Record, a continuación hable al micrófono del ordenador para grabar el sonido
- Pulse Stop Recording cuando haya terminado
- Pulse Play para reproducir de nuevo el sonido
- Si es necesario, vuelva a grabar
- Introduzca un nombre de archivo, haga clic en Save para guardar el archivo en el escenario
- El archivo está ahora preparado para hacer referencia al código



Visualización de la grabadora de sonidos Greenfoot





Con el ratón

- Greenfoot admite varios métodos de entrada además del teclado
- También existe la posibilidad de utilizar controladores, el ratón y otros dispositivos de entrada
- Puede que desee utilizar un ratón en el escenario que está creando, en lugar del teclado
- La clase Greenfoot tiene una serie de métodos que permiten obtener información sobre las acciones del ratón
- Son las siguientes:
 - `getMouseInfo()`, `mouseClicked()`,
`mouseDraggedEnded()`, `mouseDragged`,
`mousePressed()`



Uso del ejemplo del ratón

- El escenario que estamos construyendo no utiliza los controles del ratón, pero veamos un ejemplo
- Si tuviéramos un actor llamado Spider y deseáramos detectar cuando el ratón ha hecho clic en una de sus instancias, haríamos lo siguiente:

```
public void act()  
{  
    if (Greenfoot.mouseClicked(this)) {  
        //do something  
    }  
}
```

Uso del ejemplo del ratón

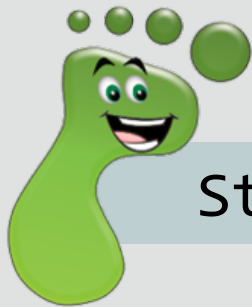
- Si quisiéramos detectar si se ha hecho clic en el ratón en cualquier otro sitio, usaríamos la clase `MouseInfo`
- A continuación, vemos el código que movería la instancia actual a la ubicación en la que se ha hecho clic con el ratón

```
public void act()
{
    MouseInfo mouse = Greenfoot.getMouseInfo();
    if(mouse!=null){
        if (mouse.getButton() == 1) {
            setLocation(mouse.getX(),mouse.getY());
        }//endif
    }//endif
} //end method act
```



Obtención de entrada de teclado del usuario

- Puede que haya un punto en el programa en el que desee obtener la entrada del usuario
- Es decir,
 - solicitando su nombre
 - solicitando una velocidad de arranque, etc.
- A partir de la versión 2.4.1 de Greenfoot esto ya es posible gracias al método de Greenfoot llamado ask()



```
String Greenfoot.ask(String message)
```




Obtención de entrada de teclado del usuario

- El método ask() mostrará el mensaje como una petición de datos y obtendrá el resultado a modo de cadena
- Es decir, pidamos al usuario su nombre y almacenémoslo en el nombre de variable
- Mientras Greenfoot espera su respuesta, el mundo y sus actores se detendrán

```
public void act()  
{  
    String name = Greenfoot.ask("Please input your name: ");  
} //end method act
```

- Se generará

Please input your name:

OK

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
 - Control de teclado
 - Reproducir sonidos
 - Interacción del ratón
 - Ask

Inténtelo

Comportamiento condicional

Esta actividad necesita que se inicie con el archivo de proyecto que se guardó en el tema anterior FrogFly_L7T9.

FrogFly_L7T9.zip

Instrucciones:

Escriba una sentencia if en el método act() que moverá la rana un paso hacia delante cuando se pulse la flecha "arriba" y un paso hacia atrás cuando se pulse la flecha "abajo".

Agregue una sentencia if en el método act() que moverá la rana hacia adelante y a su vez a la derecha 5 grados si se pulsa la tecla de flecha "derecha", y una sentencia if que moverá la rana hacia adelante y a su vez a la izquierda 5 grados si se pulsa la tecla de flecha "izquierda".

Guarde el escenario como FrogFlyL8T10. Compile y ejecute el escenario. Compruebe que la rana se mueve como se espera al pulsar las teclas de flecha.

Inténtelo

Definir un método para reproducir sonidos

Esta actividad necesita que comience con el archivo de proyecto que se guardó en el tema anterior FrogFlyL8T10.

FrogFlyL8T10.zip

Instrucciones:

Abra el editor de códigos de la clase Frog y defina un método denominado `frogSound()` que emite un sonido si se pulsa la tecla "s" del teclado.

Nota: Tendrá que guardar un archivo de sonido en la carpeta de sonidos de su escenario, o grabar un sonido mediante la función Sound Recorder de Greenfoot.

Llame el método `frogSound()` en el método `act()`.

Compile el escenario. Ejecute el escenario y compruebe cómo funciona. Pulse la tecla "s" y compruebe que el sonido se reproduce.

Guarde el escenario como FrogFlyL8T11.

Summary

- En esta lección, debe haber aprendido lo siguiente:
 - Escribir sentencias de programación para incluir sonido en un programa
 - Escribir sentencias de programación para incluir movimientos de teclado en un programa
 - Escribir sentencias de programación para recibir el estado del ratón
 - Escribir sentencias de programación para recuperar una respuesta del usuario



ORACLE

Academy

