

Actividad 5: La mansión zombie v2

Una aproximación terroríficamente gráfica al código

Módulo: Programación

Hugo María Vegas Carrasco

Índice

Introducción.....	3
Reglas del juego.....	3
Inicio del juego.....	3
Turno.....	6
Combatir contra un zombie.....	7
Buscar por la habitación.....	10
Avanzar a otra habitación / Salir de la mansión.....	11
Curarse.....	12
Fin del juego.....	12
Guardar y cargar partida.....	13
Implementación.....	14
Especificación de la entrega.....	15

Introducción

Bienvenidos OTRA VEZ a la mansión zombie. Parece que os gusta pasar miedo; estáis en el lugar indicado.

Esta versión es una actualización de la anterior versión que trabajamos en el primer trimestre. Ahora somos más sabios y precavidos, y además conocemos la potencia de las interfaces gráficas y los ficheros.

En este **juego** interactivo el usuario tendrá una ventana donde irá eligiendo una serie de acciones, basadas en la situación actual de la habitación donde se encuentra. Más adelante, cuando hablemos de las reglas del juego, entraremos en detalle.

En el **juego** simularemos una partida un tanto rolera, donde los combates y ciertas decisiones irán de la mano de tiradas de **dados**.

El **superviviente** irá avanzando, luchando contra molestos **zombies** los cuales intentarán que no salga con vida de la siniestra mansión.

Prepara tu teclado, tu ratón y tu mente para adentrarte una vez más en el oscuro mundo de la programación. Sólo aquellas personas que sean valientes con su código saldrán ilesas de este lugar...

Reglas del juego

A continuación vamos a describir de nuevo la lógica del **juego** para tener una visión global de cómo se tiene que comportar el mismo en base a las decisiones que tome el **superviviente**.

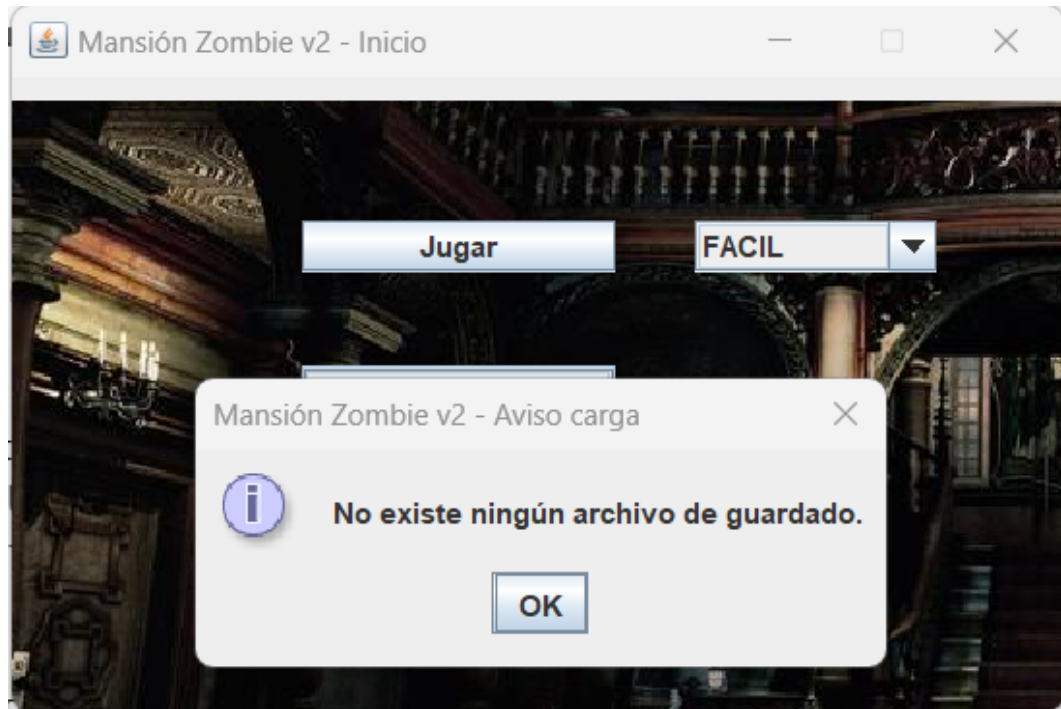
Inicio del juego

Ahora lo primero que va a ver el superviviente (jugador) cuando inicie el juego será una ventana principal parecida a la siguiente:

