

ANÁLISIS Y DISEÑO JUGADOR BARQUITOS

SISTEMAS MULTIAGENTES



25
AÑOS

Universidad de Jaén

Manuel Gallego Chinchilla

INTRODUCCIÓN

Descripción del problema

Estamos haciendo un juego donde intervienen varios agentes. Un agente central, un agente grupo, un agente tablero y diferentes agentes jugador. Yo voy a tratar de mi agente jugador correspondiente, que sólo va a saber jugar a los barquitos.

Objetivos para la solución del problema

Debo lograr hacer un jugador que juegue sin trampas a los barquitos, y que intente ganar la partida.

ANÁLISIS

Agente Jugador barquitos

Este agente será el encargado de jugar las partidas de barquitos que nos proponga el agente central de juego.

Será diseñado con la finalidad de ser mejor jugador que el resto de participantes y así poder ganar el máximo número de partidas individuales y torneos.

Tareas que realizará

- * Decidir si juega una partida a la que se le ha propuesto jugar.
- * Responder con la decisión tomada.
- * Decidir donde poner inicialmente los barcos en el tablero (solo si ha aceptado jugar).
- * Llevar actualizada la información del tablero.
- * Evaluar la información más actualizada del tablero y decidir que movimiento es más conveniente hacer. (Cuando le toque el turno)
- * Procesar el resultado de nuestro movimiento, para saber si ha sido tocado, hundido o agua. También en esta tarea podemos saber si hemos ganado la partida ya que llevaremos una cuenta de los barcos que le quedan al rival.

Ontologías

El uso de ontologías nos ayuda a la hora de enviar mensajes entre los protocolos de comunicación, gracias a ello contamos con una serie de conceptos para la comunicación, en nuestro caso los conceptos han sido implementados como clases.

Viendo las tareas que debemos realizar, y los diagramas de comunicación proporcionado por el profesorado. Podemos hablar de dos tipos de conceptos dentro de la ontología, uno referente al uso interno del agente de los barquitos, y otro tipo de conceptos referentes al uso de la comunicación con el jugador.

Voy a describirlos de manera separada, en este caso, ya que nos encontramos en el análisis, voy a ir explicando por encima los que usaremos. Ya en la parte de diseño me centraré más en la implementación dentro de nuestro proyecto.

* Conceptos referentes al uso interno de la clase:

- * Hay que saber representar los diferentes tipos de barcos.
- * El número de barcos de cada tipo que tenemos cada jugador.
- * Si se quiere rechazar una oferta de partida, el jugador debe saber internamente la causa.

* Conceptos referentes a la comunicación del jugador con otros agentes:

- * Se deben colocar barcos al iniciar la partida.
- * Se debe saber el estado de la partida.
- * Hay que saber como representar el movimiento una vez le toque el turno.
- * Representación de la colocación de los barcos.
- * Un último concepto referente al resultado del movimiento realizado.

Todos estos puntos se convertirán en conceptos, en la fase de diseño.

