Memoria Entrega 1

Martín Fernández Cid Pablo Freire Gullón
David Fernández Vázquez Pablo Fernández Camino

03/04/2025

Contents

1 Comprensión del proyecto

Para la realización de este proyecto, el primer paso es, obviamente, comprender tanto los diferentes objetivos a alcanzar a lo largo de estas semanas como las herramientas que se nos han sido proporcionadas para la realización de dicho trabajo, ya que, sin un profundo entendimiento del problema a resolver, la complejidad del proyecto incrementará exponencialmente y las soluciones alcanzadas no serán todo lo buenas que podrían llegar a ser.
En el caso de nuestro equipo, esta comprensión se realizó leyendo los diferentes objetivos publicados en la plataforma Moovi para, posteriormente, analizar el código del esqueleto proporcionado (tanto el de Java como el de AgentSpeak), algo que ayudó a todos los

miembros del equipo a pensar en posibles soluciones para las diferentes tareas a realizar de forma concurrente al ya mencionado entendimiento del código fuente.

2 Organización de la solución

Como es común en cualquier proyecto organizado, antes de comenzar a trabajar para cumplir con los objetivos planteados, se debe definir el orden en el que se realizarán las tareas, la repartición de las mismas entre los diferentes integrantes del equipo y unos plazos para evitar la acumulación de trabajo y/o la creación de una solución deficiente debido a la falta de tiempo.
En base al análisis de las tareas a realizar y el esqueleto proporcionado comentado en el anterior apartado, nuestro grupo concluyó que obtendríamos un mejor resultado siguiendo el plan asociado a cada semana. Es decir, intentaríamos pensar e implementar una solución a cada uno de los objetivos de la semana 5 (siendo esta la primera semana del curso referente al proyecto) antes de pasar a los objetivos de la semana 6 y viceversa.
Este enfoque acabó sirviendo de guía durante todo el trabajo, y es por eso por lo que, en los siguientes apartados de esta memoria, se comentarán las soluciones alcanzadas siguiendo este orden.

3 Adaptación del código

Una vez explicado el proceso de análisis y comprensión del proyecto junto con la creación de un plan a seguir para la realización de este, cabe destacar un paso previo al diseño de la solución requerida que, aunque corto, fue de vital importancia. Y es que para poder comenzar a trabajar en los requisitos de esta primera entrega, era necesario realizar pequeños cambios y adaptaciones al esqueleto, como, por ejemplo, el cambio a nombres de métodos o variables relativos a un proyecto independiente al nuestro, de forma que todos los integrantes del equipo pudieran tener una mejor experiencia y ofrecer unos mejores resultados a la hora de realizar su trabajo.

4 Objetivos semana 5

Como hemos comentado anteriormente, los objetivos definidos bajo la sección de “Semana 5” en la plataforma Moovi fueron los primeros en ser realizados. A partir de este punto, se comentarán los problemas y soluciones de cada uno de los requisitos del proyecto, haciendo especial hincapié en el razonamiento detrás de la obtención de dicha solución.

4.1 Creación de zona de almacenamiento de medicamentos

Una de las tareas mas rápidas y sencillas de implementar y, a la vez, una de las más importantes es la creación de un elemento cuya función será almacenar las medicinas que, tanto el agente Robot como el agente Owner, tendrán que obtener para la realización de las diferentes acciones correspondientes (consumirla o transportarla entre otras).
Resulta obvio que los objetos predefinidos en el proyecto (la nevera y el lavavajillas), no son válidos para almacenar algo de tan suma importancia como son los medicamentos que necesita una persona, por lo que escogimos la solución más común para esta problemática

en la vida real: guardarlos en un pequeño armario.
Para implementar esta solución, fue necesario obtener un par de imágenes con el objetivo de representar dicho objeto cerrado (en caso de que ningún agente interactúe con él) y abierto (si alguien está cogiendo algún medicamento) además de revisar las partes del código referentes a la definición de objetos del entorno (como los anteriormente mencionados), lo cuál era la forma más sencilla y segura de añadir un nuevo elemento al entorno de la casa, ya que tan solo requeriría seguir la misma estructura del código ya existente.

