

Práctica 3:

Implementación de la funcionalidad del sistema, dado su diseño dinámico

Desarrollo de esta práctica

En esta práctica se implementan los métodos que faltan finalizando la implementación de la primera versión del juego. Para ello deberás utilizar los diagramas de secuencia suministrados

Por otro lado, se añadirán los elementos necesarios para poder jugar al juego de forma rudimentaria desde la consola.

A pesar de haber probado todos los métodos implementados en esta práctica y en las anteriores, tal como se recomendaba en la práctica 1 para todas las prácticas; es posible que, jugando partidas, te des cuenta de nuevos errores en el código. En estos casos se debe crear un pequeño programa principal de prueba que *force* la situación que ha provocado el error como ayuda a localizarlo y comprobación de que se ha corregido. Debido a la aleatoriedad del juego será muy poco probable que se vuelva a dar el mismo error en un tiempo razonable.

Implementando el resto de métodos

Para la implementación de los métodos que faltan se proporcionan diagramas de secuencia. La implementación se debe ajustar a lo especificado en dichos diagramas. Forma parte del aprendizaje, y de la evaluación, saber interpretar los diagramas y traducirlos a código.

Consideraciones adicionales relativas a la implementación

En algunos métodos es necesario eliminar elementos de objetos contenedores durante el recorrido de los mismos dependiendo de una condición. No todas las formas de recorrer contenedores permite realizar este proceso directamente. Investiga la forma correcta de hacerlo en cada lenguaje y pregunta a tu profesor/a si tienes dudas al respecto.

Investiga también si los contenedores que utilizas disponen ya de métodos que permitan conseguir con una única llamada el efecto indicado en el párrafo anterior.

Añadiendo una vista textual y un controlador

Como material asociado a esta práctica se incluyen además las siguientes clases:

- *TextUI*: una clase que hará el papel de vista. La vista será la única clase que interactúe con el jugador humano mostrando en la consola el estado del juego y recibiendo sus instrucciones sobre la dirección en que desea que se desplace el personaje. Esta clase incluye ya implementado un método para leer de forma robusta de la consola la dirección de movimiento. En Ruby se utilizarán los cursores y no será necesario pulsar Enter/Intro para confirmar la entrada de información y en Java las letras *w,a,s,d* habituales en muchos juegos y en este caso sí será necesario confirmar con Enter/Intro la entrada de la dirección. Solo tendrás que encargarte de implementar el método *showGame*.

El método recibe una instancia de un objeto con el estado completo del juego, representado casi en su totalidad mediante cadenas de caracteres y solo tendrás que plasmar esa información en la consola. Este método será el llamado cada vez que se muestre el estado del juego al jugador humano.

- *Controller*: esta clase se encarga de coordinar el juego en sí (una instancia de la clase *Game*) y la vista.

Para poder jugar solo necesitarás crear un programa principal donde se cree una vista, un juego y finalmente un controlador e indicar al controlador que comience la partida.

Para mantener separadas estas nuevas clases del resto, haz que la vista, el controlador y el programa principal estén cada una en un paquete Java distinto. En Ruby crea una carpeta y un módulo para cada una de ellas.

Para finalizar

Ya se ha completado el juego. Solo queda crear pequeños programas principales para probar los últimos métodos implementados en esta práctica en búsqueda de posibles errores y ya se puede pasar a jugar unas partidas.

Comprobando lo aprendido hasta ahora

Una vez finalizada esta práctica y las anteriores deberías saber y entender los siguientes conceptos:

- Los indicados en el guion anterior.
- Saber interpretar e implementar un diagrama de secuencia.
- Saber diseñar e implementar pequeños programas para probar métodos concretos.
- Saber localizar un error y corregirlo. Saber usar el depurador como ayuda a la localización del error.