EJERCICIOS DE UTILIZACIÓN DE TIPOS ABSTRACTOS DE DATOS

1. El juego *pasiegos vs zombies*.

Aprovechando el auge y gancho comercial de los muertos vivientes, también conocidos como *zombies, caminantes* y en ocasiones *infectados*, una empresa decide crear un sencillo juego denominado *pasiegos vs zombies*.

La historia es bastante simple: debido a una extraña alineación de diversos planetas, se ha producido una mutación en una cierta bacteria que se aloja en el cerebro de los cuerpos humanos de personas fallecidas, haciéndolos volver a la vida bajo la forma de *zombies*. Dichos *zombies* poseen como ciertas características comunes, como el emitir sonidos guturales ininteligibles, andar de forma lenta y cojeando, y lo que es más molesto, tratar de comerse vivos a seres humanos no zombies.

Para evitar esto último, los habitantes del Valle del Pas emprenden una lucha sin cuartel contra los muertos vivientes. Los habitantes del Valle del Pas se defienden con dos armas:

(1) *Lanzamiento de sobao duro*: Se ha descubierto que los sobaos con una alto porcentaje de mantequilla y que llevan más de un mes fuera de la nevera y en un ambiente de escasa humedad, adquieren una consistencia que adecuadamente lanzado, puede poner fin a la vida de un *zombie*.

(2) *Lanzamiento de bolo pasiego*: Al igual que lo anterior, se ha descubierto que considerando un zombie como un bolo, el impacto en el propio *zombie* con una bola de bolos puede poner fin de inmediato su vida. De no ser así, el *zombie* quedará aturdido, teniendo el pasiego aún la oportunidad de acabar con su vida al birle.

El aspecto de la interfaz gráfica de este juego es el que aparece en la Figura 1.

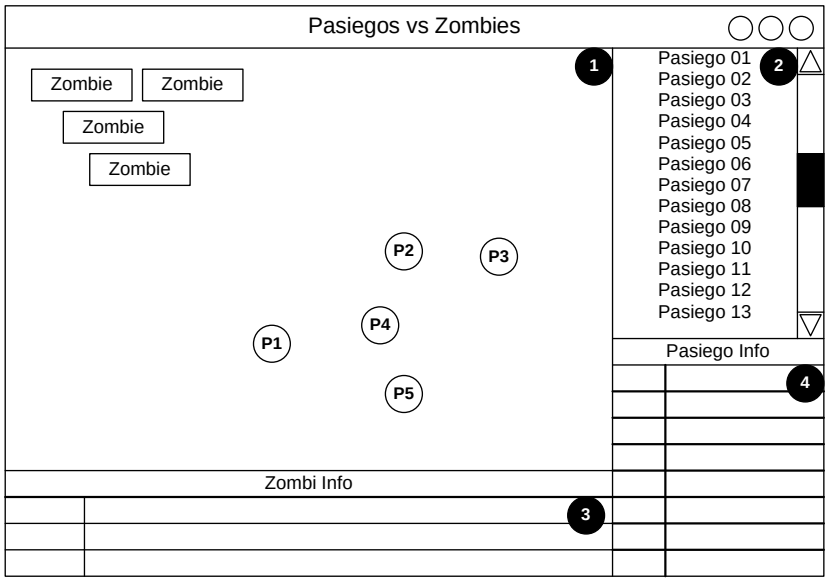

Figura 1. Boceto de la interfaz del juego *Pasiegos vs Zombies*

En la parte central (Figura 1, etiqueta 1) aparece el terreno de juego. Lo que el usuario ve es un bello paisaje del Valle del Pas con varios zombies y pasiegos moviéndose y matándose entre ellos. En la parte derecha, arriba (Figura 1, etiqueta 2) de la pantalla se muestra una lista, ordenadas por mote, con los pasiegos actualmente vivos que están luchando contra los zombies. En la parte inferior derecha (Figura 1, etiqueta 4), se muestran los datos de un pasiego concreto, mientras que en la parte inferior de la pantalla (Figura 1, etiqueta 3) se visualizan los datos de un *zombie* particular.