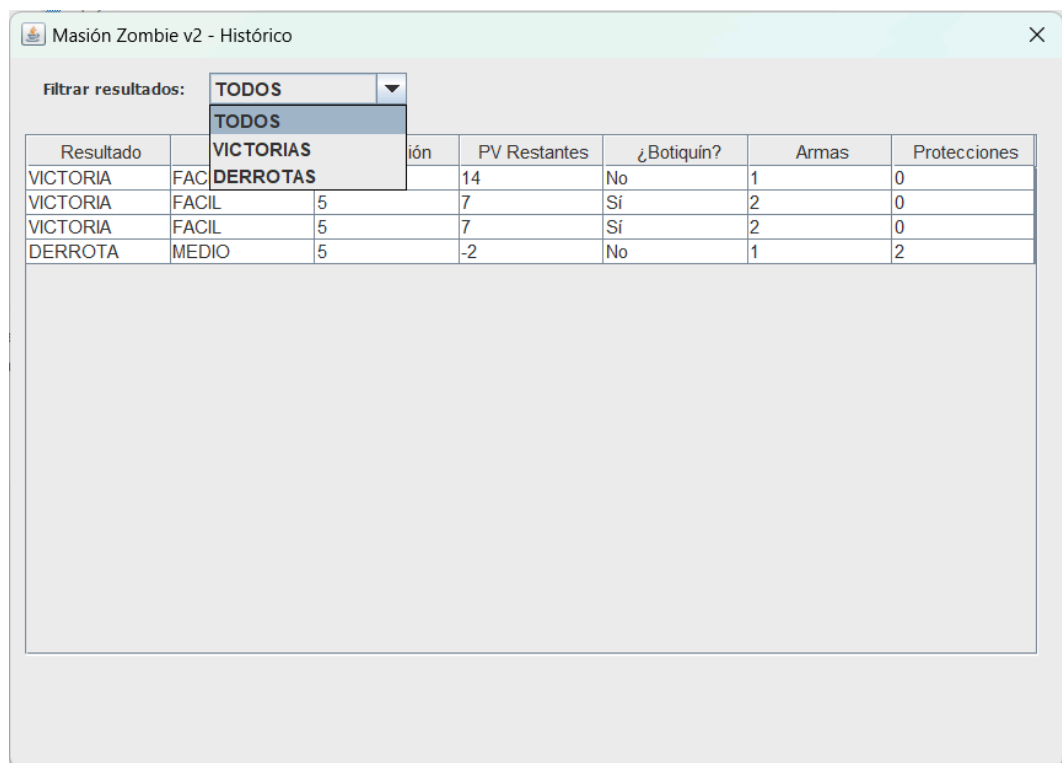


En ella el jugador podrá realizar las siguientes acciones:

- **Jugar:** empezará una nueva partida en base a la dificultad seleccionada (que por defecto será "fácil"). Escoja la opción que escoja, se tendrán que inicializar todos los componentes relativos al **juego** y al **superviviente**.
- **Seleccionar la dificultad:**
 - ◆ Fácil: la mansión se supera jugando 5 habitaciones.
 - ◆ Medio: la mansión se supera jugando 7 habitaciones.
 - ◆ Difícil: la mansión se supera jugando 10 habitaciones.
- **Cargar partida:** podrá recuperar los datos de la partida que tenga guardada. En caso de no haber ninguna partida guardada se deberá notificar.



→ **Ver histórico:** el jugador podrá revisar los resultados de sus últimas partidas. Además podrá filtrar si quiere ver todas, sólo las victorias o sólo las derrotas.



Respecto al **juego** tendremos que tener en cuenta los siguientes datos (entre paréntesis el valor inicial):

- Número máximo de habitaciones que vamos a jugar (5-7-10).

- Número de intentos de búsqueda que quedan en la habitación donde se encuentra el jugador (3).
- Habitación actual en la que se encuentra el jugador (1).
- Número de zombies que están activos en la habitación (1).

Respecto al **superviviente** tendrán en cuenta los siguientes datos (entre paréntesis el valor inicial):

- Número máximo de puntos de vida (20).
- Puntos de vida actuales (20).
- Puntos de ataque (4).
- Lleva botiquín (No).
- Cantidad de armas (0).
- Cantidad de protecciones (0).

