Universidad de San Carlos de Guatemala.

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Introducción a la Programación y Computación 1 Sección C

Catedrático: Ing. Moisés Velásquez

Tutor académico: Carlos Campanero, William Corado

Practica no. 3

Manual Técnico

Nombre: Diego Enrique Arriaga Meléndez

Canet: 202003892

Fecha: 2/04/2021

El presente documento que va dirigido a un público con conocimientos técnicos sobre el área de programación. En este se encuentran los recursos utilizados para la realización del programa, entre estos recursos se encuentran el IDE (Integrated Development Environment) que se utilizó, el lenguaje en el que se programó y el equipo utilizado para desarrollar el programa.

Objetivos del Programa

**GENERALES:**

Otorgar al usuario un juego entretenido y fácil de entender para que cualquier persona pueda jugarlo.

**ESPECIFICOS:**

* Crear un juego al cual se pueda jugar en los momentos de aburrimiento.
* Diseñar un juego accesible para cualquier persona sin importar su edad.
* Diseñar una interfaz intuitiva, para que el usuario pueda moverse fácilmente en ella.

Índice

Especificaciones técnicas...............................................................................……………………1

Hardware utilizado……………………………………………………………………………….1

Software utilizado………………………………………………………………………………..1

Listado de clases creadas………………………………………………………………………………2

Objetos globales………………………………………………………………………………………2-4

Configuracion.java…………………………………………………………………………….2-3

Puntuaciones.java…………………………………………………………………………….3-4

Clase Serializable……………………………………………………………………………………..4-6

Clase Menu\_Principal.java…………………………………………………………………………..6-7

Clase Redireccionar.java……………………………………………………………………………….7

Clase Panel\_Configuracion.java…………………………………………………………………...8-10

Clase ListadoPuntajes.java……………………………………………………………………….10-12

Clase MovimientoNave.java…………………………………………………………………………..12

Movimiento de los Extraterrestre…………………………………………………………………13-14

Clase Juego.java…………………………………………………………………………………...14-26

Subclase Apracion\_Objetos………………………………………………………………15-18

Subclase Reloj……………………………………………………………………………...18-19

Subclase DisparoN…………………………………………………………………………19-20

Subclase Actu………………………………………………………………………………21-24

Acciones de Teclado……………………………………………………………………….25-26

Diagrama de Flujo……………………………………………………………………………………...27

Especificaciones Técnicas

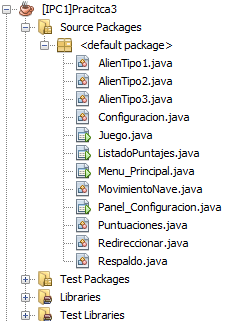
Hardware utilizado

* Maquina: Computadora Portátil HP 14-dq1005la.
* Procesador: Intel™ Core™ i7-1065G7 CPU @ 1.30GHz 1.50 GHz.
* RAM instalada: 8.00 GB (7.70 GB utilizable).
* Unidad de estado sólido de 256 GB Intel® + Memoria de 16 GB Intel® OptaneTM.
* Monitor LED de alta definición, diagonal 14,0 pulgadas.

Software utilizado

* Sistema perative: Windows 10 Home Single Language.
* Versión: 20H2.
* Lenguaje de Programación e IDE: Java.
* IDE: Neatbeans 8.2
* JDK: jdk1.8.0\_111.

Listado de clases creadas.



Antes de explicar el código hay que aclarar que para diseñar los *JFrames* se utilizo la función *drag and drop* de *NetBeans*. Por lo que hay código que se genera automáticamente.

Objetos globales.

Existen dos clases tipo java que contienen objetos que son llamados por otras clases, estas clases son **Configuracion.java** y **Puntuaciones.java**.

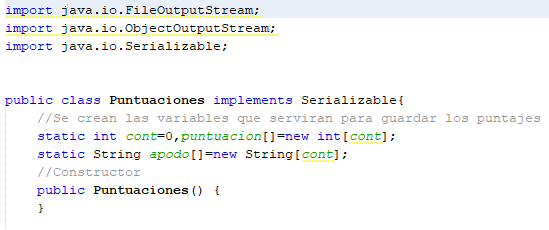
**Configuracion.java**

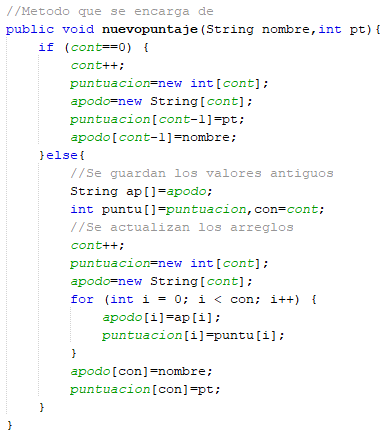
En esta clase se encuentran diferentes variable las cuales funcionan como indicadores de las condiciones con las que se iniciaran un nuevo juego.

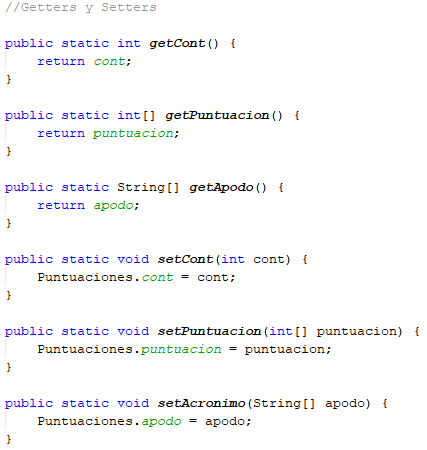


**Puntuaciones.java**

Esta clase posee un par de arreglos, uno que almacena las puntuaciones conseguidas en cada juego y el nombre del jugador que las consiguió. También posee un método que se encarga de almacenar una nueva puntuación al terminar cada partida.

