

Los siguientes ejercicios muestran cómo realizar animaciones sencillas utilizando hilos.

## Ejercicios hilos y animación

1. Vas a realizar el ejercicio 3 de la hoja anterior con Timer (bola movimiento horizontal) pero ahora implementándolo con hilos. Para ello importa el proyecto *Bola movimiento horizontal AL* y complétalo.

El panel será la tarea Runnable así que deberá implementar el método `run()`.

Cuando se crea el panel habrá que crear el hilo e iniciarlo para que la animación comience la primera vez que se dibuje el panel.

Dentro del método `run()` se implementará el bucle de animación en el que se actualizarán las coordenadas de la bola adecuadamente teniendo en cuenta que:

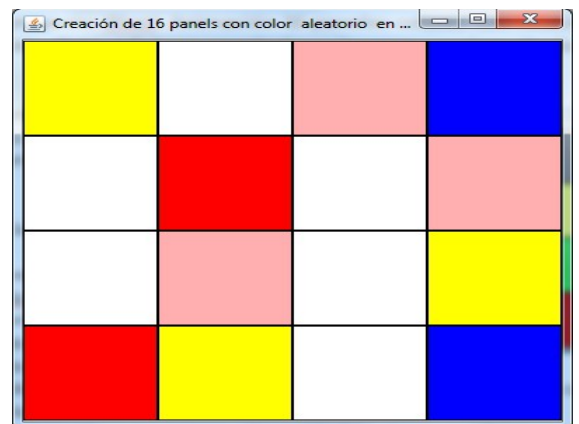
- si va hacia la derecha la coordenada `x` se incrementa en 5. El tope lo marca la línea vertical. Si la bola toca esta línea cambiará de dirección
- si va hacia la izquierda `x` se decrementa en 5. El tope lo marca el borde izquierdo del panel. Si la bola lo toca cambia otra vez de dirección
- para saber la dirección de la bola utiliza una variable booleana *direccion*
- usa los métodos privados `actualizar()` y `pausa()`

2. Importa el proyecto *Panel 16 colores hilos AL*. Descarga y ejecuta (haciendo doble click) el fichero *panel16colores.jar* para ver el resultado del programa que se pide.

Se trata de realizar 16 paneles cada uno con un color aleatorio. Cada panel está definido como una clase interna y será un hilo de ejecución diferente.

Completa en la clase `PanelColor`:

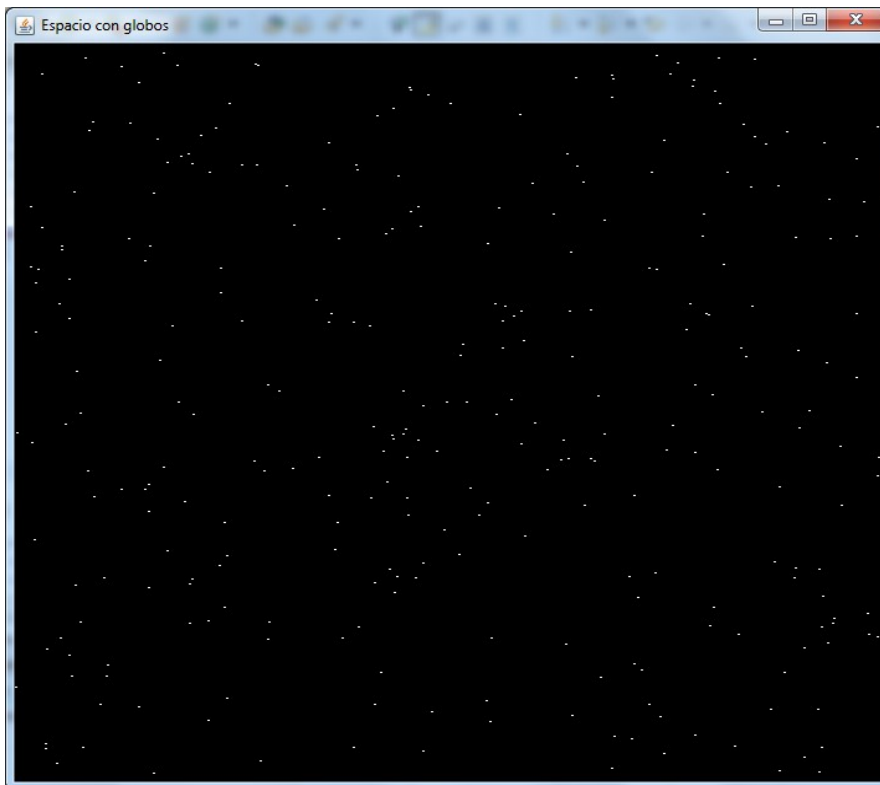
- ✗ el constructor que crea e inicia el hilo e inicializa el atributo *color* con un color aleatorio que se obtiene a través del método `generarColor()`
- ✗ el método `generarColor()` que define un array de objetos `Color` con 6 colores diferentes y devuelve un color aleatorio de entre los 6
- ✗ como el panel será una tarea Runnable habrá que implementar el método `run()`. Aquí se hará el bucle de animación que se ejecutará indefinidamente. En cada ciclo de animación se obtiene un nuevo color para el panel, se dibuja el panel y se duerme el hilo una serie de milisegundos.