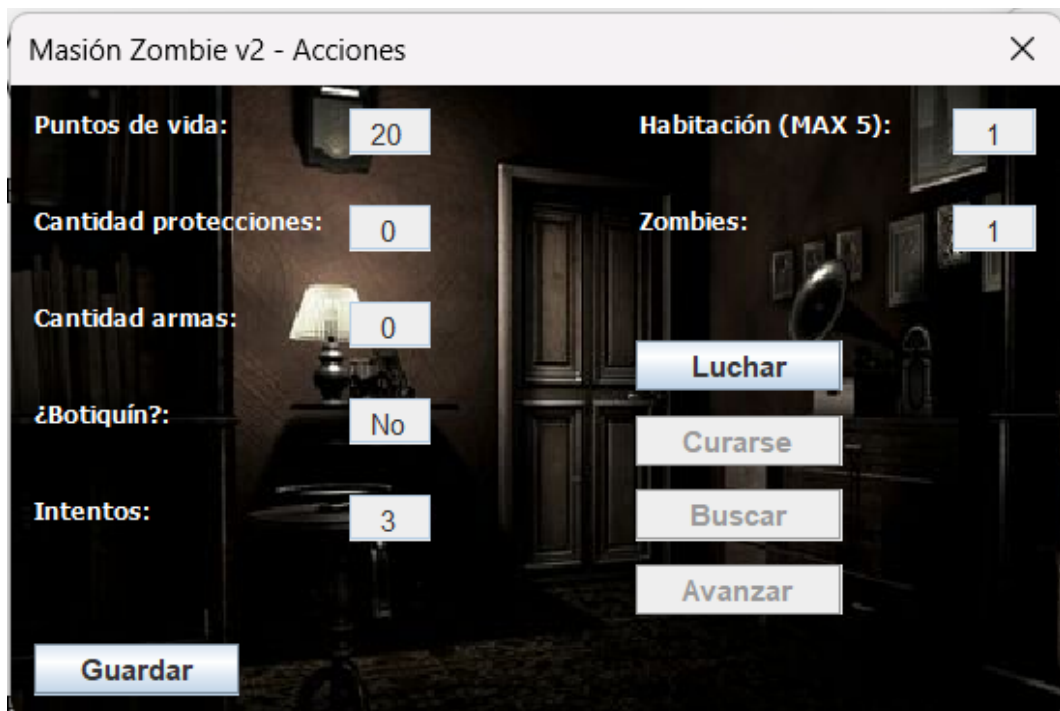
Turno

El turno se va desarrollando en base a una serie de acciones que puede tomar el **superviviente**. Numeradas son:

1. Combatir contra un **zombie**: sólo estará disponible si hay al menos un **zombie** activo en la habitación.
2. Buscar por la habitación (indicando los intentos que quedan disponibles): sólo estará disponible si no hay **zombies** activos en la habitación y queda al menos un intento disponible en la habitación actual.
3. Avanzar a otra habitación / Salir de la mansión: sólo estará disponible cuando no haya **zombies** activos en la habitación.
4. Curarse: sólo estará disponible cuando no haya **zombies** activos en la habitación y tengamos un botiquín encima.

Con estas premisas el **superviviente** irá avanzando en la mansión. Podrá intentar salir de forma directa de la misma o podrá arriesgarse a conseguir mejoras y ayudas para que el paso a las siguientes habitaciones sea más sencillo realizando búsquedas. ¡Pero ojo! buscar es arriesgado; podremos atraer cosas indeseadas a nuestra habitación actual.

La ventana relativa a todas estas acciones será parecida a la siguiente:



Habrá que controlar que los botones estén disponibles o no en base a si se cumplen las reglas anteriormente descritas. Además ninguna de la información relativa a la partida (vidas, objetos, ...) puede ser editable; los valores sólo irán cambiando en función de las acciones del jugador y será la propia aplicación la que gestione dicha información.

Combatir contra un zombie

Cuando elegimos esta opción nos enfrentamos a un sólo **zombie**, independientemente de la cantidad de **zombies** que haya en la habitación. Para representar que los **zombies** son seres diferentes con características distintas (unos están más pochos que otros) cuando combatamos, el **zombie** que va a luchar se generará de forma automática con unos atributos al azar. La información relativa a un **zombie** es la siguiente:

- Puntos de vida actuales.
- Puntos de ataque.

Como podemos ver, se parecen un poco a los **supervivientes**, signo de que antes eran otra cosa muy diferente a lo que son ahora.