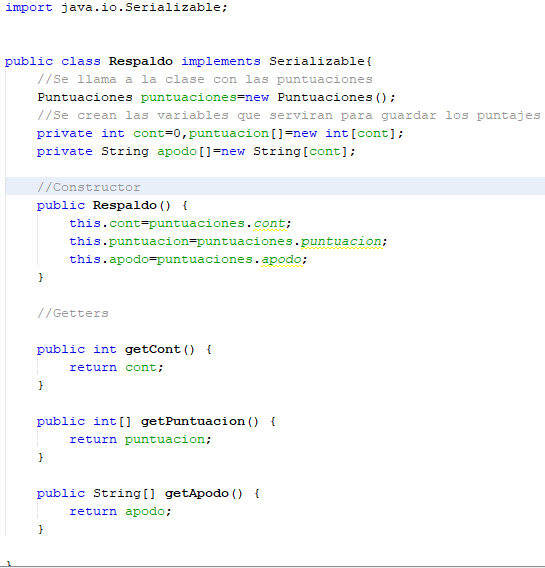






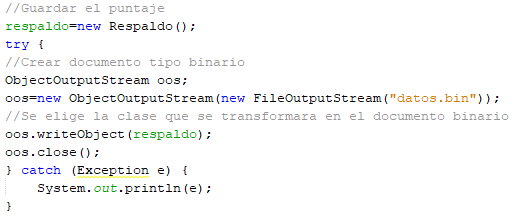
Clase Serializable

Una de las funciones principales del programa es guardar los puntajes conseguidos en cada partida y que estos se guarden incluso después de cerrar el juego. Por lo mismo hay una clase que se encarga de almacenar la información que posea la clase **Puntuacione.java** en variables privadas y no *static*, la clase encargada de esto es **Respaldo.java**.



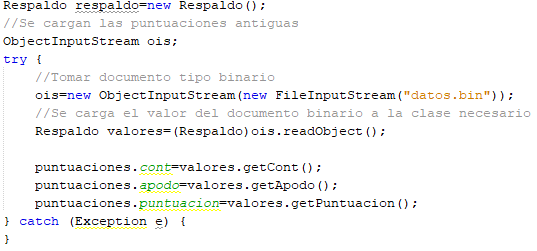
El constructor se encarga de que cada vez que se llame a la clase esta reciba de manera automática la información de la clase **Puntuaciones.java**.

El método encargado de guardar la información en un archivo binario es el siguiente:



Este método se encuentra únicamente en la clase **Juego.java**.

El método encargado de cargar el archivo cada vez que se abre el programa es:



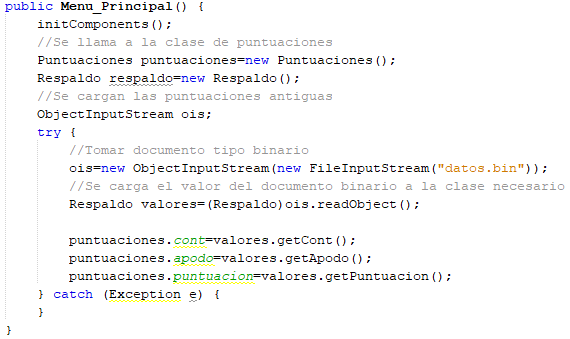
Este método se encuentra únicamente en la clase **Menu\_Principal.java**.

Clase **Menu\_Principal.java**

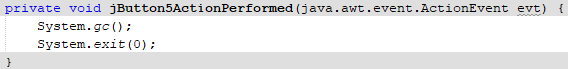
Esta es la clase principal del programa, y genera el menú de inicio del juego



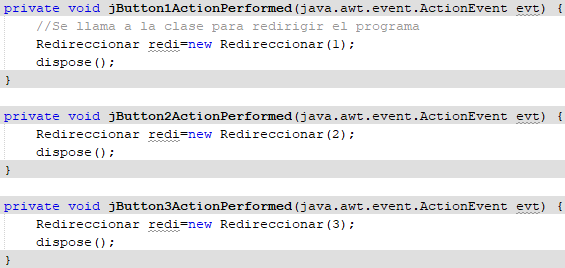
La primera parte del código que se creo en la clase es la encargada de cargar el archivo binario en el programa.



El siguiente método es el que sirve al botón *Salir* para cerrar el juego en la ventana principal.

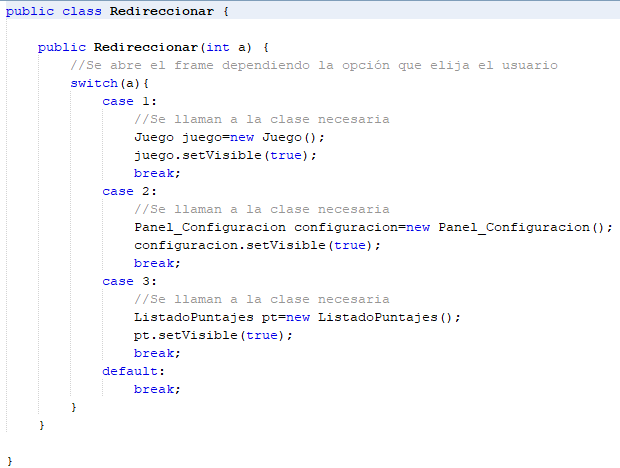


Los siguientes métodos de la clase se encarga de mandar información a la clase **Redireccionar.java** para que esta se encargue de abrir la ventana respectiva al botón que presiono el botón y cerrar la ventana actual.



