

## JPlanetas

### NORMAS DEL EXAMEN

1. Descargar el enunciado y los códigos suministrados.
2. Preparar el entorno del examen:
  - a. Consola.
  - b. Explorador de archivos.
  - c. Editor de texto.
  - d. Ventana del navegador con la doc de la API:
    - i. <https://docs.oracle.com/en/java/javase/19/docs/api/index.html>
3. Comenzar la grabación con OBS: **NO empezar el examen hasta no estar seguros de que la grabación se ha iniciado.**
4. Abrir el editor de texto.
  - a. Escribir la palabra POO.
  - b. Copiarla.
  - c. Pegarla dos veces.
5. Pulsar la combinación de teclas Windows + Tab.
6. Comenzar el examen.

El objetivo del examen es desarrollar el escenario sobre el cual se implementará un futuro juego que siga la lógica que se describe a continuación.

### Escenario del juego

El escenario del juego pretende crear unos elementos de fondo que tengan unos movimientos que aportarán dinamismo al juego. Estos elementos básicos a mostrar serán planetas (círculos).

Los planetas vendrán predefinidos en un fichero de texto suministrado (juego.txt), como se muestra a continuación. Observa que no existen separadores de los diferentes campos, por lo que la forma de identificar cada campo vendrá dado por su tipo de dato. En cada línea hay un total de 5 campos, que se corresponde con: coordenada x, relleno, coordenada y, color y tipo de animación.

Por ejemplo, para la línea siguiente:

12true456rojo1

Obtendremos los siguientes campos:

x: 12  
relleno: true  
y: 456  
color: rojo  
animación: 1

**IMPORTANTE:** los campos no vienen dados por el número de caracteres, si no por su tipo. Ver el fichero y los distintos tipos de líneas.

```
12true456rojo1
245false1azul2
1true42verde3
245false111azul2
-1false33rojo2
1false-1rojo2
-1true-1rojo3
```

Fichero juego.txt

Cuando los valores de las coordenadas x o y sean -1, significará que las deberá generar el programa de una forma aleatoria.

Especificaciones generales del juego:

- El fondo del juego será negro.
- Todos los planetas a representar tendrán un **tamaño fijo** de 100 pixels. Este valor podrá modificarse en cada ejecución de una forma sencilla.
- Las **dimensiones** del escenario del juego será de 1000x1000. Al igual que en el caso anterior, podrá modificarse en un futuro.
- Los posibles **colores** de los planetas serán *rojo*, *verde* y *azul*. La asociación de los colores en español a colores Java deberá diseñarse de la forma más escalable posible, pensando que en un futuro podrán aparecer nuevos.
- El último campo, **animación**, podrá tener los valores 1, 2 ó 3. El valor 1 significará que no tiene animación, el valor 2 que tendrá un movimiento descrito por una función cuadrática proporcionada por el profesor y el valor 3 que, además de realizar el movimiento descrito anteriormente, también realizará un efecto de zoom.

### Animación de los planetas

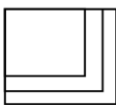
La función cuadrática que determinará **la trayectoria de los planetas** se suministra con el enunciado del examen y está ya implementa en la clase Funcion, mediante el método **public static double[][] generarPuntos(int x, int y, int fps)**. Los argumentos a recibir son la posición del planeta a partir de la cual se generarán los puntos y los FPS. Esto signiifca, que se generarán los puntos necesarios para realizar la animación en un segundo, si los FPS son 60, se generán 60 puntos que representarán las nuevas posiciones del planeta a visualizar en en el segundo siguiente. El método devuelve un array de los nuevos puntos almacenados en una matriz n x m, siendo n el número de puntos creados y m las dos coordenadas del punto, x e y.

### Ayuda:

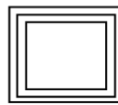
`double[]` representa un array de doubles. `[1.0, 2.0, 3.0, 4.0]`

`double[][]` representa un array de arrays de doubles. `[ [1.0, 2.0], [3.0, 4.0] ]`

El **efecto zoom** ampliará el objeto desde su punto central (no desde su origen de coordenadas) hasta llegar a un 150% de su tamaño original en 1 segundo también. Por ejemplo, para un planeta de 100 pixels de tamaño y una tasa de refresco de 60 FPS, aumentará su tamaño desde 100 pixels a 150 pixels, con incrementos de 0,83 pixels cada actualización (50/60).



Zoom erróneo



Zoom correcto

Recordad que el tamaño de los planetas establecido que ahora es de 100 pixels, puede cambiar en cualquier ejecución futura.

Los planetas siempre deberán estar visibles en la pantalla en su totalidad, por lo que no deberán moverse si se fuera a dar el caso.

### Eventos

Como se está desarrollando un prototipo del escenario del juego, se va a permitir interactuar con él a través de los eventos de teclado. Una vez que se inicie el juego la pantalla estará en negro y no aparecerá ningún planeta.

Cuando se pulsen las siguientes teclas, el juego reaccionará como se describe:

- SPACE: se añade un planeta al escenario de todos los definidos en el fichero. La agregación de los planetas será secuencial. Si en el fichero se definen 5 planetas (5 líneas), se deberá pulsar 5 veces la barra espaciadora para que se muestren todos ellos. Si se pulsara la barra más veces, no pasará nada.
- ENTER: los planetas que aparezcan en el escenario comenzarán a animarse. Tal y como se ha descrito anteriormente, las animaciones durarán aprox. 1 segundo. Se deberá programar el juego para que la duración de la animación sea flexible.
- R: se mostrarán solo los planetas de color rojo del escenario, no del total del fichero.
- G: idem pero de color verde.
- B: idem pero de color azul.
- A: se mostrarán todos los planetas agregados al escenario previamente con el SPACE.
- S: se guardarán todos los objetos del fichero a un nuevo fichero de texto con los valores aleatorios generados y los colores traducidos a Java. Además, los planetas estarán ordenados por la coordenada x. El fichero tendrá extensión CSV y tendrá los campos separados por comas. El nombre del fichero será "juego-timestamp.csv". El timestamp es

un número único determinado por la hora del sistema que proporciona la clase `java.util.Date` mediante su método `getTime()`.

```
1, true, 4, -16711936, 3
1, false, 599, -65536, 2
12, true, 456, -65536, 1
245, false, 111, -16776961, 2
399, true, 399, -65536, 3
800, false, 67, -16776961, 2
991, false, 33, -65536, 2
```

Fichero juego-1683706042344.csv

Herramientas que se podrán utilizar en el examen:

- Editor de texto.
- Compilador.
- JShell: si alguien desea ejecutar código Java de una forma rápida.
- No se podrá consultar ningún sitio web.
- Solo se podrá consultar la API que se encuentra en la siguiente URL:  
<https://docs.oracle.com/en/java/javase/19/docs/api/index.html>
- No se podrá utilizar ningún IDE (Visual Studio Code, IntelliJ, Eclipse, etc.)

Se pide:

- Implementar todas las clases para dar respuesta al enunciado cumpliendo con los principios de orientación a objetos vistos en la asignatura.
- Seguir la estructura de paquetes que desee el alumno.
- Se podrá utilizar cualquier clase del JDK y estilo de programación visto en clase: funcional, por ejemplo.
- Se podrán utilizar import con \*.
- El examen deberá ser grabado con OBS.