|  |  |  |
| --- | --- | --- |
| Descripción: Escudo UGR | E.T.S. Ingeniería Informática y Telecomunicación  Universidad de Granada  **Ciencias de la computación e i.a** |  |

Práctica 8

## juego del

## Comecocos

|  |
| --- |
| **ALUMNOS:** |
| Miguel Casares Robles  D.N.I:76660965-X  Correo electrónico:miguelo277@correo.ugr.es  Alejandro Vílchez Mancilla  D.N.I:76660965-X  Correo electrónico:tupac21@correo.ugr.es |
|  |

# índice de contenidos

1. Diagrama UML
2. Solución al programa:

2.1. Clase Pac.

2.2. Clase MuevePac.

2.3. Clase Fantasma.

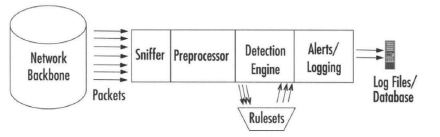
2.4. Clase MueveFantasma.

2.5. Clase Tablero.

2.6. Clase Reloj.

1. Mejoras Implementadas:

# Diagrama UML



**Figura 1. Arquitectura de Snort.**

# Solución al programa

Para analizar la solución de este programa empezaremos por las clases más sencillas y avanzaremos en progresión hasta analizar todas las clases. En nuestro caso, empezaremos por las clases **Pac**, **Fantasma** y **Tablero.**

**2.1. Clase Pac.**

La clase Pac contiene los métodos destinados a controlar los parámetros para controlar el comecocos. Como datos miembro contiene cuatro variables que controlan la posición del elemento. Así como el número de vidas restantes, la interacción en la que se encuentra, las coordenadas iniciales, el estado y la puntuación que se recoge.

Los métodos principales de esta clase se encargan de modificar los valores de estado a cuanto el comecocos se refiere. (Comecocos como figura principal que controlamos e interactúa con el resto del programa). Son métodos los siguientes: **getPosicionX(), getPosicionY(), getDireccion(), getPuntuacion(), getVidas(), revivirPac(), quitarVidas(), getEstado(), setDireccion(), cambiarDireccion(), sumarPuntuacion().**

El método **revivirPac(),** devuelve el elemento comecocos a la posición e instantes en el que se encontraba cuando su indicador de Vidas llegó a cero.

El método **quitarVidas()**, descuenta el número de vidas cada vez que el elemento comecocos choca contra un fantasma, el momento que las vidas llegan a cero, refiere a **revivirPac().**

Otro de los métodos utilizados en la clase es **cambiarDireccion()**, que modifica los parámetros de origen para mover el elemento comecocos.

Por último, el método **sumarPuntuacion()** recoge los valores que son utilizados para obtener una puntuación y los acumula.

**2.2. Clase MuevePac.**

En la clase MuevePac hacemos uso de métodos utilizados en la práctica 7 de esta asignatura, el Tetris, en el que se utilizan hebras para el movimiento del elemento comecocos.

La clase implementa la interfaz Runnable, con datos miembros como "continuar" y "suspendflag" que controlan la hebra, un indicador del frame en el que nos encontramos y una variable para indicar el nivel.

El método **MuevePac()**, que es un constructor se encargará de inicializar el frame e ir actualizando el nivel, el retardo y la hebra que mueve el elemento comecocos.

El método **Run()** incluye las sentencias necesarias para que el comecocos se mueva, reinicie su posición y todo lo referente al movimiento, en este método, a su vez, se actualiza la puntuación del juego. Se debe gestionar la excepción de la interrupción del juego cuando el elemento comecocos sea comido por un fantasma.

Otros tipos de métodos se encargar de controlar el funcionamiento de la hebra, como el método **suspender(), abrirPuerta()**, que controla cuando se inicia la salida de fantasmas, **reanudar(),parar(), getParado(),** que muestra si la hebra esta parada.

