Universidad Carlos III

Grado en Ingeniería Informática

Inteligencia Artificial en las organizaciones

Curso 2021-22

Práctica Agentes de IAO

Agente participante

Ana Tian Villanueva Conde - 100405817@alumnos.uc3m.es

Índice

Introducción

Este documento refleja la implementación de los agente tablero y jugador a través de la plataforma JADEX. El tablero tendrá el objetivo de actuar como la organización del juego de mesa Junta interactuando con los diversos jugadores participantes de la partida y tratando sus peticiones de manera adecuada.

Comenzaremos por la explicación del xml generado, centrándonos en las creencias que hemos necesitado y la asociación de los planes desarrollados con estas últimas y los mensajes relevantes para activar y desarrollar un conjunto de cinco protocolos.

Los planes los explicaré con más detalle en la sección siguiente de manera separada, y me centraré en cómo han sido implementados haciendo uso de la ontología previamente desarrollada y los mecanismos que ofrece JADEX.

Para terminar mencionaremos nuestras reflexiones y opiniones sobre la práctica y experiencia de desarrollar sobre la plataforma JADEX, así como posibles problemas que hayan surgido. Sin mayor demora comenzamos con la explicación del núcleo del agente tablero, el xml.

Explicación del XML

Se deben justificar tanto la forma en la que se invocan los planes y adoptan los objetivos como las creencias que se piensa que habrá de utilizar la organización y por último la forma en la que el agente organización hará uso del DF.

El agente participante adopta el objetivo de registrarse en el DF e interactúa con el tablero enviando y recibiendo mensajes y va decidiendo su estrategia según le convenga una acción u otra.

Estos mensajes tienen asociados un plan, el cual intentará cubrir todo el protocolo, y esperan una respuesta por parte del tablero. También tienen asociados unas creencias, que es lo que cree el jugador sobre sí mismo, o sobre el resto de jugadores, aunque esto no tiene porque ser la realidad, pues únicamente el tablero es quien almacena esta información. Las creencias son:

- myself: información sobre el propio jugador.
- miTablero: información sobre el tablero, utilizado únicamente para saber si existe un tablero al que se pueda unir.
- Jugadores: creencia sobre el resto de jugadores, importante para desarrollar una estrategia de juego.
- localizaciones: almacena las localizaciones que ya ha probado al intentar asesinar y no volver a repetir una localización.
- Cartas: almacena las cartas que tiene el jugador en mano, si tiene que utilizar una carta de su mazo, entonces la elimina de sus creencias.
- playerWinning: jugador que va ganando por el momento.
- Enemigo: lista con los jugadores enemigos en toda la partida.

Implementación de los planes en Java

Explicaciones individuales de los ficheros java implementados

---

En esta sección voy a explicar los planes que he elegido y las diferentes situaciones para ejecutar dicho plan para el agente participante.

Unirse partida

Este es el primer protocolo en ejecutarse y el que da inicio a la partida. El jugador “myself” busca el tablero para unirse y manda un request para que éste acepte. El tablero comprueba si tiene menos de 7 jugadores y hace agree si es así, o refuse en caso contrario. Para simular esta situación me he creado un tablero con 6 jugadores y he simulado que un jugador más quiere entrar a dicho tablero.

Una vez mandado el request, el jugador espera la respuesta del tablero. Si el tablero envía un agree, entonces significa que el jugador puede unirse a la partida y actualizar la familia a la que pertenece. Por el contrario, si manda un refuse entonces el jugador no puede unirse a la partida.

En caso de que se pueda unir a la partida, el jugador recibe el mensaje con el jugador que se quiere unir a la partida y con una lista de jugadores que son los que ya se encuentran en la partida. Como había mencionado anteriormente, para realizar las pruebas he decidido probar con 6 jugadores, por tanto unirsePartida.getAlljug() devuelve una lista con estos jugadores ya inicializados en tablero y los actualiza en la base de creencias del jugador participante.