La forma en que vamos a generar el **zombie** es la siguiente: tanto los puntos de vida como los de ataque serán el resultado de generar un número aleatorio entre 0 y 1, el cual vamos a sumarle después un incremento de 2 + la habitación en la que nos encontremos -1. La fórmula sería:

$$\text{numeroAleatorio}(2) + 2 + (\text{Habitación} - 1)$$

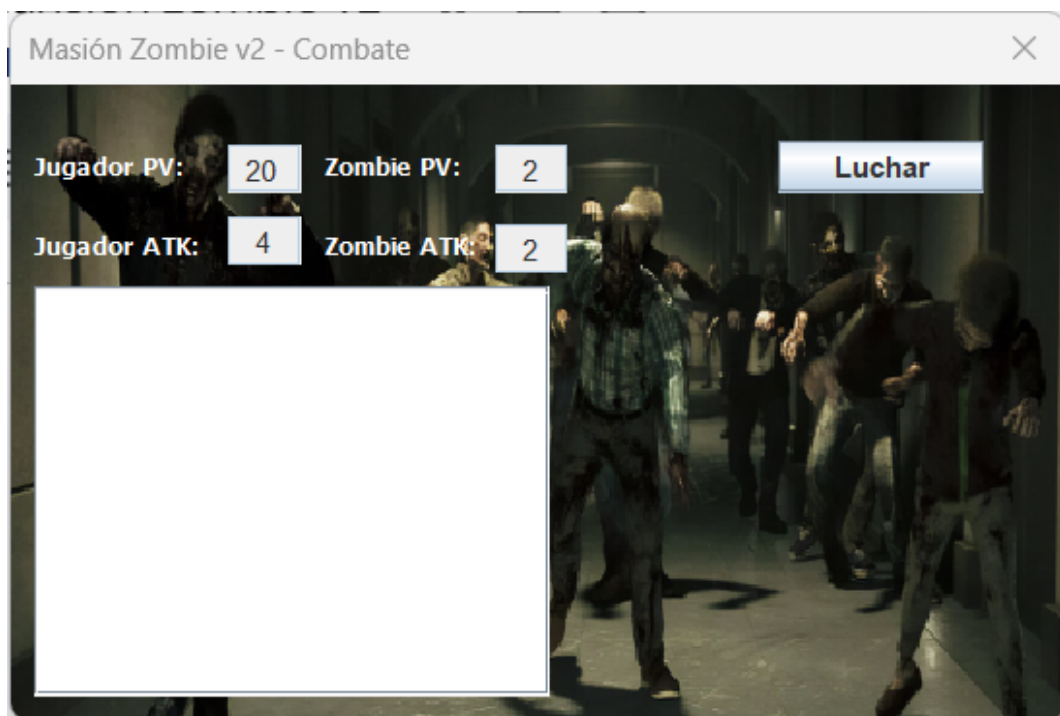
Esto va a hacer que a medida que vayamos avanzando, las características mínimas de los **zombies** irán subiendo. El componente aleatorio sirve para que incluso el **zombie**, con un poco de suerte, pueda darnos un buen susto ya desde las primeras habitaciones.

Una vez el **zombie** está listo debemos luchar hasta que uno de los combatientes muera. Primero ataca el **superviviente** (para algo está un poco más vivo) y en caso de que el **zombie** siga en pie este responderá.

La forma en que se asigna daño es la siguiente:

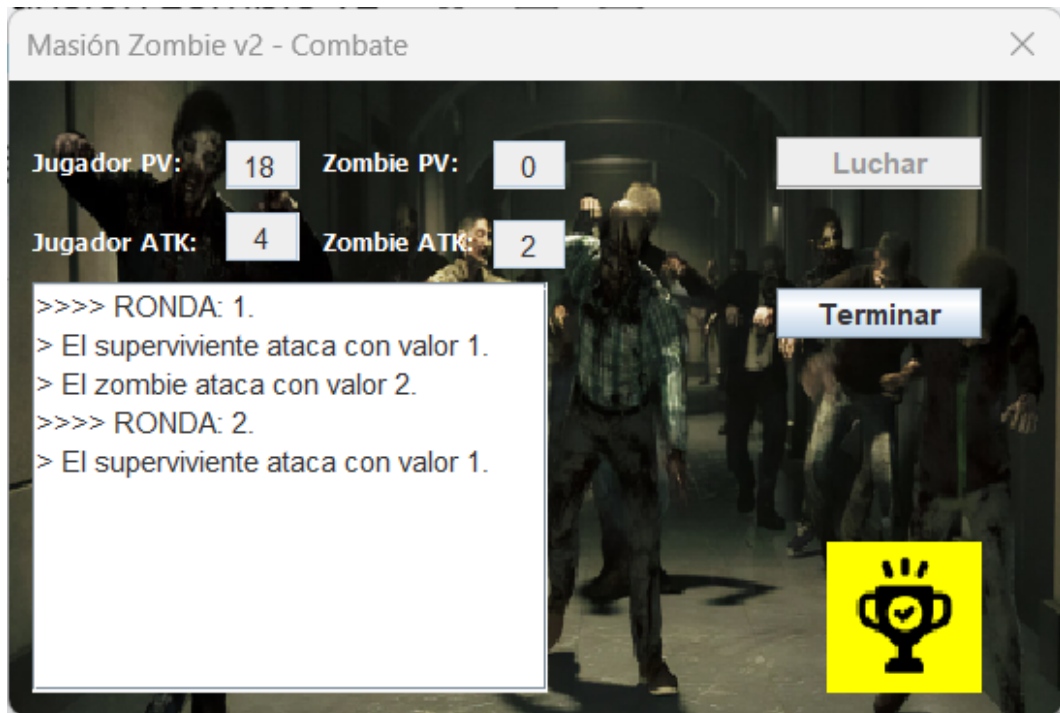
- El **superviviente** ataca con un **dado** igual a su atributo de puntos de ataque. En caso de que tenga armas, se sumarán al resultado de la tirada ataque (1 por cada arma), no al valor del **dado**.
- El **zombie** recibe el ataque y resta el valor a sus puntos de vida.
- Si el **zombie** sigue vivo le toca atacar. Para ello, igual que el **superviviente**, lanza un **dado** con sus puntos de ataque, con la diferencia de que esta vez no se suma nada adicional al ataque porque los **zombies** no van armados.
- Por último, el **superviviente** recibe el ataque y resta el daño del **zombie** a sus puntos de vida; sin embargo, por cada protección que lleve encima, ese daño se verá reducido en 1.

