UNIVERSIDAD
COMPLUTENSE
MADRID

Tecnología de la Programación

Presentación de la Práctica 1

(Basado en la práctica de Miguel V. Espada)

Ana M. González de Miguel (ISIA, UCM)

Índice

1. Introducción

- ✓ Esta práctica pretende ser un **inicio** a la orientación a objetos y a Java.
- ✓ Otros **objetivos** son: uso de arrays y enumerados, manipulación de cadenas con la clase *String* y, entrada y salida por consola.
- ✓ Fecha de entrega: **22 de Octubre** de 2018 a las 9:00.
- ✓ La entrega debe realizarse utilizando el campus virtual, no mas tarde de la fecha indicada, con los siguientes contenidos:
  - Directorio *src* con el código de todas las clases de la práctica.
  - Fichero *alumnos.txt* con el nombre de los componentes del grupo.
- ✓ Durante las próximas prácticas se va a desarrollar progresivamente nuestra propia versión del juego **plantas contra zombis**. Esta primera usa la interfaz consola.

- **Plantas contra zombis** es un juego desarrollado para móviles muy popular estilo *tower defense*. Según la wikipedia: "los zombies aparecen mientras suena una alarma. 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. Los zombis presentan diferentes atributos y habilidades, como cavar por debajo de las plantas o saltar por encima de estas con ayuda de pértigas, llevar diferentes objetos como casco o conos, escaleras o globos."
- A nivel visual el juego se presenta como **una cuadrícula** en la que el jugador coloca las plantas y los zombis aparecen del lado derecho del tablero. El jugador tiene que resistir el ataque de los zombis destruyéndolos a todos.

- ✓ Si uno de los zombis supera la defensa y llega hasta el final del tablero, **el usuario pierde la partida**.
- ✓ En el juego original hay multitud de **plantas**. Cada planta difiere en su función, en su coste, en el tiempo que tarda en crecer, la cantidad de daño que infiere a los zombis, su resistencia, etc. Consulta la wikipedia para ver la lista entera. En esta práctica solo usaremos:
  - **Lanzaguisantes**: dispara guisantes a los zombis
  - **Girasol**: proporciona sol.
- ✓ Cada planta tiene un coste que se mide en unidades de sol. El usuario puede ir acumulando soles para plantar nuevas plantas. Y estos soles se acumulan gracias a los girasoles.
- ✓ En el juego también hay distintos tipos de **zombis**. Nosotros usaremos sólo el **zombi común**.

2. Descripción de la Práctica

- ✓ En esta primera práctica el juego consta de un tablero de 4 x 8 casillas (4 filas por 8 columnas).
- ✓ En cada casilla podemos colocar una sola planta o un zombi.
- ✓ Los zombis avanzan de derecha a izquierda y ganan la partida cuando llegan al final del tablero.
- ✓ El jugador tiene que matar un número de zombis para ganar.
- ✓ En cada **ciclo de juego** se realizan las siguientes acciones:
  1. **Update**. Se actualizan los objetos que están en el tablero.
  2. **Draw**. Se pinta el tablero y se muestra la información del juego.
  3. **User command**. El usuario puede realizar una acción como sumar una planta en una casilla. También puede no hacer nada.
  4. **Computer action**. El ordenador puede añadir un zombi en una fila. Cuando y donde se colocan los zombis es aleatorio y depende del nivel.

- ✓ Los **objetos** del juego son los siguientes.
  - **Planta lanzaguisantes:**
    - Comportamiento: dispara guisantes a los zombis.
    - 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 delante.
  - **Girasol:**
    - Comportamiento: genera soles.
    - Coste: 20 soles.
    - Resistencia: 1 punto de daño.
    - Frecuencia: genera 10 soles cada dos ciclos.
    - Daño: 0 (no ejerce ningún daño a los zombis).
  - **Zombi Común:**
    - Comportamiento: avanza y come plantas.
    - Resistencia: 5 puntos de daño.
    - Daño: 1 punto de daño.
    - Velocidad: 1 casilla cada 2 ciclos.

- ✓ Las actualizaciones (**update**) son las siguientes, y en este orden:
  - Los girasoles pueden acumular sol.
  - Las plantas lanzaguisantes disparan guisantes a los zombis que estén a su alcance.
  - Los zombis avanzan (solo si no tienen ni planta ni zombi delante).
  - Si un zombi está en la casilla adyacente de planta ejerce daño a la planta (disminuye su resistencia o vida).
  - Si una planta o un zombi llega a resistencia 0 desaparece del tablero.
- ✓ El juego finaliza si durante el *update* todos los zombis son destruidos o uno de ellos llega al final de la fila.
- ✓ Cuando el juego termine debe mostrar “Player wins” o “Zombies win” según quien ha sido el ganador.
- ✓ Los elementos del juego se actualizan en el orden en que fueron introducidos.