Tanto los zombies como los pasiegos poseen un identificador generado por el sistema en el momento de su creación y que es único para cada individuo. Dicho identificador es un entero distinto de cero. Además, cada pasiego o zombie posee un natural entre 0 y 100 que indica cuanta vida o energía le queda. Tanto los zombies como las personas poseen un nombre y dos apellidos (los zombies fueron personas en un pasado no muy lejano).

El número de zombies al inicio de cada partida es fijo y no existe regeneración de zombies, es decir, no pueden aparecer nuevos zombies durante la partida.

Además, del nombre y los apellidos, todo pasiego posee un mote. Curiosamente, no existe una gran variedad de nombres y apellidos en el Valle del Pas, por lo que es frecuente encontrarse con gente que se llama de forma muy similar, e incluso igual. Por ejemplo, es normal que si existe un “Roberto Mantecón Sañudo”, exista también un “Roberto Sañudo Mantecón”. No obstante, no se conoce dos pasiegos con un mismo mote. Por tanto, se puede considerar que el mote es único para cada pasiego. Además cada pasiego posee de inicio un número finito de sobaos endurecidos y una bola de bolos con la cual luchar contra los zombies. Dado el carácter competitivo de estos habitantes, cada pasiego debe guardar los zombies que ha matado, para luego poder alardear de sus hazañas en el bar del pueblo.

Desde un punto de vista más técnico, el área central de la pantalla es una inmensa matriz que contiene información acerca de lo que hay que dibujar en pantalla y lo que contiene cada pixel. Si un pixel está ocupado por un pasiego, contiene el identificador de dicho pasiego. Si está ocupado por un zombie, contiene el identificador de dicho zombi. Si dicho píxel no tiene ni un pasiego ni un zombie, contiene como identificador el valor 0.

Cuando se selecciona un zombie en la pantalla, se debe mostrar la información sobre el mismo en la zona de *Zombie Info*. Lo mismo ocurre con un pasiego, pero tanto cuando se selecciona el pasiego desde la ventana de juego como desde la lista de pasiegos vivos (Figura 1, etiqueta 2). En ambos casos, dicho pasiego pasa a ser el pasiego activo y es el que puede ejecutar acciones sobre los zombies, tales como lanzar sobaos o jugar a los bolos con ellos.

En ambas acciones el pasiego selecciona un zombie y ejecuta la acción. Un lanzamiento de sobao le resta al zombie 25 puntos de vida, mientras que un bolazo le resta 50 puntos de vida. Además el pasiego activo puede pasar a modo “*generación de nuevos pasiegos*”, modo en el cual el pasiego entra en un estado de “*hibernación*”. Al salir de este estado, se produce un nuevo pasiego, que se añadirá a la colección de pasiegos vivos.

Para la construcción de la interfaz gráfica, se utilizan las siguientes clases. La descripción de los métodos de dichas clases se proporciona en el Apéndice A.

(1) ***PasiegoGraphicalList***: Es el componente de la Figura 1, etiqueta 2. Es una lista de pasiegos con scroll lateral. Cuando se selecciona con el ratón un pasiego de la lista, se invoca el método *onPasiegoSelected*. Cada vez que nace o muere un pasiego, hay que invocar la operación *displayList* para actualizar dicha lista.
(2) ***PasiegoInfoDisplay***: Es el componente de la Figura 1, etiqueta 4. Muestra la información de un pasiego determinado. No responde a eventos de ratón.
(3) ***ZombieInfoDisplay:*** Es el componente de la Figura 1, etiqueta 3. Muestra la información de un zombie determinado. No responde a eventos de ratón.
(4) ***GameZone:*** Es el tablero de juego por donde se mueve los zombies y los pasiegos. Se trata de un componente gráfico avanzado que muestra una escena de juego en 3D. Cuando se pincha sobre él, se ejecuta el método *onPixelClicked*.

Siempre que un método de estás clases es invocado como consecuencia de la detección de un evento, estos métodos invocan métodos de una clase *Juego*, que centraliza el control del juego, para que actúe en consecuencia y actualice el estado del resto de los componentes gráficos de forma acorde.