La ventana que gestiona este combate es la siguiente:

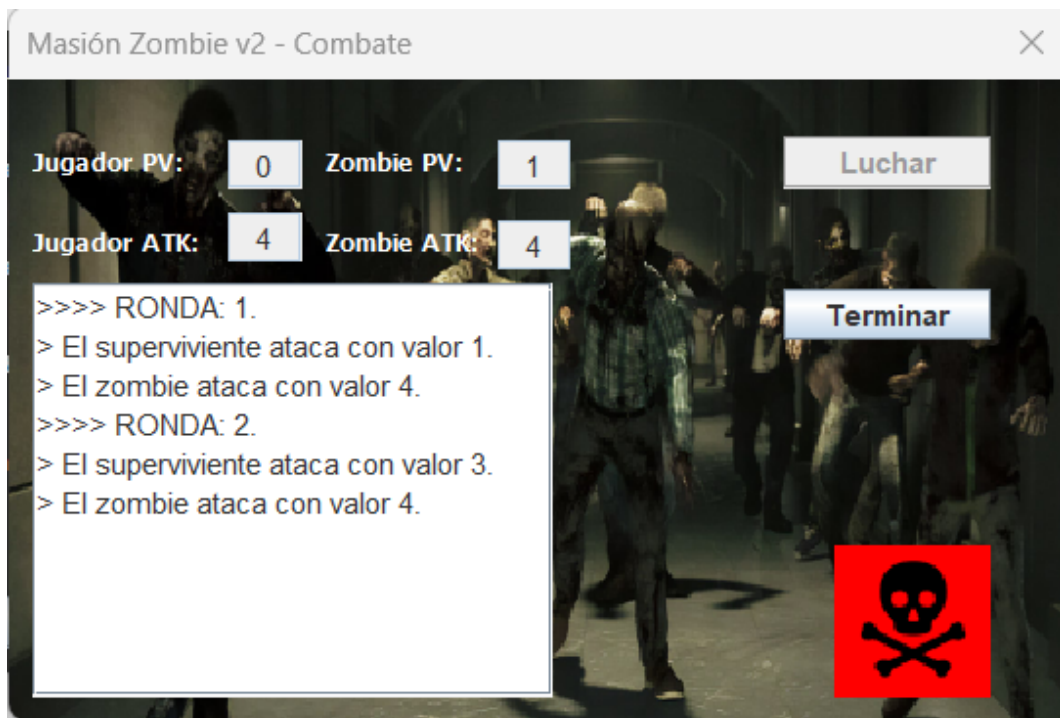


Para simular los turnos de combate deberemos pulsar el botón de “Luchar”, momento en el cual se llevará una ronda de combate completa que haga lo descrito más arriba.

En caso de que el superviviente gane, se notificará a través de un icono de victoria:



En caso de ser derrotado, el icono cambiará:



Si el **zombie** muere, nos tendremos que asegurar de reducir el número de **zombies** que hay en la habitación.

Por supuesto, todo el daño recibido tras el combate se debe mantener y reflejarse en la ventana de acciones.