El método **actualizaRetardo()** controla la velocidad con el que se mueven las figuras conforme el nivel.

El método **funcionesVarias()** muestra en el tablero los elementos Vida y Puntuación.

**2.3. Clase Fantasma.**

La clase Fantasma funciona prácticamente igual que la clase Pac en el que los datos miembros de la clase hacen referencia a la posición de cada uno de los fantamas, los cuales se diferencian por un color, y también se hacen referencia al estado de los mismos.

En esta clase se encuentra el constructor **distanciaPac()**, que define el movimiento de los fantasmas con respecto su distancia al elemento comecocos.

El método **Fantasma()** define el color de cada fantasma y su movimiento y dirección mediante una hebra.

En esta misma clase se encuentran los métodos **setDireccion(), getDireccion(), getColor(), getPosicion(), setPosicion(), getEstado(), setEstado().** En los cuales se hacen referencia al movimiento, estado, color y posición de los fantasmas.

**2.4. Clase MueveFantasma.**

La clase MueveFantasma

**2.5. Clase Tablero.**

La clase Tablero ha sido generada conforme los pasos descritos en el guión 7.

Los constructores **Tablero()** se encargan de generar un nuevo formulario con el esquema del tablero, son métodos que se generan automáticamente. Se utilizan el tablero como plantilla y después se utiliza la clase FormaTablero, para darle un formato a la plantilla.

Por otra parte, el método **initTablero()** es un método que se encarga de generar todos los elementos que forman parte de nuestro tablero, mientras que método **imprimirTablero(),** se encarga de mostrar en el panel los elementos.

En esta misma clase, hemos incluido los métodos **seChoca(),** que hacen referencia a que si elemento comecocos o los fantasmas chocasen en alguna parte del tablero delimitada, cambien de dirección.

**2.6. Clase Reloj.**

En la clase Reloj es una hebra que controla el tiempo de los elementos en el que transcurrido un tiempo si el elemento comecocos no completa el tablero a este se le resta un vida y se resetean los fantasmas. Los datos miembro de esta clase indica el frame del elemento comecocos y un contador desde el cual se empieza a contar cuanto tiempo queda hasta restar una vida.

En esta clase también se incluyen los métodos **run(), getParado(), parar(), reanudar(), suspender(),** que informan del estado y paran o reanudan el juego. El método **run()** es la hebra que se utiliza para el contador que funcione mientras el elemento comecocos está moviendose, y si le comecocos no termina el recorrido antes de que termine el tiempo, se le resta una vida.

# Mejoras Implementadas

Las mejores implementadas en este programa comienzan con::

Mejora 1: Se han incluido los puntos pequeños y grandes en el tablero de forma que:

* Los puntos pequeños suman 10 puntos y los grandes 50, esto ha sido recogido en el método **sumarPuntuacion().**
* Los puntos obtenidos son mostrados en pantalla con la puntuación hasta el momento.
* El juego acaba cuando el comecocos come todos los puntos del laberinto inicial o se queda sin vidas.

Mejora 2: Se ha dado formato a la forma detallada mostrada. Para ello damos formato al tablero inicial en la clase **FormaTablero.**

Mejora 3: Se ha incluido una forma de reanudar y pausar el juego en un botón menú en la interfaz, así como inicializar una nueva partida. Añadidos en la hebra de la clase **Reloj**.

Mejora 5: Se han utilizado diferentes aspecto para el comecocos según sea su dirección de movimiento.

Mejora 6: Hemos hecho que aparezcan los cuatro fantasmas. Estos irán saliendo poco de la caja y perseguirán al elemento comecocos. Tanto el movimiento, la vulnerabilidad y que persigan al elemento comecocos es implementado en la hebra de la clase **mueveFantasma**.

Mejora 13: Se ha asociado un tiempo máximo para completar un nivel, el tiempo reservado se reducirá según nivel, esto ha sido implementado en una clase **Reloj**, con un contador llamado **TIEMPOINICIO.**