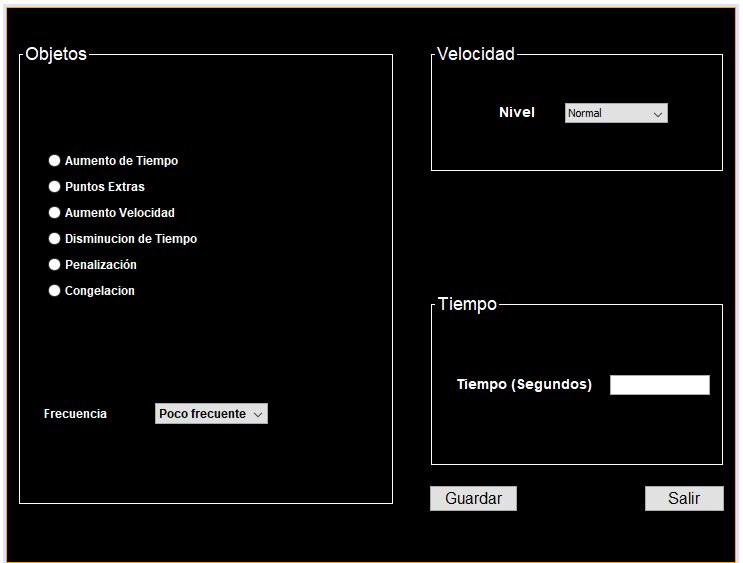
Clase **Redireccionar.java**

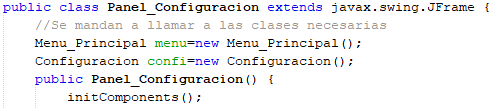
Esta es la clase que se encarga de abrir la ventana necesaria.

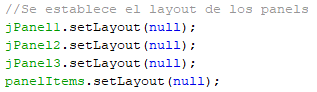


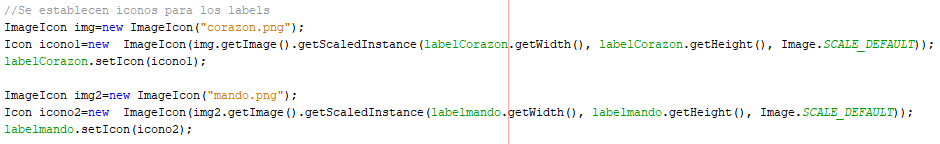
Clase **Panel\_Configuracion.java**

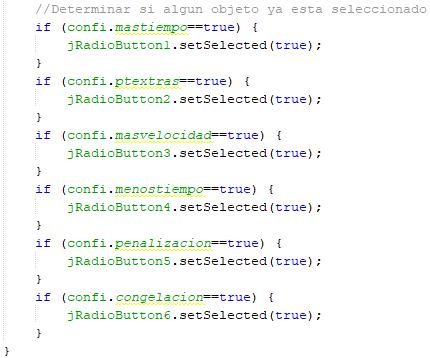
La siguiente clase sirve para que el jugador pueda modificar las características del juego.



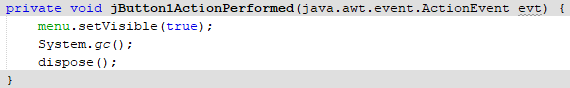




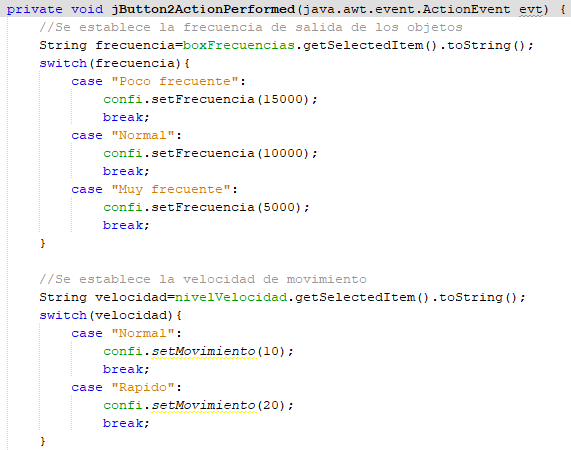




Botón para cerrar la ventana.



Botón para guardar los cambios.

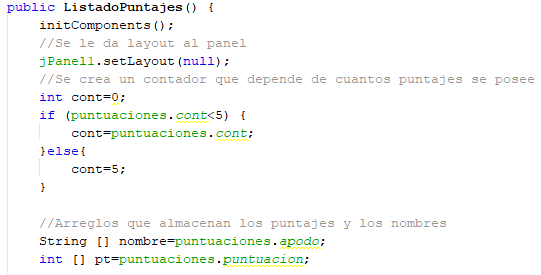


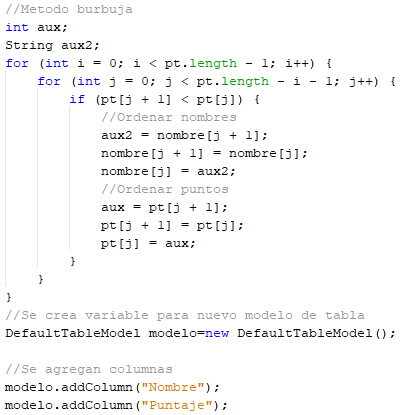


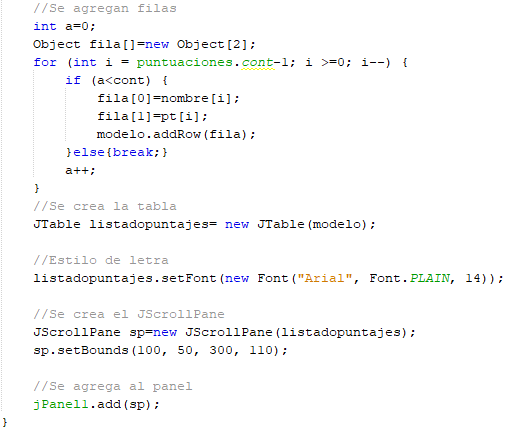
Clase **ListadoPuntajes.java**

La siguiente se encarga de acomodar los arreglos de puntajes y nombres en orden para poder mostrar los 5 mejores puntajes.