Buscar por la habitación

Cuando ya no hay **zombies** en la habitación, es un buen momento para buscar algo que de ventaja al **superviviente**. Sin embargo, no todo va a ser tan sencillo, al otro lado de la puerta hay unos seres hambrientos que están esperando a escuchar el mínimo ruido para salir corriendo en busca de su comida favorita.

Para realizar la búsqueda lanzamos un **dado** de 100 y según el resultado obtendremos unos resultados u otros:

- 1-75: Hacemos ruido (ver resultados de ruido).
- 76-90: Encontramos un botiquín.
- 91-95: Encontramos una protección (+1).
- 96-100: Encontramos un arma (+1).

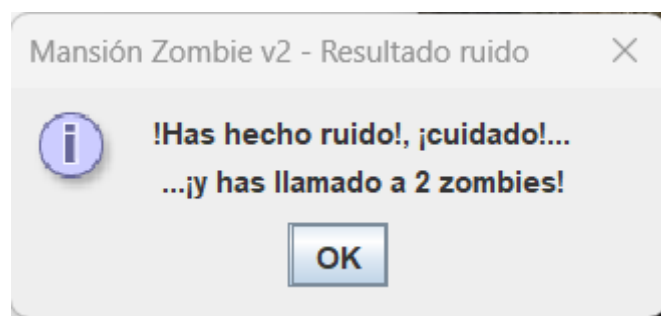
Las armas y las protecciones se van acumulando; los botiquines no, un **superviviente** o tiene botiquín o no lo tiene, y en caso de tener ya uno y encontrar otro no podrá cogerlo.

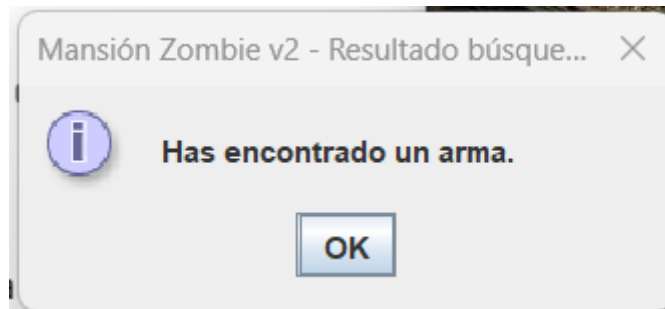
Como podemos ver, sólo hay un 25% de obtener algo que merezca la pena, pero en el riesgo está el premio. En caso de hacer ruido, tendremos que lanzar otro **dado** de 100 y ver los siguientes resultados:

- 1-40: No pasa nada.
- 41-80: +1 **zombie** en la habitación.
- 81-100: +2 **zombies** en la habitación.

Independientemente de los resultados obtenidos, siempre nos tendremos que asegurar de reducir el número de intentos de búsqueda que le quedan al **superviviente** en la habitación actual.

Al clicar en el botón buscar, el resultado de dicha búsqueda se notificará de la manera correspondiente y se deberá actualizar el estado de la ventana en consecuencia.





Avanzar a otra habitación / Salir de la mansión

Esta acción se da una vez que hemos terminado con los **zombies** de la habitación. No es obligatorio gastar las búsquedas.

Al pasar de habitación hay que tener en cuenta que la siguiente habitación debe prepararse para que todo esté disponible desde cero:

- Siempre hay un **zombie**.
- Se pueden llevar a cabo 3 búsquedas.
- El número de habitación en la que estamos avanza.

Un tema importante de cara a la experiencia de juego es detectar cuándo estamos en la última habitación. En este caso iremos mostrando el número de habitación de alguna forma, junto al número máximo de habitaciones que tiene la partida:



Muchas veces un detalle de este tipo puede suponer la diferencia entre que el **superviviente** tome la decisión correcta o no.

Curarse

Esta acción se da una vez que hemos terminado con los **zombies** de la habitación y tenemos algún botiquín disponible.

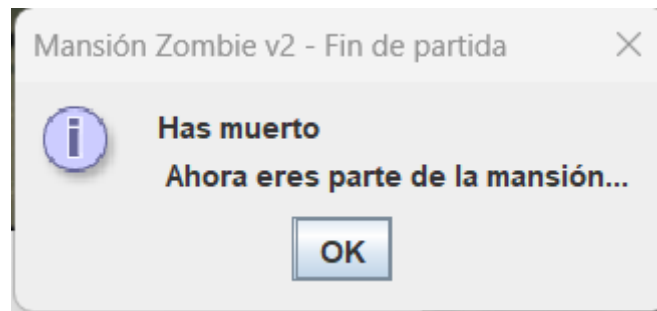
Un botiquín cura 4 puntos de vida al **superviviente**.

