



Apellidos:

Nombre:

Grupo: PL-

Equipo:

CRC-32:

--	--	--	--	--	--	--	--	--	--

Juego: El Inmortal

Material inicial (disponible en la carpeta TPP del escritorio)

Al inicio del examen el estudiante dispondrá de la solución completa (casi completa) del juego desarrollado durante las prácticas de la asignatura. Se han dejado tres soluciones (en ZIPs) que se corresponden con el criterio desarrollado por cada profesor durante dichas clases. Aun así, TODAS las soluciones funcionan, se basan en la misma solución al problema, y solo cambian en matices. Se detallan a continuación:

alex.zip: Dejado por el profesor Alejandro y cuya solución utiliza semáforos para garantizar la exclusión mutua. Además, se utiliza siempre la clase *Coordenadas* para indicar posiciones en el tablero.

oscar.zip: Dejado por el profesor Óscar y cuya solución que utiliza regiones sincronizadas para garantizar la exclusión mutua, siguiendo la solución explicada en clase para los grupos PL6 y PL7. Sólo se utiliza la clase *Coordenadas* para el retorno del evento *posicion()*, en el avatar la posición se mantiene mediante un par de enteros: *fila* y *columna*.

pedro.zip: Dejado por el profesor Pedro y cuya solución es similar a la previa. Sólo se diferencia en que se declara una matriz de objetos para garantizar la exclusión mutua en lugar de utilizar directamente la matriz *campoBatalla*.

Adicionalmente, en la carpeta TPP están disponibles:

- Las APIs de Java 8 en formato HTML. Para acceder a la documentación utiliza el archivo *apis.html*
- La biblioteca utilizada en las prácticas de la asignatura, *POE_library.jar*.
- La documentación HTML de la biblioteca. Para acceder a ésta utiliza el archivo *POE_library.html*.

Ejercicios solicitados para el examen

Ejercicio 1 [1,5 puntos]

Como se dijo más arriba, la solución está casi completa (aunque funcional y sin error alguno) pero le falta añadir el código que se considere necesario para tratar el evento de que un cliente se desconecte. Se ha de tener en cuenta que el cliente se puede desconectar con normalidad, al final de la partida, o bien a la mitad de esta. Por lo que habrá que contemplar ambas situaciones. En el último caso (desconexión a mitad de la partida) la partida deberá seguir normalmente, pero considerando que el jugador la ha abandonado.

Ejercicio 2

Se deberá añadir al juego la funcionalidad de que un jugador pueda teletransportarse a cualquier casilla del tablero al azar y que no esté ocupada por ningún otro jugador. La operación solo se podrá llamar un máximo de *MAX_TELETRANSPORTES* veces durante la partida. Para ello se deberá:

- a) **[2 puntos]** Añadir la constante anterior al código del servidor (junto a las constantes ya existentes) y una nueva operación a la clase servicio:

void teletransporte () throws TeletransporteNoValido;

Esta operación teletransportará al avatar a una casilla del tablero elegida al azar. En caso de superar el máximo de teletransportaciones permitido lanzará la excepción *TeletransporteNoValido*.

- b) **[3 puntos]** Añadir el código necesario en el cliente para que el jugador puede utilizar la nueva operación.

Ejercicio 3 [3,5 puntos]

Se deberá añadir al juego la funcionalidad de que el cliente (el jugador) pueda conocer los nombres (alias) de los jugadores que aún siguen vivos. **En ningún caso podrá conocer (llegar al cliente) ningún dato más de los avatares que no sea el nombre (alias).**

El jugador deberá ver en cada visualización del menú (junto con sus estadísticas) la lista de jugadores vivos, pero, además, si un jugador muere, el cliente deberá detectar dicha situación e indicarlo al usuario con el siguiente mensaje en pantalla (antes o después de sus estadísticas):

“El jugador XXXX ¡HA MUERTO! ¡SOLO PUEDE QUEDAR UNO!”

Añádase el código que se considere necesario para que el juego tenga esta nueva funcionalidad. El estudiante deberá decidir el nombre y especificaciones de todo aquello que se pueda necesitar (nuevas funciones, nuevas constantes, etc).

La solución que se entregue deberá seguir el patrón de programación orientada a eventos de forma absolutamente correcta y coherente con lo que ya está implementado.