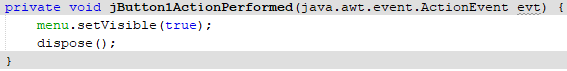






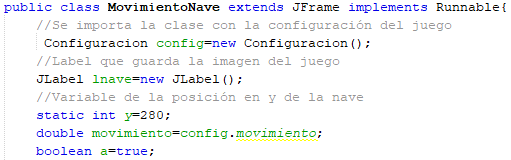


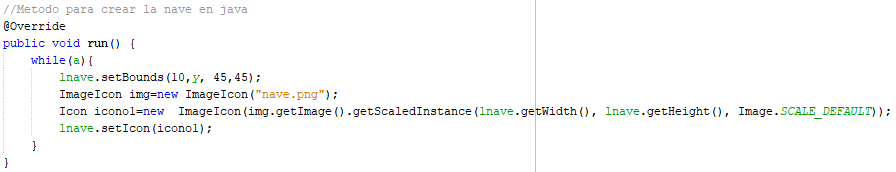
Botón para cerrar la ventana y regresar a la ventana principal.



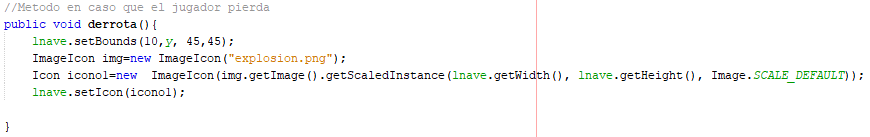
Clase **MovimientoNave.java**

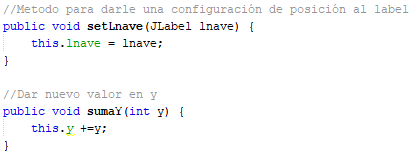
La siguiente clase sirve como hilo que se encarga del movimiento de la nave en el juego.





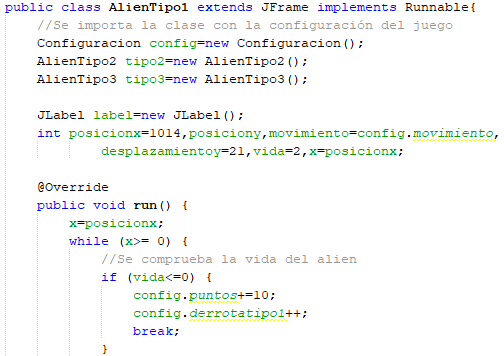
Método que se llama cuando se pierde la partida porque se acaba el tiempo o porque los enemigos alcanzaron el extremo izquierdo de la pantalla.

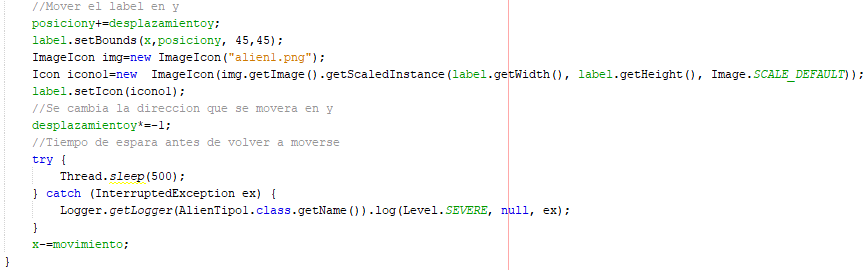


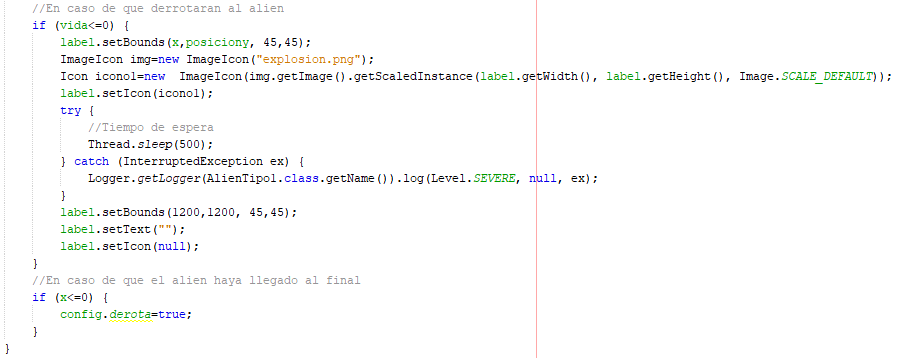


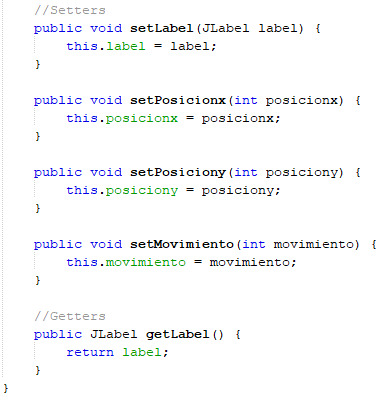
Movimiento de los Extraterrestres.

En el juego hay tres tipos de enemigos, el movimiento de cada uno se maneja en una clase distinta pero cada una es igual, la razón de esto es para poder diferenciarlo al momento de determinar una colisión entre el disparo de la nave y un enemigo, las clases que se encargan de los enemigos son **AlienTipo1.java**, **AlienTipo2.java** y **AlienTipo3.java**.