Se debe dar la opción de curarse al **superviviente** incluso aunque esté con la vida máxima (el sabrá lo que hace...). Hay que controlar que la vida máxima no supere la que se especifica para el **superviviente**.

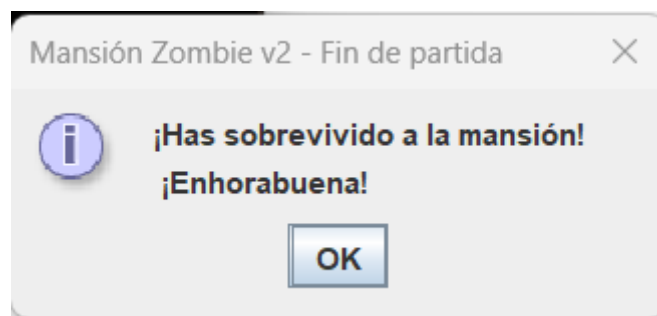
Fin del juego

En cada turno siempre se deben hacer las comprobaciones finales de partida que serán las siguientes:

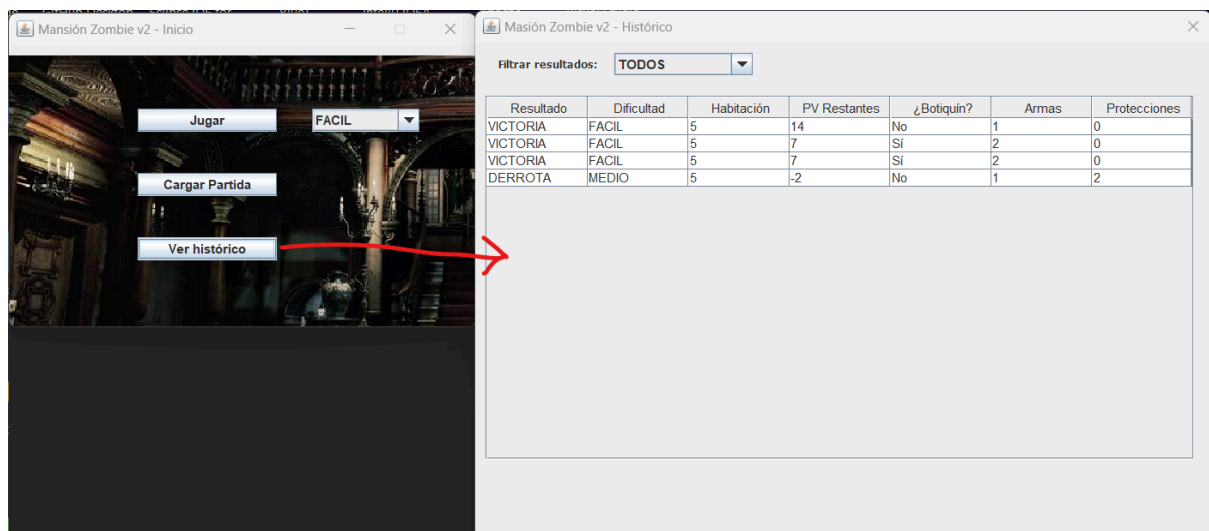
- El superviviente ha perdido todos sus puntos de vida -> GAMER OVER.



- El superviviente sale de la mansión -> VICTORIA.



Será en este punto donde se guardará de forma automática el resultado de la partida en un **archivo de texto**, que será luego el archivo que se utilice para cargar la ventana del histórico desde la ventana principal:



