

Universidad de Alcalá - Escuela Politécnica  
Departamento de Ciencias de la Computación

# PROGRAMACIÓN ORIENTADA A OBJETOS (Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería de Computadores, Ingeniería Informática y  
Sistemas de Información

## EJERCICIO 2

### 1. Juego Plantas vs Zombis

En esta práctica se pretende desarrollar el juego *Plantas contra Zombis*, donde los zombis van apareciendo y el jugador va disponiendo diferentes plantas con distintas características de ataque o defensa para detener a la horda de zombis que intenta devorar los cerebros de los residentes.

A nivel visual el juego se presenta como una cuadrícula en la que el jugador coloca plantas con diferentes características; los zombis aparecen de un lado del tablero y el jugador tiene que resistir el ataque de los zombis, destruyéndolos todos. En el caso de que uno de los zombis supere la defensa y llegue hasta el final del tablero, el usuario pierde la partida.

### 2. Descripción de la práctica

Se va a considerar que el juego consta de un tablero de  $F \times C$  casillas (F filas por C columnas) las cuales podrán ser definidas por el usuario al crear el juego.

En esta versión del juego se tiene que implementar el juego mediante la interfaz consola.



Fig. 1. Tablero del juego

Para ello se pondrá en la línea de comandos:

$N <n^\circ \text{filas}> <n^\circ \text{columnas}> <\text{dificultad}>$

En cada casilla se puede colocar una sola planta o puede haber un zombi. Los zombis van avanzando de derecha a izquierda y ganan la partida cuando llegan al final del tablero. Por su parte, el jugador tiene que matar a un número de zombis determinado para ganar. El juego finalizará si no quedan más zombis por salir y no hay ningún enemigo en el tablero, en ese caso ganarán las plantas o si algún zombi consigue llegar al extremo izquierdo del tablero, en ese caso ganarán los zombis. También podemos interrumpir el juego si en la línea de comandos introducimos el comando "S".

Inicialmente el jugador comienza con 50 soles y el tablero aparecerá vacío (fig. 2).

```
(Teclear ayuda para lista de comandos. <Enter> para terminar el turno.)  
> N 7 7 MEDIA  
Comenzando la partida.  
  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
|       | |       | |       | |       | |       | |       | |  
|-----|  
Tienes 50 soles y es el turno 1  
(Teclear ayuda para lista de comandos. <Enter> para terminar el turno.)  
>
```

Fig. 2. Tablero inicial

Aunque en el juego original se consideran diferentes tipos de plantas, en este caso, únicamente se van a considerar dos tipos de plantas: Girasoles (proporcionan sol) y LanzaGuisantes (disparan guisantes a los zombis).

Además de funciones diferentes, cada planta tiene un coste, que se mide en unidades de sol o soles. El usuario puede ir acumulando soles para plantar nuevas plantas. Los soles se acumulan gracias a un tipo de plantas especiales: los girasoles.

También se pueden considerar diferentes tipos de Zombis, aunque en este caso particular únicamente implementaremos uno: **ZombieComun**: (Zombi de jardín común).

En cada turno de juego se realizan secuencialmente las siguientes acciones:

- 1) **Turno**: Se introducen por línea de comandos, el comando deseado para realizar la acción a realizar. Por ejemplo: añadir una planta en una casilla específica. El usuario puede no hacer nada en un turno, pulsando *Enter* para pasar el turno
- 2) **Actualizar Tablero**: Se actualiza el tablero y se realiza la acción por cada personaje que aparece en el tablero. Las LanzaGuisantes disparan guisantes, los zombies avanzan, los girasoles acumulan sol, y se colocan en el tablero nuevos enemigos si es necesario. La colocación de estos será aleatoria y dependerá del nivel.
- 3) **Dibujar Tablero**: Se dibuja el tablero y se muestra la información del juego

## 2.1 Objetos del juego

En esta sección se describe el tipo de objetos que aparecen en el juego y su comportamiento.

### Planta LanzaGuisantes:

- **Comportamiento**: dispara guisantes a los zombis causándoles daño
- **Coste**: 50 soles
- **Resistencia**: 3 puntos de daño
- **Frecuencia**: 1 guisante por ciclo
- **Daño**: Cada guisante proporciona 1 punto de daño
- **Alcance**: Solo dispara recto y hacia adelante

### Planta Girasol:

- **Comportamiento**: Genera soles
- **Coste**: 20 soles
- **Resistencia**: 1 punto de daño
- **Frecuencia**: Genera 10 soles cada 2 ciclos
- **Daño**: 0 (No ejerce daño a los zombis)

### ZombiComun:

- **Comportamiento**: Avanza y come plantas
- **Resistencia**: 5 puntos de daño
- **Daño**: 1 punto de daño
- **Velocidad**: 1 casilla cada 2 ciclos

**Tablero**: formado por un array de un número de filas x número de columnas de celdas que almacenará la situación de la partida.

Como se puede observar en el comportamiento de los objetos, todos los personajes deberán realizar una acción.

La aplicación tiene que hacer uso de las clases:

- *ExcepcionPlanta*, donde se controlará si no hay suficiente dinero, si se desea colocar la planta fuera del tablero, si la casilla está ocupada o si la celda en la que se desea colocar es la última columna.
- *ExcepcionJuego*, donde se controlará si hay una partida en juego y se quiere comenzar otra, si faltan argumentos para crear la partida, si se ha introducido un nivel de dificultad no valido.

*Opcionalmente el alumno podrá hacer uso de otro tipo de excepciones creadas por él.*

A continuación, se describe lo que ocurre en cada parte del bucle del juego.