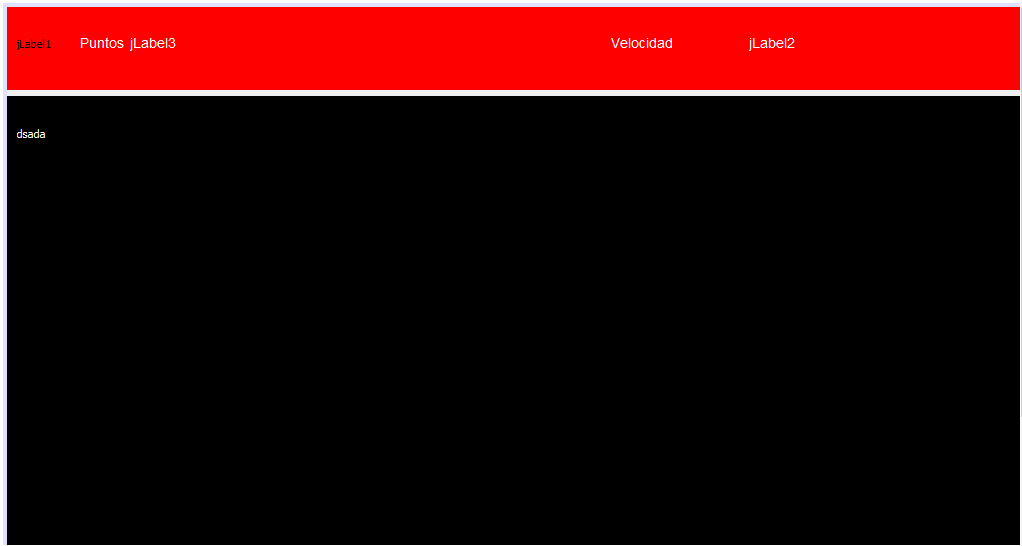




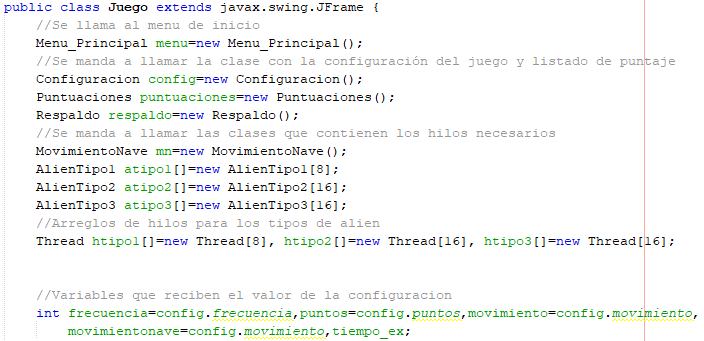


Clase **Juego.java**.

Esta clase es la que se encarga de generar la partida.

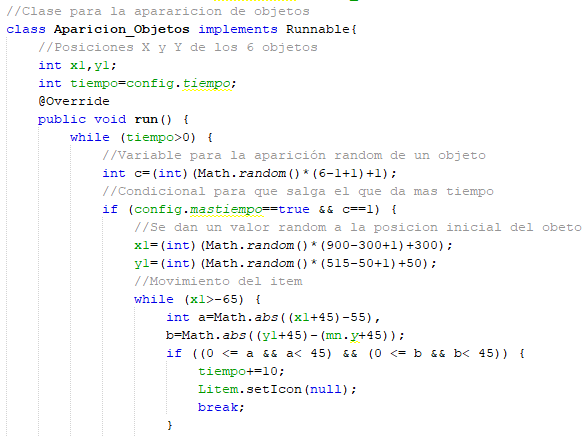


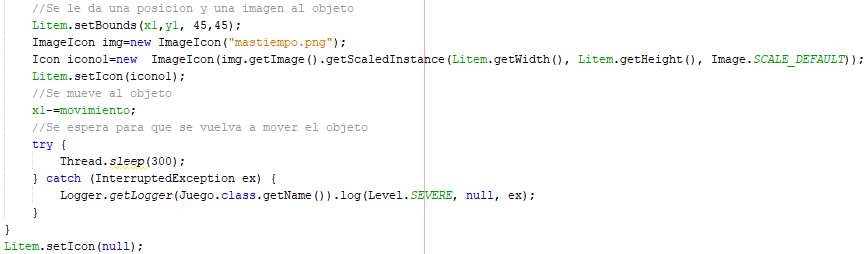
Esta clase posee varias subclases que servirán como hilos de diferentes acciones dentro del juego.

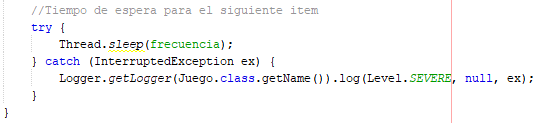
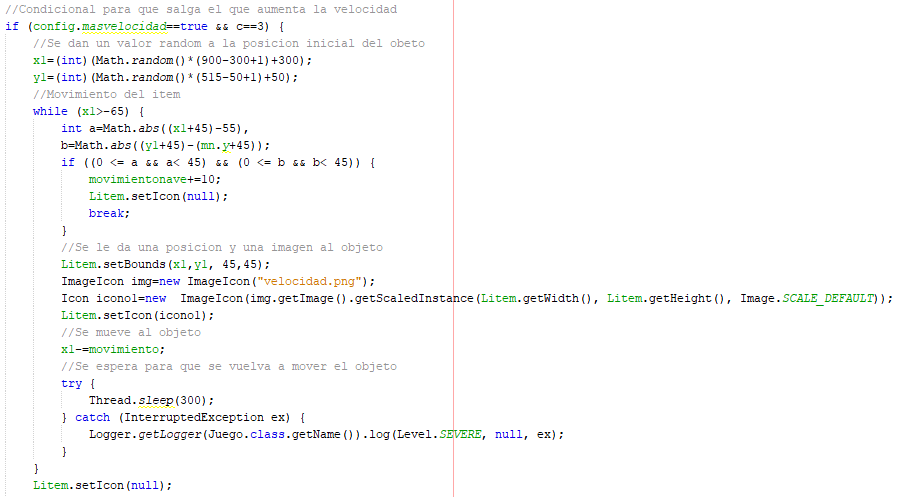
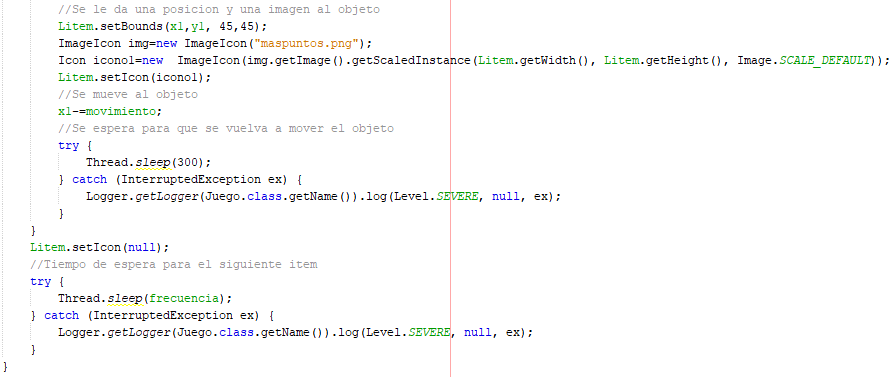
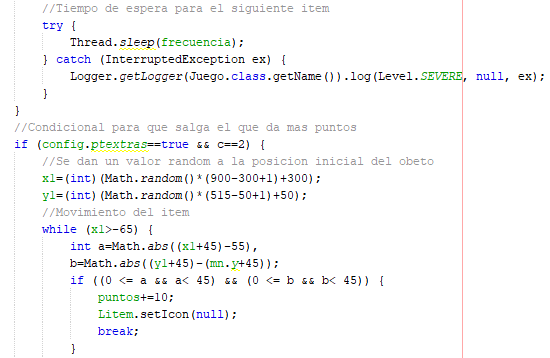