4.2 Implementación del comportamiento del armario

Tal y como hemos especificado en el apartado anterior, el armario debe aparecer abierto o cerrado en función de si uno de los agentes está interactuando con él, y, además, debe mostrar el número y tipo de medicamentos almacenados.
Para que el armario se abra en caso de que uno de los agentes interactúe con él, los integrantes del equipo llegamos a una conclusión. Si uno de los dos agentes ha ido al armario es porque tiene que coger algo ahí, es decir, tiene que interactuar con él. Este razonamiento nos guió a la que sería nuestra solución y es que, siguiendo el código ya existente para el resto de objetos de la casa, bastará con hacer que la imagen que se nos muestra sea la del armario abierto en caso de que uno de los agentes esté en una casilla colindante, y que la imagen sea el armario cerrado en caso contrario.
Para mostrar dichos medicamentos que hay disponibles se ha utilizado un texto en las casillas colindantes al armario de las medicinas donde se mostrarán las dos primeras letras del nombre del medicamento como identificador seguidas de un número que será un indicador de cuantas medicinas de ese tipo quedan. Dicho número se actualizará dinámicamente gracias al uso de un método que devolverá en todo momento el número restante de un determinado medicamento, solucionando así el problema de mostrar los decrementos de las medicinas cada vez que un agente coge una del armario.

4.3 Correcto posicionamiento de los agentes en la zona

Uno de los cuatro objetivos relacionados con esta primera semana es que los agentes se posicionen uno al lado del otro en caso de estar ambos interactuando con el armario de los medicamentos. Debido a nuestra implementación del movimiento de los agentes y la definición de una posición válida para interactuar con los objetos del entorno, el primer agente en llegar se colocará al lado del armario (la casilla exacta dependerá de la posición de la que venga el agente, ya que siempre buscaremos el camino más corto), mientras que el segundo agente buscará otra posición para realizar dicha interacción.
Esto provocará que ambos agentes puedan estar al lado del armario a la vez, pero dichos agentes no estarán al lado el uno del otro.
Nuestro equipo ha intentado pensar en posibles soluciones para esta problemática, como, por ejemplo, que el primer agente en llegar le envíe su posición al otro agente y ajustar la posición de destino en base a esta o que el agente que llegue segundo tenga un plan para recibir información por parte del primero para así poder interrumpir su interacción con el armario y que se puedan colocar uno al lado del otro. Sin embargo, no hemos sido capaces de implementarlas correctamente, por lo que se mantiene el comportamiento definido en el primer párrafo de esta sección.

4.4 Algoritmo de movimiento

El diseño de un algoritmo de movimiento es uno de los elementos más importantes a desarrollar en todo el proyecto. Sin uno bien definido, los agentes podrían chocar con alguno de los obstáculos presentes en la casa o no obtener el mejor camino posible, algo que puede tener repercusiones graves como retrasos a la hora de entregar la medicina al agente Owner o, directamente, que un agente no pueda alcanzar un punto en concreto de la casa.
Tras una investigación sobre posibles formas de abordar este problema, la solución escogida por nuestro equipo de trabajo ha sido el uso del algoritmo A* junto con la distancia Manhattan (que será la más adecuada debido a que los agentes solo pueden moverse en vertical y en horizontal).
Para realizar su implementación, decidimos definir en el proyecto el elemento de Nodo (que será lo que evalúe nuestro algoritmo para encontrar el camino deseado) junto con el A* haciendo uso de clases en Java.
El algoritmo examinará toda la casa, tratando cada casilla como un nodo que contará con una serie de características (si es un obstáculo o no, el coste de desplazarse hasta él...). Una vez realizado esto, se establecerá el nodo del que parte el agente y aquel al que quiere llegar, para posteriormente calcular los costos de desplazamiento (es decir, cuantos movimientos serán necesarios), tanto desde el nodo del que parte el agente al inicio como desde el nodo actual. Este último se conoce como coste heurístico, y es aquí donde se implementa la distancia Manhattan, la cual nos servirá para calcular esa distancia sin tener en cuenta caminos diagonales. De esta forma, se obtiene una ruta lo más corta posible de una forma óptima.

5 Objetivos semana 6

Una vez realizados correctamente los objetivos de la semana 5, ya han sido implementados buena parte de los requisitos a satisfacer, pero aún queda definir el correcto tratamiento de la toma de medicinas y la simulación del comportamiento natural de los agentes.

5.1 Simulación del comportamiento de los agentes

En uno de los apartados de la sección anterior ya hemos definido como se ha realizado la implementación del "pathfinding" y la lógica detrás de esta. Sin embargo, esto no es más que una herramienta necesaria para poder simular con cierto grado de precisión como se comportarían los agentes en la vida real.
La primera solución que se nos ocurrió fue que el agente Robot solo pudiera desplazarse a un par de puntos específicos de la casa (la habitación donde están las medicinas y el salón donde suele estar el agente Owner), mientras que Owner se movería aleatoriamente por toda la casa, pero este enfoque fue descartado debido a la reducida utilidad que tendría un robot que no pudiera desplazarse por toda la casa libremente y la alta complejidad de introducir movimientos completamente aleatorios.
Pensando en otras posibles soluciones para este punto, llegamos al siguiente razonamiento: en la casa hay una serie de casillas con objetos de interés que son las sillas y camas para que Owner descanse, una nevera de donde obtener alimentos, una entrada principal para recibir pedidos, un lavavajillas y un armario con medicinas. Cuando una persona va a algún sitio, es porque tiene algo que hacer en este sitio, por lo que no tendría sentido