## 2.2 Funcionamiento del Juego

### 2.2.1 Turno

Las actualizaciones que ocurren en cada turno son:

- Los Girasoles pueden acumular sol.
- Las plantas LanzaGuisantes disparan a los zombis que estén a su alcance.
- Los zombis avanzan (un zombi no puede avanzar si tiene a otro zombi o a una planta delante).
- Si un zombi está en una casilla adyacente a una planta, ejerce daño a la planta.
- Si una planta o zombi llega a 0 de resistencia entonces desaparece del tablero.

El juego finalizará si todos los zombis son destruidos y no hay más zombis por salir o uno de los zombis llega al final de la fila. Cuando el juego termine se debe mostrar quién ha sido el ganador: "Plantas ganan" o "Zombis ganan".

**Las actualizaciones siempre se hacen en el mismo orden: Girasoles, LanzaGuisantes y Zombis. Siempre se actualizan los elementos del juego en orden en el que fueron introducidos.**

### 2.2.2 Actualizar Tablero

En cada ciclo se pintará el estado actual del tablero, así como el ciclo del juego en el que nos encontramos (inicialmente 0), el número de soles acumulados y el número de zombis que quedan por aparecer.

El tablero se pintará por el interfaz consola utilizando caracteres ASCII, como se mostró en la figura 2.

Al lado de cada objeto en el tablero (plantas o zombis) aparece la vida que les queda (entre corchetes). Por ejemplo, en la figura 3, se muestra G(1) y L(3), es decir, Girasol 1 vida y LanzaGuisantes, 3 vidas.

También se muestra el prompt del juego para pedir al usuario la siguiente acción.

```

> L 2 1
|-----|
|G(1) |   |   |   |   |   |   |   |
|-----|
|L(3) |   |   |   |   |   |   |   |
|-----|
|   |   |   |   |   |   |   |   |
|-----|
|   |   |   |   |   |   |   |   |
|-----|
|   |   |   |   |   |   |   |   |
|-----|
|   |   |   |   |   |   |   |   |
|-----|
|   |   |   |   |   |   |   |   |
|-----|
|   |   |   |   |   |   |   |   |
|-----|
Tienes 0 soles y es el turno 5
(Teclear ayuda para lista de comandos. <Enter> para terminar el turno.)
> |

```

Fig. 3. Tablero con partida iniciada

### 2.3 Línea de comandos

Se le preguntará al usuario qué es lo que quiere hacer, a lo que podrá contestar una de las siguientes alternativas:

- **N** *<filas>* *<columnas>* *<Dificultad>*: Nueva partida (Dificultad: BAJA, MEDIA, ALTA, IMPOSIBLE).
- **G** *<fila>* *<columna>*: colocar girasol. Únicamente se podrá añadir un nuevo Girasol por turno y si tiene el número suficiente de soles. No podrá añadir un Girasol en una casilla ocupada por otra planta o por un zombi.
- **L** *<fila>* *<columna>*: colocar LanzaGuisantes. Únicamente se podrá añadir un nuevo LanzaGuisantes por turno y si tiene el número suficiente de soles. No podrá añadir un LanzaGuisantes en una casilla ocupada por otra planta o por un zombi.
- **S**: Salir de la aplicación.
- **<Enter>**: Pasar Turno

- *ayuda*: este comando solicita a la aplicación que muestre la ayuda sobre cómo utilizar los comandos.

#### Observaciones sobre los comandos:

- La aplicación debe permitir comandos escritos en minúscula y mayúscula o mezcla de ambos.
- Si el comando está mal escrito, no existe o no se puede ejecutar, la aplicación mostrará un mensaje de error.
- En el caso de que el usuario ejecute un comando que no cambia el estado del juego, o un comando erróneo, el tablero no se debe repintar.

## 2.4 Niveles de Dificultad

Uno de los parámetros de entrada será el nivel de dificultad. Definiremos cuatro niveles BAJA, MEDIA, ALTA, IMPOSIBLE.

Cada nivel determinará:

- *Número de turnos en los que salen zombis*.
- *Número de zombis en la partida*: será el número de zombis total que aparecen en la partida, dicho parámetro está relacionado con la dificultad del tablero.
- *Turnos iniciales sin zombis*: número de turnos que pasan sin que aparezcan zombis en el tablero. Dicho parámetro está relacionado con la dificultad del tablero

	<b>BAJA</b>	<b>MEDIA</b>	<b>ALTA</b>	<b>IMPOSIBLE</b>
<b>Nº Turnos Reparto Zombis</b>	30	30	30	30
<b>Nº Zombis en la Partida</b>	5	15	25	50
<b>Turnos iniciales sin Zombis</b>	10	7	5	5

Tabla 1: Configuración para cada nivel de dificultad

Cuando aparece un zombi el ordenador lo coloca en una de las filas del tablero aleatoriamente y en la última columna de la matriz.

### 3. Entrega de la Práctica

Se deberá entregar el proyecto NetBeans realizado por cada alumno o grupo (de dos alumnos como máximo). Dicho proyecto deberá estar listo para arrancar la aplicación, para comprobar que se cumplen las especificaciones formuladas. El código deberá estar correctamente documentado y generada la documentación con JavaDoc.

Se comprimirá en un fichero indicando el *primer apellido de cada miembro del grupo y el curso 18\_19*. Se entrega en la sección "Trabajos" de Blackboard en el grupo correspondiente.

*La Fecha tope de entrega será 6 de Mayo de 2019.*