Entregable 2 SID
JADE
Q2-2022

Gotanegra Estañol, Miquel Montón Gimeno, Pablo Mas Baulas, Pasqual

Marzo 2022

1. Descripción del problema

Tenemos que desarrollar un agente, el cual, con un número arbitrario de estos en un sistema determinado, tiene que jugar con el resto de agentes en el sistema, al juego del prisionero, en su variante *turn-taking prisoner's dilemma,* que básicamente lo que hace es jugar al juego del prisionero de manera secuencial. El objetivo es conseguir la mínima suma de penalizaciones, en las distintas rondas.

2. Decisiones de diseño

El agente, persigue 4 *goals* distintos, que agrupamos de la siguiente manera:

De manera secuencial persigue *GoalRegister*, *GoalFindAvailablePlayers*, y después de manera paralela se persiguen los 2 *goals* restantes, que son el *GoalOfferPlay* y *GoalReceivePlay.*

Estos goals tienen sus PlanBody que es donde programamos su funcionalidad. Aquí es dónde se toman las decisiones de diseño del agente.

Los PlanBody de register y find available players se ejecutan sólo una vez, mientras que los otros dos nunca acaban como SUCCESFUL y por tanto se vuelven a ejecutar continuamente.

Beliefs

La base de conocimiento de el agente se basa en beliefs, el agente guarda información en los siguientes belief:

- **CC,CD,DC,DD:** Valores de las penalizaciones de las decisiones del agente.
- **Historial:** historial de movimientos del agente.
- **Score:** suma de la penalización del agente.
- **Jugadores pendientes:** jugadores disponibles para jugar que aún no han jugado con el prisionero

- **Lastmove:** guarda el último movimiento realizado contra cada uno de los agentes, para poder hacer el cómputo de la score.

Plan Register

Este plan es el encargado de registrar al agente en el DirectoryFacilitator, con type “player” para poder ser encontrado por el resto de agentes con este nombre. Es importante que al acabar de registrarse correctamente, para que no se intente perseguir posteriormente, lo cual provocaría un error FIPA, para eso indicamos como *SUCCESSFUL* el *goal*.

Plan FindAvailablePlayers

Como hemos dicho anteriormente, como todos los agentes, nada más ser creados persiguen el goal de registrarse, después puedes buscar todos los agentes disponibles, para así poder buscar a todos los prisioneros disponibles en el sistema. Estos se añaden como jugadores_pendientes, sin incluir a el agente mismo. Además después inicializamos el último movimiento de todos estos agentes encontrados, como nulo. Y entonces declaramos el *goal* como *SUCCESSFUL.*

Plan OfferPlay

Se selecciona un prisionero random de los jugadores pendientes del agente, y a partir de los datos de las penalizaciones de la gente, selecciona la opción más adecuada, y la envía al agente seleccionado, además se guarda el movimiento en last_move.

Plan ReceivePlay

Si el agente recibe un mensage válido, a partir de la jugada del agente que envía el mensaje, el agente coge la respuesta con una puntuación mínima. Entonces registramos la jugada, y a partir de esto actualizamos el score y el historial de movimientos del agente, y finalmente enviamos la respuesta.

3. Distribución del trabajo

Miquel: Código inicial y versión final de la funcionalidad del agente.
Pablo: Documentación y primera versión de la dinámica de juego del agente.
Pasqual: -