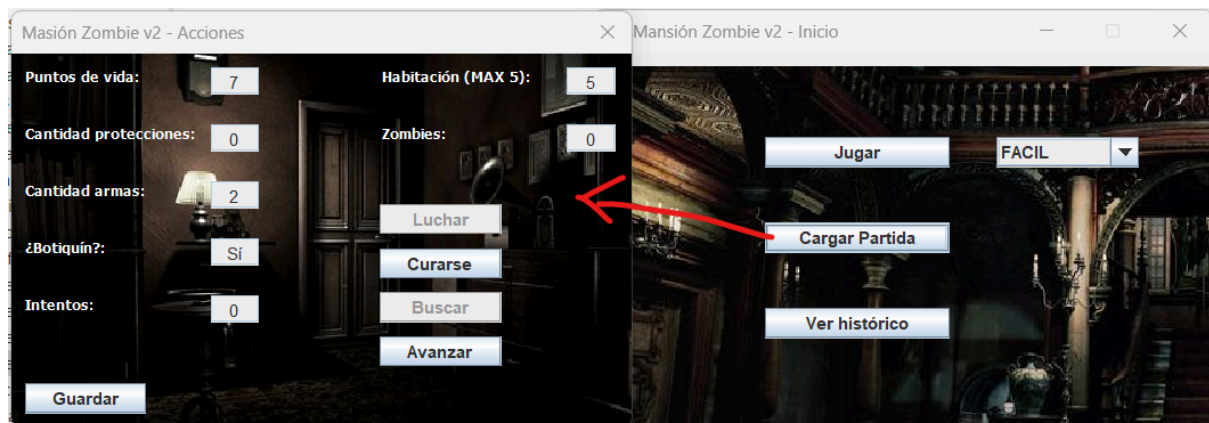
Guardar y cargar partida

Mientras estemos jugando, desde la ventana de acciones tendremos siempre disponible en alguna parte de la pantalla un botón para guardar la partida:

Al clicar, este botón guardará el estado completo de la partida en el punto en que estamos. Esto involucra la siguiente información a tener en cuenta para ser guardada:

- Dificultad de la partida.
- Habitación actual.
- Intentos restantes que nos quedan en la habitación actual.
- Número de zombis activos que quedan en la habitación
- Datos del Jugador:
 - Vidas
 - Armas
 - Botiquín
 - Protecciones

Toda esta información debe ser guardada en un **archivo binario**. En la ventana principal si clicamos en el botón de “Cargar” recuperará los datos de la partida que tengamos y automáticamente nos abrirá la ventana de acciones con todos los parámetros correctamente establecidos:



Implementación

De cara a la implementación que debemos hacer tenemos que tener en cuenta los siguientes puntos:

- Tener controlado el código. Si se piden números, controlar que no se pueda poner otra cosa y avisar correctamente. Si nos dan a elegir entre varias opciones, que el valor elegido esté entre esas opciones y si no también se indique de forma correcta.
- Tener el código bien estructurado. Habrá que leer el enunciado e identificar las clases que se van a necesitar. Cada clase tendrá que tener los constructores que se necesiten, getters, setters y resto de métodos que nos sirvan para que la jugabilidad sea lo más sencilla posible.
- Ver cual es la mejor elección de los tipos de datos de los atributos y si vamos a necesitar definir alguno constante.

- Crear los métodos correctos e intentar que todo esté organizado de la forma más lógica (sabiendo que no hay dos códigos iguales).
- Ver si puede ser bueno el uso de tipos enumerados personalizados para tener controlados ciertos estados de la partida o resultados de las acciones.
- Llamar a cada cosa con un nombre claro que defina para qué se usa o que hace, usando notación *CamelCase*, por si alguien tiene que trabajar con nuestro código lo pueda entender perfectamente.
- Hacer para cada clase sus pruebas pertinentes para ver que funcionan todos los métodos que hemos implementado. Hay que probar cada cosa por separado para que luego cuando las juntemos tengamos el menor número de problemas posible.
- Usar el depurador para ver que el código se comporta de la manera que esperamos y que sigue los caminos establecidos para cada una de las pruebas.

Especificación de la entrega

El ejercicio se entregará en formato ZIP con el siguiente nombre:

ACTIVIDAD5_PROG_MANSIONZV2.zip

El .zip debe contener:

- El proyecto completo de la mansión zombie usando la librería Swing de Java, la **documentación Javadoc** de todas las clases, atributos y métodos y los ficheros java que hagan funcionar la actividad.
- Un **documento .pdf anónimo** con nombre ACTIVIDAD5_PROG_MANSIONZV2.pdf que debe tener las páginas numeradas y cumplir con la siguiente estructura:
 - Portada: Título del trabajo, Fecha.
 - Páginas numeradas.
 - Índice (navegable).
 - Introducción: explicando brevemente el trabajo que se va a entregar.
 - Manual técnico: explicaciones de cómo y por qué se ha desarrollado el software de esta manera; cambios en decisiones de diseño, implementación, etc...
- Reflexión final:
 - ¿Qué me ha parecido y qué me ha aportado la actividad?
 - Dificultades y retos encontrados.
 - Alguna propuesta de mejora de la misma.

IMPORTANTE

No se admitirán entregas más allá del día y hora estipulados como fecha de entrega final. Sólo se admitirán aquellos trabajos cuya entrega se haya realizado a través de la plataforma online Blackboard, por lo que **se recomienda ser previsores y no dejar la entrega para última hora.**