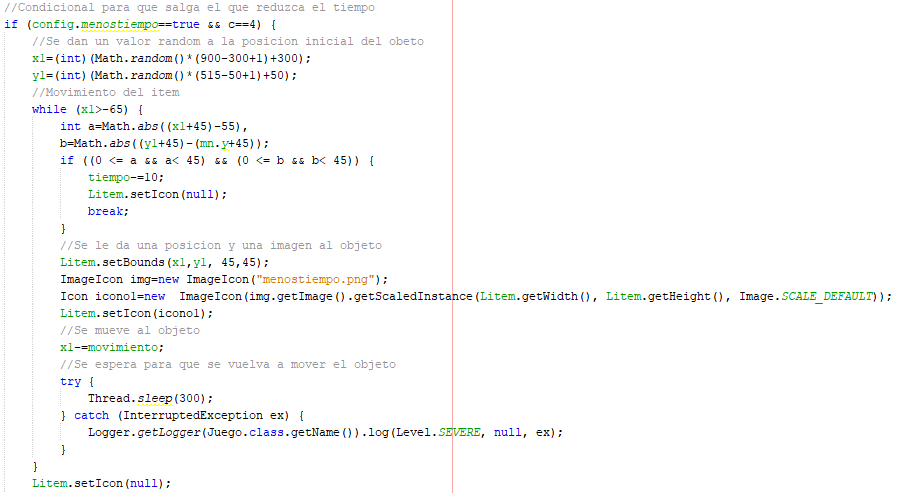
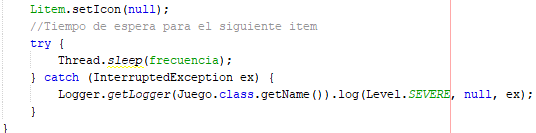
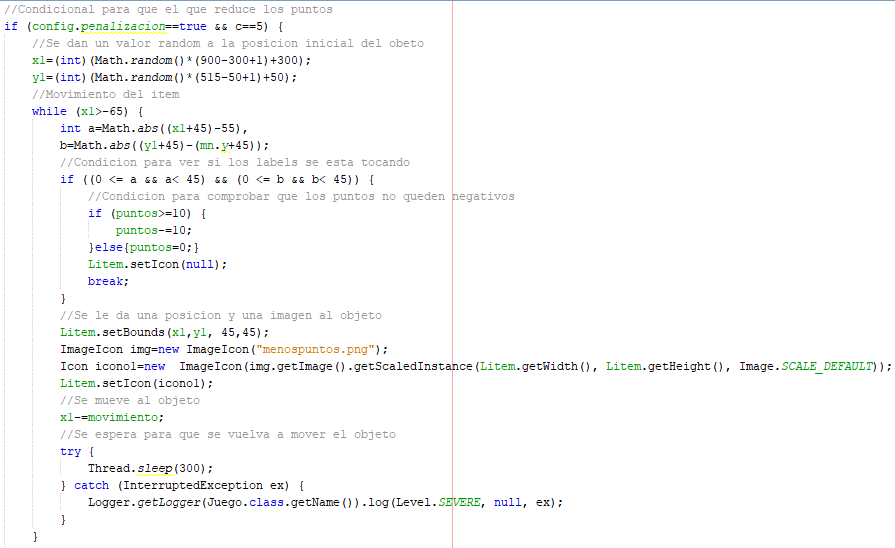
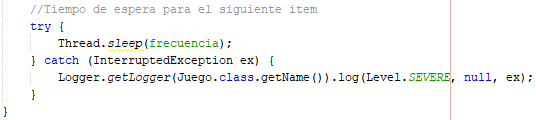
**Subclase Aparcion\_Objetos**

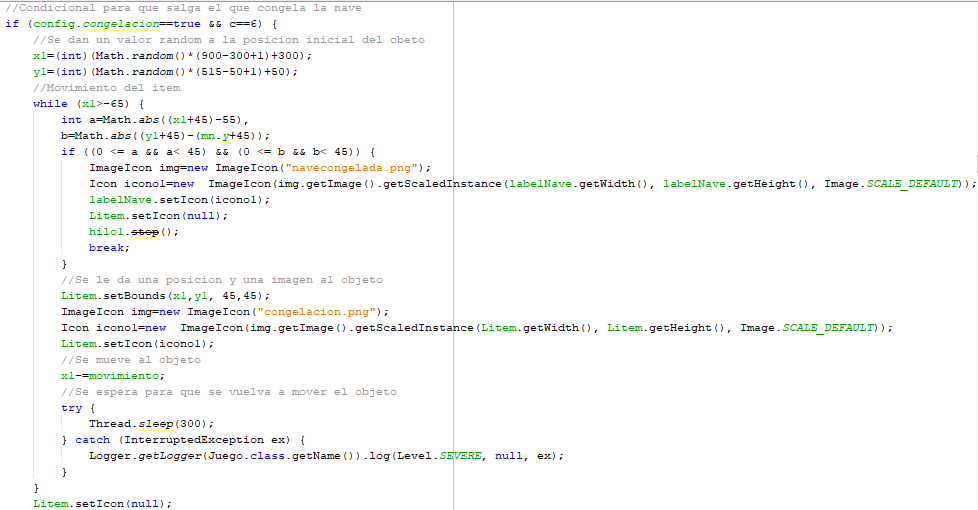
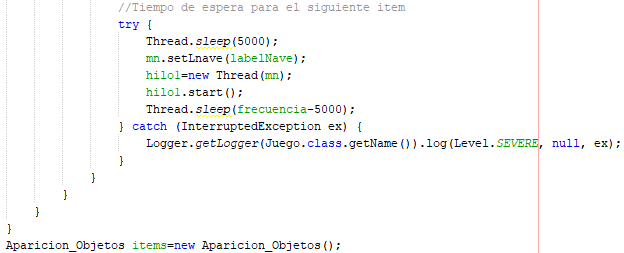
Esta subclase se encarga de que en la pantalla aparezcan los objetos que puede tocar el jugador, y la acción que realizan cuando el jugador lo toca.