Con la información proporcionada se pide:

(1) Especificar que atributos contendrían las clases *Zombie* y *Pasiego* resaltando los invariantes que deben cumplir sus atributos.
(2) Especificar los atributos de una clase *Juego*, indicando que TADs son los más adecuados para almacenar tanto los pasiegos como los zombies.
(3) Para cada TAD identificado en el punto 2, indicar que implementación resultaría la más adecuada, justificándolo adecuadamente.
(4) Indicar que modificaciones habría que realizar, si hubiese que realizar alguna, sobre las clases *Pasiego* y *Zombie*, para poder almacenarlos en los TADs elegidos en el punto 2 y usando las implementaciones escogidas en el punto 3.
(5) Añadir a la clase Juego el procedimiento *addPasiego(p : Pasiego)* que sirva para dar de alta un nuevo pasiego. Calcular su complejidad.
(6) *addZombie(z : Zombie)* que sirvan para dar de alta un nuevo zombie en el juego. Calcular su complejidad.
(7) Implementar la operación *onPixelClicked* añadiendo para ello las operaciones que se consideren necesarias en la clase *Juego*.
(8) Implementar la operación *onPasiegoSelected* añadiendo para ello las operaciones que se consideren necesarias en la clase *Juego*.
(9) Calcular la complejidad de todas las operaciones creadas en los puntos (7) y (8).
(10) Implementar la operación *lanzamientoSobao(idZombie: Natural, idPasiego : Natural)* en la clase *Juego*. Se puede asumir que ambos identificadores son correctos, pero nada más. Dicha operación debe ejecutar el lanzamiento de un sobao del pasiego que se pasa como parámetro al zombie que se pasa como parámetro.
(11) Calcular la complejidad de la operación del punto (10).

(12) Implementar la operación *lanzamientoBolo(idZombie: Natural, idPasiego : Natural)* en la clase *Juego*. Se puede asumir que ambos identificadores son correctos, pero nada más. Dicha operación debe ejecutar el lanzamiento de un sobao del pasiego que se pasa como parámetro al zombie que se pasa como parámetro.

(13) Calcular la complejidad de la operación del punto (12).

(14) Diseñar una función de dispersión para la clase *Zombie*.

(15) Diseñar una función de dispersión para la clase *Pasiego* que no involucre al identificador del mismo.

NOTA: Se puede suponer que existen implementaciones predefinidas para los TADs seleccionados. La interfaz de dichos TADs se puede suponer similar a la de las especificaciones algebraicas disponibles en moodle.

Apéndice A. Descripción de las clases de la interfaz gráfica

```
CLASE ZombieInfoDisplay ES
    // Visualiza en el componente gráfico la información del zombie q
   // que se le pasa como parámetro
   // Pre :  z  != NULO
    PROC display (z : Zombie); // O(1)
FINCLASE

CLASE PasiegoInfoDisplay ES
    // Visualiza en el componente gráfico la información del zombie q
   // que se le pasa como parámetro
   // Pre :  p  != NULO
    PROC display (p : Pasiego); // O(1)
FINCLASE

CLASE GameZone ES
  // Cada casilla de esta matriz contiene el identificador del zombie o pasiego que está
  //dibujado en
  HEREDABLE pantalla: VECTOR DIM 1080 DE VECTOR DIM 1024 DE NATURAL;
  HEREDABLE controlJuego: Juego;

  // Este método se invoca cuando se hace click con el ratón sobre un píxel de la zona de juego
  // Pre: dimensionesCorrectas(fila, columna)
  PROC onPixelClicked(fila, columna : NATURAL);

  // Borra un zombie de tablero de juego
  PROC removeZombie(z : Zombie); // O(1)

  // Borra un zombie de tablero de juego
  PROC removePasiego(z : Zombie); // O(1)
FINCLASE
```

CLASE *PasiegoGraphicalList* ES
// Muestra una lista de pasiegos por pantalla
PROC displayList(pasiegos: Lista(Pasiego)) // O(n) [n es el tamaño de la lista]

// Este método se invoca cada vez que se selecciona un pasiego de su lista.
PROConPasiegoSelected(mote: CadenaCaracteres);
FINCLASE

CLASE Juego ES
HEREDABLE list : PasiegoGraphicalList;
HEREDABLE zInfo : ZombieInfoDisplay;
HEREDABLE pInfo : PasiegoInfoDisplay;
HEREDABLE zone : GameZone;

HEREDABLE pasiegoActivo : Pasiego;

// Se supone que existe un constructor adecuado que inicialice correctamente estas
// variables

FINCLASE

Pablo Sánchez Barreiro.