Esta réplica de la información se debe a que en algunos casos, el jugador participante no tiene la misma información que el tablero, ya sea por que el presidente ha propuesto un presupuesto y haya dado más dinero del que correspondía a otro jugador.

La gestión de los asesinatos

En esta sección voy a explicar las decisiones del jugador para realizar un request al tablero e intentar asesinar. Las decisiones del jugador se basan en las siguientes comprobaciones:

- Se encuentra en la fase de asesinato, pues el jugador sabe que sino su petición será rechazada por el tablero y no le interesa.
- El jugador tiene la carta de asesinato y para saber si tiene la carta, el jugador busca en las crencias AgentCard si se encuentra la carta de tipo 5, que corresponde a la carta de asesinato.

A partir de este punto comienzan las estrategias del jugador para ganar la partida. El jugador participante representa a un experto cuyo objetivo es tener la mayor cantidad de dinero posible. Como al asesinar a un jugador, el jugador asesino se queda con el dinero del otro jugador, mi jugador experto siempre va a intentar asesinar siempre que cumpla las condiciones mencionadas anteriormente.

Por tanto la estrategia que voy a seguir es completamente objetiva en este caso. En primer lugar comprueba aquellos jugadores que puede asesinar, pues si los jugadores se encuentran en el exilio o

aún no han elegido localización, entonces no interesa asesinar porque el tablero rechazará dicha petición. Una vez guardados los jugadores a los que se puede asesinar en la variable chooseTarget, el jugador buscará aquel que tenga más dinero en este momento (según la información que almacena en la creencia de jugadores, que puede no ser información real).

Finalmente en waitIntento de asesianto, el jugador sabe si se ha podido producir el asesinato, o en caso contrario si el tablero le ha rechazado el intento de asesinato. Si el tablero acepta el intento de asesianto, este devuelve el jugador objetivo que ahora se encuentra muerto y el propio jugador asesino que ahora tiene las cartas y el dinero del otro jugador. Por tanto, tengo que actualizar la base de creencias del propio jugador, eliminando las facts anteriores y añadiendo las actuales.

El golpe de estado

Al igual que anteriormente, comprobamos que el jugador se encuentra en la fase de golpe de estado y que el propio jugador no es el actual presidente. El golpe de estado consiste en mandar un mensaje de request al tablero para que este analice si se puede realizar o no el golpe de estado.

La decisión que toma el jugador para dar el golpe de estado se basa en obtener al presidente actual y comprobar que no el actual ganador y comprobar que el presidente es enemigo. Para comprobar que un presidente es enemigo utilizo el belief AgentEnemigo, que devuelve true si es enemigo y false en caso contrario. Envio el mensaje al tablero y espero su respuesta.

Una vez recibida la respuesta del tablero, si la acción recibida es darGolpe (la misma que envié pero con los nuevos valores, actualizo la creencia de mi tablero y la del propio jugador en el agente participante jugador. Por el contrario, si recibo imposibilidadGolpe, entonces significa que el tablero ha rechazado la petición y no se puede realizar el golpe de estado.

Conclusiones

Conclusiones técnicas, comentarios personales, críticas constructivas y problemas encontrados durante la realización de la práctica.

Esta práctica ha sido útil para entender mejor el funcionamiento de JADEX y el envío y recibo de mensajes entre los agentes, pues en la anterior práctica solo veíamos un lado (el tablero) y nos faltaba el agente participante.

Por otro lado, la dificultad de esta práctica ha sido que ejecutara y compilara, pues he tenido muchos problemas para ejecutar y una vez que he ejecutado con la idea que tenía en la cabeza, he obtenido muchos errores de sintaxis. Además, no estoy muy familiarizada con java, pues los últimos años se han centrado más en python y como en la anterior práctica no comprobamos que compilara, pues en esta he tenido que arreglar muchos errores.

Finalmente, un comentario sobre el video es que ha sido muy difícil, especialmente sintetizar todo lo que quería explicar en solo 5 min. Sin embargo, me ha ayudado a aclarar más aún mis ideas y al final ha sido de gran utilidad.