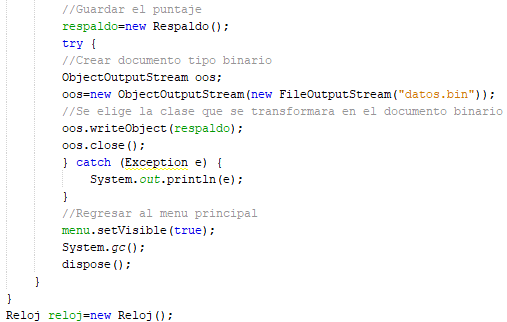
 

**Subclase Reloj**

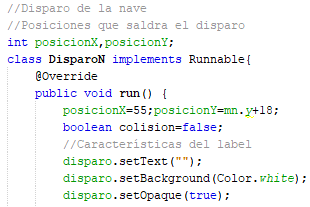
Esta subclase sirve como el hilo que se encarga de llevar control del tiempo de juego y actualizar el marcador en el encabezado.

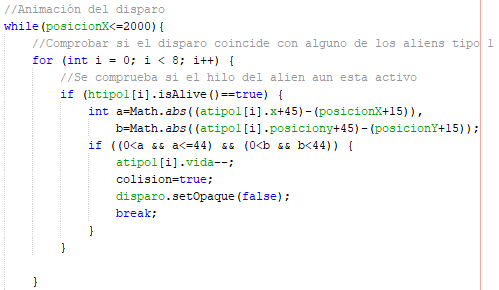
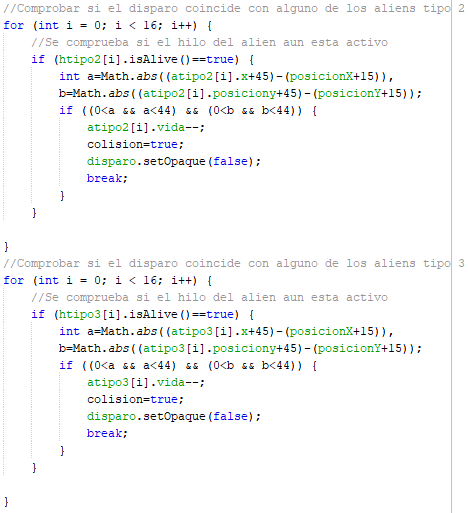
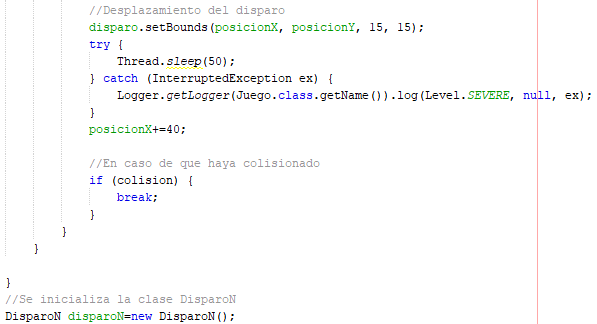


**Subclase DisparoN**

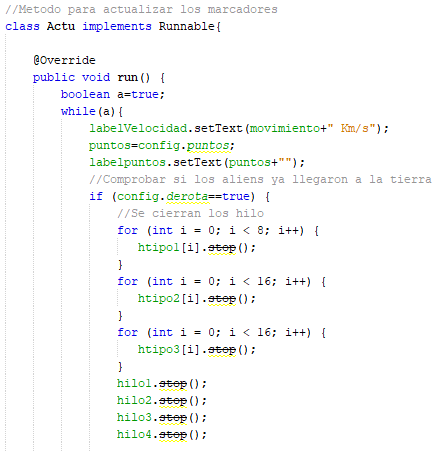
Esta subclase sirve como hilo que se encarga del movimiento del disparo de la nave y la colision de este con algún enemigo, solo en caso que el hilo del mismo este corriendo.