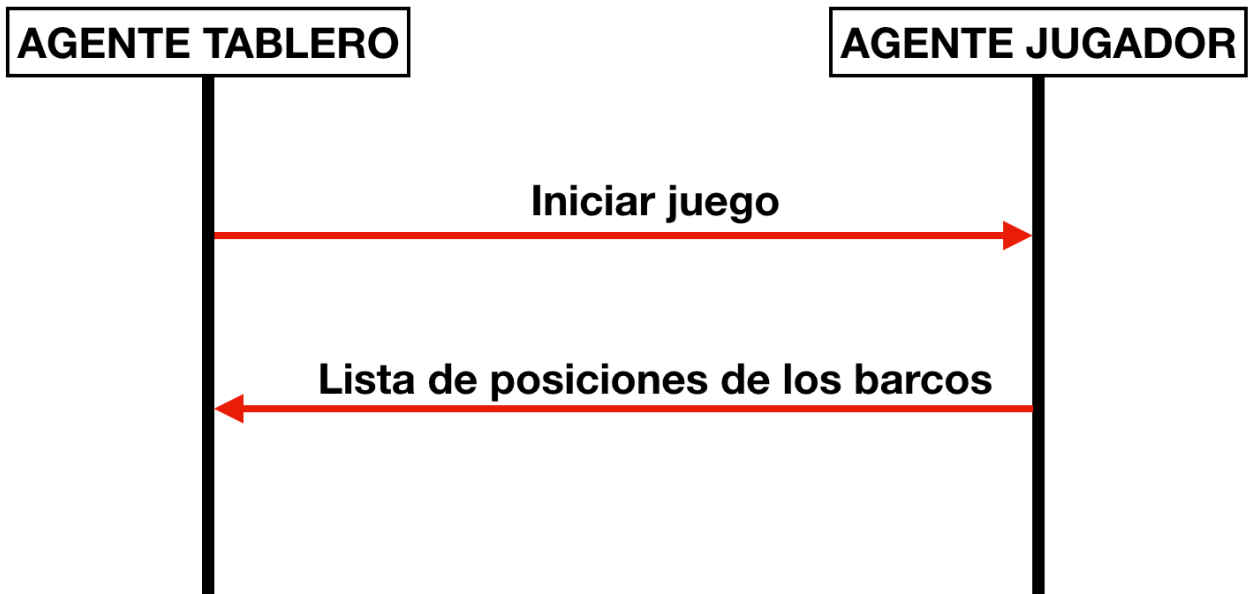
Comunicación entre agentes

En el apartado de diseño me centraré más en el uso de protocolos para la realización de las comunicaciones que voy a describir ahora.

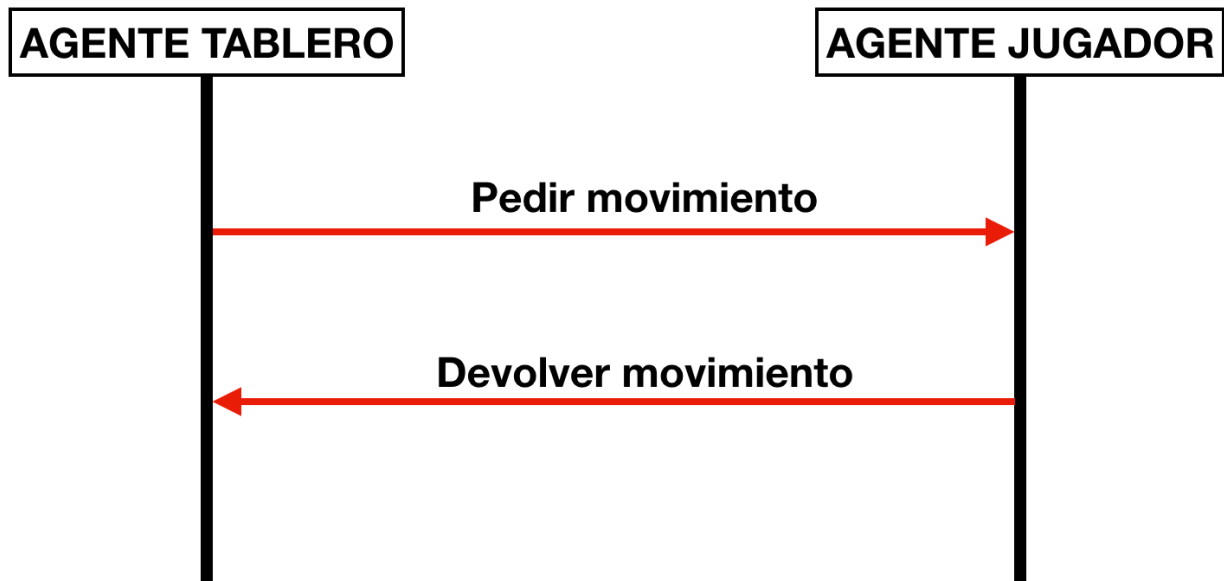
* PROPONER PARTIDA: Mi agente jugador debe responder al mensaje que le ha llegado por parte del agente central de juegos de proponer partida, en donde el agente central le propone una partida para jugar y mi agente debe responder si está interesado en jugarla o no.

* INICIAR JUEGO: El agente tablero encargado de llevar la partida, nos mandará un mensaje para que le respondamos con la posición inicial de los barquitos. En esas posiciones que le mandemos, el agente tablero será el encargado de colocarlos. Nosotros sólo deberemos decidir que posiciones iniciales nos interesa colocar los barquitos.

* PEDIR MOVIMIENTO: El agente tablero nos pedirá un movimiento en el momento que nos toque el turno de juego. En ese momento debemos evaluar como va la partida y ver cuál sería nuestro mejor movimiento para acercarnos a la victoria. Los primeros movimientos de la partida serán



posiciones aleatorias hasta que toquemos un barquito rival, ya que al principio no tenemos ningún conocimiento de la posición de los barcos rivales.



A la hora de devolver el movimiento, tenemos que responder de todos modos, es decir, si es mi turno devolveré el movimiento y si no es mi turno devolveré un mensaje predeterminado. Esto lo hacemos para saber que no se ha caído ninguno de los jugadores implicados en la partida.

DISEÑO

Agente jugador barquitos

Estructuras de datos usadas

- * Matriz de dimensiones del tablero por cada partida: Guardaré las posiciones que ya he consultado del tablero rival y su resultado. Gracias a esta matriz podré ir almacenando el transcurso de la partida en cuanto al tablero rival, para que así sea más fácil la evaluación del próximo movimiento a realizar.
- * ArrayList de partidas: Como el jugador podrá jugar a más de una partida a la vez, se deben almacenar todas las partidas aún activas para que el jugador sepa en todo momento en la partida en la que se encuentra y pueda hacer movimientos disponibles (Para así no repetir un movimiento que ya había dicho antes).
- * Entero donde almacenará el número de partida activas a las que está jugando.
- * Entero donde almacenará el número máximo de partidas que va a poder jugar al mismo tiempo.

