

PROGRAMACIÓN ORIENTADA A OBJETOS

(Laboratorio de Prácticas)

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

EJERCICIO 3

1. Juego Plantas vs Zombis

En esta práctica se pretende realizar una segunda versión del juego Plantas vs Zombis desarrollado en el ejercicio 2. En esta versión y como novedades más importantes estará la conversión en aplicación gráfica y la posibilidad de realizar un torneo en el que juegan contra la máquina jugadores registrados y de los que almacenaremos sus resultados. Para ello se incluirán las siguientes modificaciones:

1. Se incluirán dos nuevos tipos de plantas y de zombis, uno obligatorio y otro opcional.

a. Nuevas plantas:

- i. **Nuez:** es una planta que sirve como barrera. Tiene coste 50 soles y resistencia de 10 puntos de vida. Esta planta no produce ningún daño.
- ii. **OPCIONAL: Petacereza:** es una planta que explota y quita 10 puntos de daño a todos los zombis que están a su alrededor (en casillas contiguas). La planta una vez que explota desaparece. La explosión ocurre dos turnos después de ser plantada. Los zombis también se pueden comer a la planta que tiene de resistencia 2 puntos de vida. Su coste es 50 soles.

b. Nuevos zombis:

- i. **Caracubo:** es más lento, pero más resistente. Camina un paso cada 4 ciclos y tiene resistencia 8 puntos de vida.
- ii. **OPCIONAL: Deportista:** es más rápido, pero menos resistente. Camina un paso cada ciclo y tiene resistencia 2 puntos de vida.

Estos nuevos zombis ejercen el mismo daño que el zombi común del ejercicio 2. Los diferentes tipos de zombis aparecen con la misma probabilidad.

2. Antes de empezar una partida necesitamos que el usuario se registre en la aplicación introduciendo su DNI y nombre. Si ya está registrado solo necesita introducir su DNI para proceder a localizar su información y empezar a jugar. Se utilizará un Hashmap para almacenar los datos de los jugadores del torneo.

3. Al finalizar cada partida, el usuario obtiene *puntos* en caso de que haya ganado la partida. Para ello se sumará el valor de todas las plantas que haya en el tablero más el de los soles que no se hayan gastado.
4. Serialización y archivos de texto:
 - a. Se guardan los datos de los jugadores de la aplicación, incluyendo partidas jugadas, ganadas / perdidas, dificultad de la partida y puntos obtenidos en cada una de las partidas y en total (clasificadas por dificultad).
 - b. Incluir la opción de crear una ficha de un jugador, dado su NIF. Esta opción crea un fichero de texto con todos los datos del mismo (personales, partidas, puntos, etc.).
 - c. *OPCIONAL*: El sistema permitirá salir de una partida a medias, guardando su estado, y continuar en otro momento.
5. Se incluirá la opción de visualizar, en cualquier momento, el ranking de jugadores. Se mostrará una tabla con los nombres de los jugadores ordenados por puntuación total, en función de la dificultad.
6. La interfaz de la aplicación pasa a ser gráfica con la librería Swing y de diseño libre. Para mostrar el tablero se pueden seleccionar múltiples opciones, os proponemos dos sencillas:
 - a. Utilizar un “JTextArea” donde se pueden poner los caracteres de la misma forma que en la consola de Java.
 - b. Utilizar una matriz de “JLabel” donde se pueden mostrar las plantas y los zombis (podéis encontrar las imágenes de los personajes del juego en internet). Para facilitar su diseño se puede hacer que el número de filas y columnas sea fija, por ejemplo de 7x7.



2. Entrega

La aplicación deberá presentar una interfaz gráfica de usuario, intuitiva y fácil de utilizar. Las clases del dominio de la aplicación deberán estar en un paquete diferente a las de la gestión de la interfaz de usuario. El código de la interfaz de usuario y de las clases de la lógica de negocio debe estar estrictamente separado, de manera que estas clases puedan reutilizarse en el futuro en otros contextos, por ejemplo, si se quiere hacer una versión web de la interfaz.

Para poder comprobar un buen funcionamiento de la aplicación, se deberá poder arrancar con un mínimo de 5 jugadores registrados y 10 partidas realizadas, con toda su información asociada.

Los programas fuentes deberán ser fáciles de mantener por cualquier programador experto en Java. Deberán estar bien estructurados y contendrán los comentarios necesarios para un fácil seguimiento de código, incluyendo la documentación generada con JavaDoc.

Los requisitos recogidos en el presente documento, debe entenderse como los mínimos que debe cumplir el sistema, pudiendo ser presentadas cuantas mejoras se crean convenientes.

Entregables para Blackboard

✓ *Documento explicativo*

La documentación de la aplicación debe redactarse para ser leída por un hipotético técnico informático que tuviese que evaluar la calidad de la aplicación desarrollada. No incluirá código fuente. Los apartados a incluir son:

1. **Portada:** en la portada de la memoria deberá aparecer:
Grado en Curso 2018/19.
Práctica de POO Mayo 2019
Autor: Apellidos, Nombre – DNI
2. **Especificación de requisitos:** mediante *diagramas de casos de uso* se deberán describir las acciones principales que los usuarios de la aplicación puedan realizar.
3. **Análisis y diseño de la aplicación:** descripción del análisis y diseño de cómo se ha desarrollado la aplicación y qué elementos principales intervienen en la misma. Para realizarlo se construye un diagrama de clases sencillo, identificando las clases de la lógica de negocio.
4. **Implementación (JavaDoc):** para completar este apartado se debe haber realizado una correcta documentación de la aplicación donde se han detallado sus clases principales (clases de lógica de negocio) y su funcionalidad y las estructuras de datos utilizadas para almacenar la información que maneja la aplicación. Esto se puede realizar mediante los comentarios del código fuente para la generación de la documentación en JavaDoc.

5. **Manual de usuario (descripción del funcionamiento):** en este apartado se explicará cómo interactúa un usuario con la aplicación. Es conveniente capturar las pantallas de la aplicación (se hace con las teclas Alt + Impr Pant y luego se pega en el documento) y mostrar el funcionamiento de las distintas opciones, mensajes de error, etc.

✓ **Proyecto NetBeans**

En la entrega se incluirá un proyecto NetBeans listo para arrancar la aplicación, para comprobar que se cumplen las especificaciones formuladas. 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.

✓ **Consideraciones**

La evaluación de la práctica se llevará a cabo mediante una entrevista individual (aunque se haya realizado en grupo) en la que el alumno deberá mostrar el funcionamiento de su aplicación y someterse a las preguntas que se consideren necesarias. Las pruebas se podrán realizar en el ordenador portátil del alumno.

La entrega de la práctica con su memoria correspondiente se realizará a través de la sección Trabajos de Blackboard con fecha límite 27 de mayo de 2019. Cada profesor establecerá el lugar, fecha y hora para realizar la defensa de la práctica. Esta fecha es orientativa ya que puede haber profesores que necesiten cambiarla.