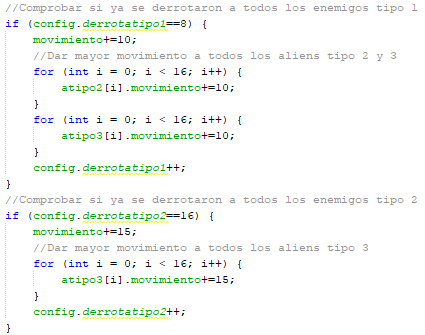


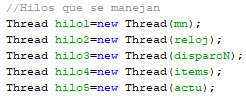
**Subclase Actu**

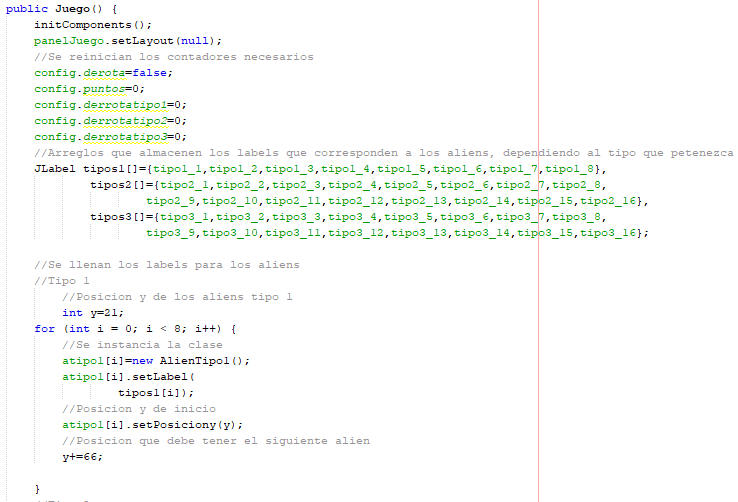
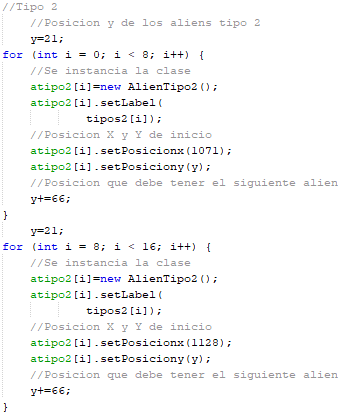
Esta subclase es un hilo que constantemente actualiza los datos que muestra.

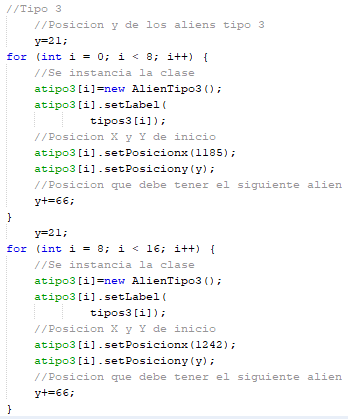


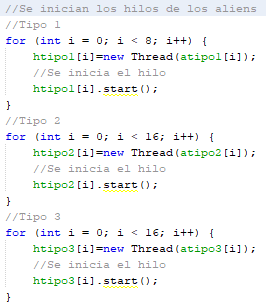


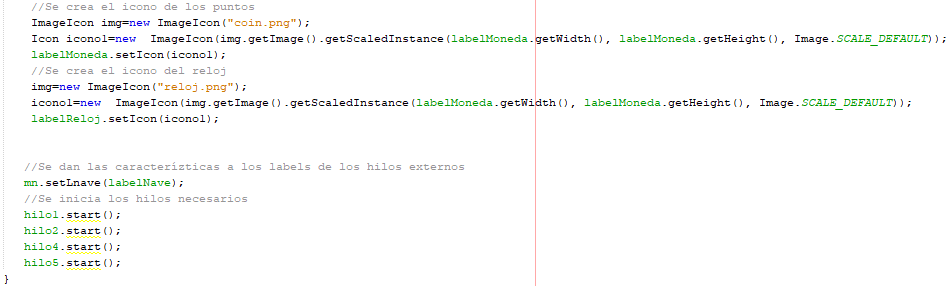




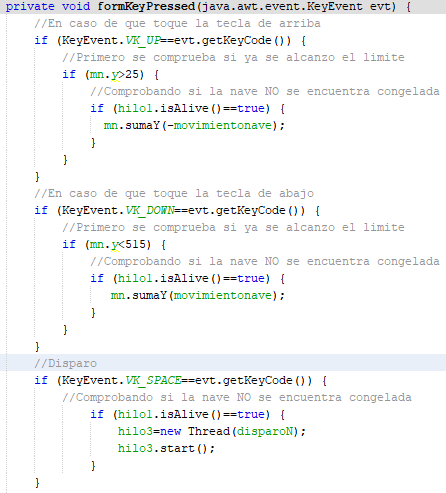


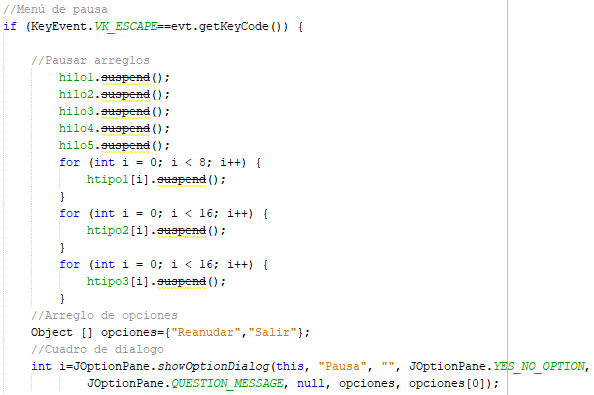


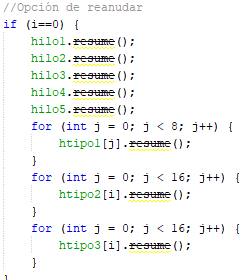


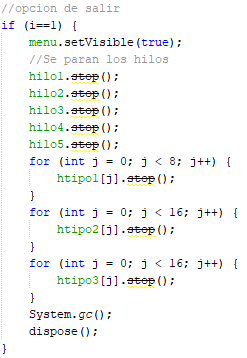
**Acciones de Teclado**

Al frame se le agrego un evento que se activa siempre que el usuario toca una tecla en su teclado.









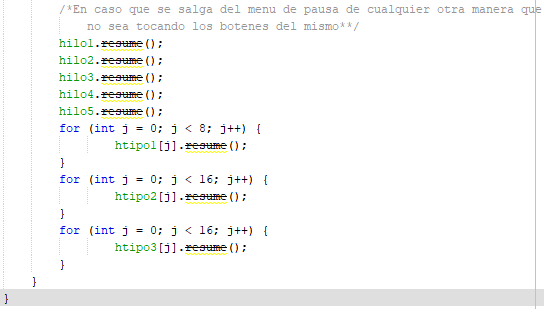


Diagrama de Flujo