Diseño de las tareas de los agentes

- * **Decidir si juega una partida a la que se le ha propuesto jugar:** Si el número de partidas activas no supera el máximo de activas aceptará la nueva partida, en caso contrario la rechazará.
- * **Responder con la decisión tomada.**
- * **Decidir donde poner inicialmente los barcos en el tablero (solo si ha aceptado jugar):** Esta tarea cuenta con una restricción, que es la de que alrededor de un barco debe haber solo agua, es decir, que dos barcos no pueden ser adyacentes.

Para resolver esto y que mis barcos sean puestos de manera totalmente aleatoria para así no repetir posiciones entre partidas, lo que he hecho ha sido lo siguiente:

1.- Compruebo la longitud del barco que quiero poner, y hago un aleatorio para sacar una coordenada X y la coordenada Y.

2.- Lo que hago es comprobar si puedo poner ese barco, en la posición inicial X,Y en orientación vertical u horizontal.

Para realizar esta comprobación voy comprobando las coordenadas en las que estaría dicho barco, y si alguna coordenada adyacente, ya cuenta con otro barco, pues devuelvo FALSE, y genero otro par X, Y.

3.- Una vez que he comprobado que si he podido poner barco en la posición deseada, paso a la colocación del siguiente barco.

- * **Llevar actualizada la información del tablero:** Tras la realización del movimiento y recibir la información del resultado del mismo, guardaremos en la matriz correspondiente de la partida que usamos de ayuda para su transcurso, si el movimiento ha sido agua, tocado o hundido.
- * **Evaluar la información más actualizada del tablero y decidir que movimiento es más conveniente hacer:** Como el estado actual del tablero rival, lo tengo guardado en una matriz, lo que hago es evaluar dicha matriz, buscando si tengo alguna posición en la que el resultado haya sido un TOCADO.
- * Si esto es así, y cuento con alguna posición del rival en la que he conseguido un tocado, hago un aleatorio en el que como resultado me dice si mi movimiento va a ser o la casilla de arriba del tocado, o la de abajo, o la de la izquierda, o la de la derecha.
- * En caso que no tenga ningún tocado aún, lo que hago es un movimiento aleatorio.
- * **Procesar el resultado de nuestro movimiento,** para saber si ha sido tocado, hundido o agua. También en esta tarea podemos saber si hemos ganado la partida ya que llevaremos una cuenta de los barcos que le quedan al rival.

Ontologías

En esta parte de las ontologías voy a explicar como hemos implementado todos los conceptos que necesita nuestro agente jugador.

Siguiendo con lo que he puesto en la parte de análisis, tenemos un concepto implementado por cada una de las características que he explicado. Dicho esto voy a relacionar esas características con las clases ofrecidas por Pedro.

- * Hay que saber representar los diferentes tipos de barcos: **TipoBarco.**
- * El número de barcos de cada tipo que tenemos cada jugador: **JuegoBarcos.**

- * Si se quiere rechazar una oferta de partida, el jugador debe saber internamente la causa: **Motivación**.
- * Se deben colocar barcos al iniciar la partida: **ColocarBarcos**.
- * Se debe saber el estado de la partida: **EstadoJuego**.
- * Hay que saber como representar el movimiento una vez le toque el turno: **MovimientoEntregado**.
- * Representación de la colocación de los barcos: **PosiciónBarcos**.
- * Un último concepto referente al resultado del movimiento realizado: **ResultadoMovimiento**.

Todos estos conceptos han sido implementados como clases java implementado el jade.content.Concept.

Para poder cargar todo el código de Ontología hemos modificado el fichero POM, para que se conectarán las dependencias con el código de Pedro.

Comunicación entre agentes

- * PROPONER PARTIDA: El agente central de juegos debe localizar a jugadores que estén dispuestos a participar en una partida, ya sea individual o torneo. Para esta comunicación utilizaremos un protocolo Propose.
- * INICIAR JUEGO: El agente encargado de llevar la partida le envía un mensaje a los jugadores de la misma, para saber donde van a colocar inicialmente los barcos. Para esta comunicación utilizaremos otro Propose.
- * PEDIR MOVIMIENTO: El agente tablero nos pedirá un movimiento en el momento que nos toque el turno de juego. Nosotros debemos responder, ya sea nuestro turno o no, ya que si no respondemos e porque hemos abandonado. Tras dar nuestra respuesta nos llegará un mensaje del resultado del movimiento realizado.