3. Importa a Eclipse el proyecto *Animacion espacio con globos hilos AL* y complétalo. *espacioglobos.jar* muestra el resultado que has de obtener. Vamos a hacer el ejercicio en dos pasos:
- primero mostraremos las estrellas y
  - luego crearemos los globos y haremos el efecto de caer sobre el espacio

### Paso 1

Hay que dibujar estrellas en posiciones aleatorias del panel y con ayuda de un hilo dar la impresión de movimiento de las mismas.



La posición de cada estrella va a estar representada por un objeto `Point` (consulta la clase en la API o el aula Moodle).

La clase `Espacio` está completa, no se modifica. Hay que completar la clase `PanelEspacio`. Lee el código incluido y asegúrate de que lo entiendes. Completa:

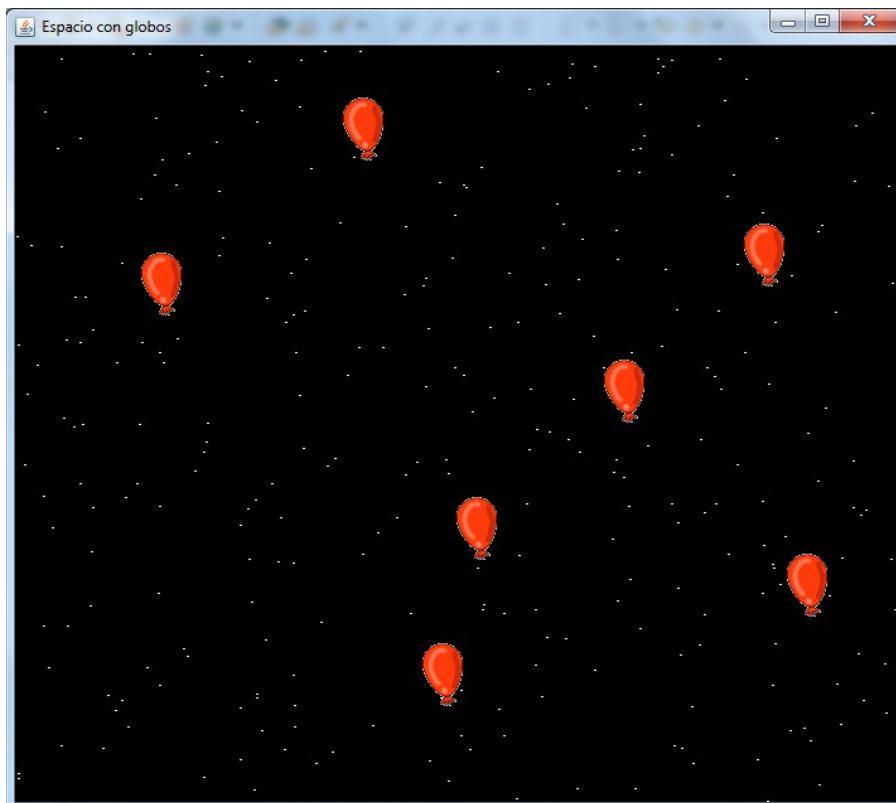
- x el constructor que crea las estrellas
  - primero crea el array y luego genera los puntos con el método privado `generarPuntosEstrellas()` y
  - crea e inicia el hilo

- x el método `generarPuntosEstrellas()` - dando valores a cada elemento del array *estrellas*. Cada estrella está representada por un punto `Point` en el que las coordenadas `x` e `y` del punto serán valores aleatorios: para la `x` un valor desde 0 hasta el ancho del panel - 1 y para la `y` desde 0 hasta el alto del panel - 1
- x el método `paintComponent()` - aquí habrá que dibujar las estrellas de color blanco. Para ello primero codificaremos el método `dibujarEstrellas()`
- x el método `dibujarEstrellas()` - recorriendo el array de puntos y dibujando cada estrella como un óvalo de tamaño 2
- x el panel va a ser la tarea `Runnable`
  - o el método `run()` incluye el bucle de animación que repetidamente genera nuevos puntos para las estrellas, dibuja el panel y duerme el hilo `MILISEGUNDOS`.

Ejecuta el proyecto y verifica que funciona.

## Paso 2

Añadir globos con efecto de caída.



Descarga la clase Globo y añádela al proyecto. Los objetos de esta clase serán los globos que irán cayendo. Define primero como atributo una colección List<Globo> que guardará en todo momento los globos (hasta un máximo de MAX – constante ya definida). Instanciala como colección ArrayList en el constructor.

Completa en la clase PanelEspacio:

- x el método paintComponent() - además de las estrellas cada vez que se dibuje el panel queremos dibujar los globos que haya en la colección. Lo haremos llamando al método dibujarGlobos()
- x el método dibujarGlobos() recorre la colección de globos y los dibuja
- x modifica el método run() que ahora deberá incluir la creación de un nuevo globo y la actualización de las posiciones de los globos además del borrado de aquellos que desaparezcan del panel (los que al ir cayendo desaparecen por la parte de abajo del panel). Lo haremos con ayuda de los métodos crearGlobo() y actualizarGlobos()
- x el método crearGlobo() añade a la colección de globos (si se puede) un nuevo globo en posición aleatoria x (ancho del panel – 1) e y (posición aleatoria un cuarto de la altura del panel – valor negativo – esto hará que la primera vez que se dibuje no se vea)
- x el método actualizarGlobos() recorre los globos y modifica su posición y en 15 (efecto caída del globo) . Si al modificar el valor de y el globo desaparece del panel se borra este globo y se crea uno nuevo)

Ejecuta el proyecto.

Crea un *jar* ejecutable desde Eclipse y ejecútalo.