- **Draw**. En cada ciclo se pinta el estado actual del tablero (con todos sus objetos) así como el ciclo de juego en el que nos encontramos (inicialmente 0), el número de soles acumulados gracias a los girasoles y el número de zombis que quedan por aparecer.
- En el enunciado tienes una imagen del tablero.
- Al lado de cada objeto del tablero aparece la vida (resistencia) que le queda.
- Después del tablero se muestra también el *prompt* del juego para pedir al usuario la siguiente acción.

- ✓ **User Command**. Se le pregunta al usuario que es lo que quiere hacer, a lo que puede responder con una de las siguientes alternativas:
  - **add <plant><x><y>**: Se añade una planta en la casilla x,y del tablero si se tiene el número suficiente de soles. No se puede añadir en una casilla ocupada por una planta o un zombi.
  - **reset**: se reinicia la partida volviendo a la configuración inicial.
  - **list**: se muestra el nombre de la plantas disponibles con su coste y su daño.

    [S]unflower: Cost: 20 suncoins Harm: 0

    [P]eashooter: Cost: 50 suncoins Harm: 1
  - **none**: el usuario no realiza ninguna acción.
  - **exit**: permite salir del juego mostrando el mensaje: “Game Over”.
  - **help**: este comando solicita a la aplicación que muestre la ayuda sobre como usar los comandos, con una breve descripción de cada uno (ver el enunciado).

✓ Observaciones sobre los comandos:

- Los comandos podrán estar escritos en mayúsculas o minúsculas.
- Las plantas se identifican en inglés o por su inicial (S, P).
- La aplicación debe permitir usar la primera letra de cada comando en lugar del nombre completo [a]dd, [n]one, [l]ist, [h]elp, [e]xit.
- Si no se indica comando se identifica como [n]one y se avanza al siguiente ciclo.
- Si el comando esta mal escrito, no existe o no se puede ejecutar, la aplicación debe mostrar un mensaje de error.
- En el caso en que el usuario ejecute un comando que no cambia el estado del juego o un comando erroneo, el tablero no se debe repintar.

- ✓ **Computer Action**. En esta primera práctica el ordenador tendrá un comportamiento pseudoaleatorio.
- ✓ Uno de los parámetros de entrada (ver más adelante) es el **nivel** del juego. Cada nivel determina:
  - El número de zombis que aparecen en la partida.
  - La frecuencia de aparición de los zombis. Es decir, la probabilidad de aparición de un zombi en un ciclo.
- ✓ Los tres niveles permitidos en el juego son:
  - EASY: Numero de zombis: 3, Frecuencia: 0.1
  - HARD: Numero de zombis: 5, Frecuencia: 0.2
  - INSANE: Numero de zombis: 10, Frecuencia: 0.3
- ✓ Cuando aparece un zombi el ordenador lo coloca en una de las filas de manera aleatoria. Para controlar este comportamiento aleatorio usamos una semilla (**seed**).

- ✓ El programa debe aceptar un parámetro obligatorio y otro opcional por línea de comandos.
  - El primer parámetro llamado **level** es el nivel del juego.
  - El segundo parámetro llamado **seed** es la semilla usada para el comportamiento pseudoaleatorio del juego.

3. Implementación

- ✓ Paquete de clases: *tp.p1*
- ✓ La práctica necesita al menos las siguientes clases:
  - ***Sunflower, Peashooter, Zombie***. Son las clases que encapsulan los elementos (objetos) del juego.
  - ***SunflowerList, PeashooterList, ZombieList***. Contienen arrays de elementos del juego y métodos auxiliares para su gestión.
  - ***Game***. Encapsula la lógica del juego.
  - ***ZombieManager***. Lleva la cuenta de los zombies quedan por salir.
  - ***SuncoinManager***. Lleva el control de los suncoins.
  - ***Level***. Es una clase enumerada para los tres niveles del juego.
  - ***GamePrinter***. Sirve para pintar el juego.
  - ***Controller***. Controla la ejecución del juego.
  - ***PlantsVsZombies***. Contiene el método *main* y lee los valores de los parámetros de la aplicación.

4. Observaciones

- ✓ Sólo se creará un objeto de la clase *Controller*. Igual sucede con la clase *Game* (que representa la partida y sólo puede haber una activa).
- ✓ Al final del enunciado (en la sección Anexo) se proporciona parte de la implementación de la clase *MyStringUtils.* Dicha clase contiene 2 métodos para formatear strings y facilitar la impresión del tablero. Su uso no es obligatorio.
- ✓ En el enunciado se proporciona también un ejemplo de ejecución. La salida de la práctica bebe coincidir con este ejemplo.

5. Apéndice de Código

```
public class MyStringUtils {
   public static String repeat(String elmnt, int length) {
     String result = "";
     for (int i = 0; i < length; i++) {
        result += elmnt;
     }
     return result;
   }
   public static String centre(String text, int len) {
      String out = String.format("%"+len+"s%s%"+len+"s", "",text,"");
      float mid = (out.length()/2);
      float start = mid - (len/2);
      float end = start + len;
      return out.substring((int)start, (int)end);
   }
} // MyStringUtils
```