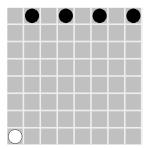
Tecnología de la Programación - Curso 2015/2016

Grado en Ingeniería Informática/Computadores/Software, Doble Grado en Matemáticas-Informática Examen de Junio (02/06/2016) - Duración: 3 horas

Puntuación máxima: 10 puntos (5 puntos sobre 10 en la calificación de la asignatura)

Enunciado

PARTE 1 (3 puntos) Se quiere añadir al programa desarrollado durante el curso el juego Fox and Hounds ($El\ zorro\ y\ los\ sabuesos$) que se juega entre dos jugadores: el primero representa al zorro y el segundo a los perros que pretenden darle caza. Consiste en un tablero 8×8 en el que inicialmente está situada una ficha que representa al zorro en la esquina inferior izquierda y cuatro fichas que representan los perros en la fila superior del tablero, como se muestra en la siguiente figura:



Las reglas del juego son las siguientes:

- Inicio del juego. El jugador que inicia el juego debe ser elegido de forma aleatoria.
- Movimientos. En cada turno, el jugador puede mover una ficha de su color a una casilla adyacente libre que esté situada en diagonal respecto a la casilla de origen. Los movimientos son distintos dependiendo del jugador:
 - El zorro se puede mover a cualquiera de las cuatro casillas adyacentes en diagonal, siempre que estén libres.
 - Los perros se pueden mover solamente a las casillas adyacentes en diagonal de la fila inmediatamente inferior. Cuando un perro alcanza la fila inferior del tablero, ya no puede moverse.
- Siguiente jugador. Los dos jugadores se van alternando siempre que puedan mover. Si un jugador no puede realizar ningún movimiento, termina el juego.
- Fin del juego. Se pueden producir los siguientes finales de juego:
 - El zorro es atrapado (pierde) si no puede moverse a ninguna casilla libre.
 - El zorro escapa (gana) si los perros no pueden mover (y el zorro sí puede mover).
 - En este juego no se puede producir la situación de empate.

Añade un paquete foxHounds dentro de practica4 con las clases necesarias para que se pueda jugar a este juego en modo consola, además de los juegos ya disponibles en la práctica. Puedes suponer que el zorro se corresponde siempre con la primera ficha de la lista de fichas. Modifica examenJunio.Main para poder iniciar el programa jugando a Fox and Hounds usando la opción "-g fh" desde la línea de comandos.

PARTE 2 (1,5 puntos) Añade un paquete foxHounds en practica5 con las clases necesarias para que se pueda jugar a este juego en el interfaz de ventana. Recuerda que el juego debe funcionar también en modo cliente-servidor.

PARTE 3 (2,5 puntos) Se quiere implementar un nuevo botón denominado pulse para producir un pulso electromagnético. La funcionalidad de este botón consiste en lo siguiente: (i) debe elegir una casilla del tablero de forma aleatoria que contenga una ficha del jugador que tiene el turno (la denominamos origen); (ii) todas las fichas que no sean obstáculos y estén situadas a una distancia del origen menor o igual a d deben moverse en diagonal a una casilla adyacente libre, que se debe elegir de forma aleatoria. Las celdas se pueden recorrer en el orden que quieras.

Modifica la vista de ventana de practica5 añadiendo un combo-box con el rango d del pulso (valores de 1 a 5) y el botón pulse, que puede utilizar el jugador que tenga el turno en lugar de un movimiento normal. En modo multiventana o cliente-servidor este botón solo debe estar habilitado cuando el jugador tiene el turno. Recuerda que el botón pulse debe funcionar correctamente también en modo cliente-servidor.

PARTE 4 (3 puntos) Se quiere modificar la funcionalidad del servidor de practica6 de manera que, de forma aleatoria, ignore peticiones de movimiento de los clientes y les notifique que deben realizar de nuevo el movimiento debido a *ruido* en la conexión. Para implementar esta funcionalidad debes hacer lo siguiente:

- Añade un combo-box en la ventana del servidor con valores de 0 a 9 para fijar el nivel de ruido, que denominaremos noiseLevel.
- Modifica el servidor de juego de forma que, cuando reciba una solicitud para realizar un movimiento, si el valor de Utils.randomInt(10) es menor a noiseLevel, en lugar de hacer el movimiento lo ignore y notifique a todos los clientes que se ha detectado ruido.
- Modifica la vista de Swing para que, cuando se reciba esta notificación, el cliente que solicitó el movimiento escriba un mensaje en el área "Status Messages" y haga un nuevo movimiento (utilizando el modo actual: manual, random, etc.). Los demás clientes deben ignorar esta notificación.

Para implementar el mecanismo de notificación, primero debes añadir el método public void onNoise() al interfaz GameObserver, y modificar todas las clases que lo implementan (tanto en basecode como en tus prácticas) para incluir ese método vacío. Después, modifica los métodos de las vistas de Swing para que tengan el funcionamiento descrito. Modifica practica6 para que el servidor pueda enviar estas notificaciones a los clientes.

Recuerda que los atributos de la clase GameServer deben declararse volatile.