que Owner se moviese al medio del pasillo o a una habitación vacía para no hacer nada. Con esto en mente, llegamos a la conclusión de que, para simular el comportamiento de los agentes, basta con asignarles una serie de zonas a las que moverse e introducir la aleatoriedad a la hora de escoger a cuál de estas desplazarse. De esta forma, el agente Robot se moverá entre la puerta principal, la nevera, el lavavajillas y el armario de los medicamentos, que son las zonas donde puede resultar de utilidad realizando algún tipo de tarea, mientras que el agente Owner se desplazará a cualquiera de los elementos de interés anteriormente mencionados. Para implementar esta solución, será neceario añadirle un plan a los agentes que obtenga un número aleatorio y, en base a dicho número, se escoja una posición a la que ir además de añadirle una percepción a un agente cuando esté al lado de uno de los anteriormente definidos como objetos de interés para facilitar las interacciones con estos.

5.2 Configuración inicial de las pautas

Para que la gestión de los medicamentos por parte de los agentes se realice de forma correcta, el agente Robot deberá saber los medicamentos que debe de tomar su dueño y la pauta de consumo de estos. Nuestro equipo ha optado por pasarle a cada agente uno serie de creencias iniciales para solucionar este problema. Concretamente, Robot recibe las cantidades de cada uno de los medicamentos y Owner las pautas asociadas a dichos medicamentos.

6 Objetivos semana 7

Por último, una vez implementados el resto de objetivos de las formas mencionadas anteriormente, nuestro equipo se propuso implementar las tareas restantes, que consistían principalmente en tratar como ambos agentes gestionan la obtención de las medicinas y su posterior transporte y/o consumo (dependiendo del agente).

6.1 Búsqueda de las medicinas por parte del Owner

Para la realización de esta tarea, el agente Owner cuenta con una serie de planes para que, de forma aleatoria, sea él el que vaya a por la medicina y no el agente Robot.
Cuando sea la hora de tomar una de las medicinas, se generará un valor aleatorio para que, en función de este, vaya uno de los dos agentes a por la medicina. En caso de que este valor dicte que debe de ser Owner el que debe ir, se ejecutarán un par de planes que harán que no siga simulando su comportamiento normal y que harán que el agente vaya al armario donde se encuentran los medicamentos, tome aquel que le corresponda, y le envíe un mensaje a Robot, que servirá para que este compruebe que la medicina ha sido consumida. Por último, Owner volverá a su comportamiento habitual (definido en el apartado anterior 5.1).

6.2 Comprobación del consumo de medicinas

Como hemos explicado al final del apartado anterior, si es el agente Owner el que va a por las medicinas, será necesario que el robot compruebe que, efectivamente, su dueño ha tomado dicho medicamento. Anteriormente mencionábamos como Owner mandaba un mensaje a Robot, y es que será este mensaje el encargado de que se ejecuten una

serie de planes de este agente que harán que Robot de desplace hasta el armario de las medicinas y que revise en su conocimiento si hay diferencia entre los medicamentos dentro del armario y los que él sabía que había anteriormente.
En caso de que ambos datos difieran, significará que Owner se ha tomado su medicina, por lo que Robot deberá actualizar sus creencias decrementando de forma correcta los medicamentos restantes en el armario.
En caso contrario, Robot se da cuenta de que Owner no ha tomado la mediación y muestra un mensaje por pantalla indicándolo.

6.3 Entrega de las medicinas por parte del Robot

En caso de que sea el robot el que vaya a por las medicinas (el proceso de elección del agente que va a por las medicinas ha sido explicado en el apartado 6.1), hemos dividido nuestra implementación en una serie de pasos. Primero, Owner recibirá una creencia por parte de Robot diciéndole que no se mueva de la posición en la que está. Hemos escogido este enfoque ya que, en caso de que la medicina deba tomarse en un lapso de tiempo muy específico, hacer que el Robot tenga que recorrer toda la casa buscando a su dueño podría acarrear problemas. De esta forma, el robot le llevará al dueño la medicina de forma muy rápida, evitando así posibles complicaciones.
Una vez Owner ha recibido dicha creencia y está quieto esperando, Robot va al armario a por la medicina correspondiente y la transporta hasta la posición donde su dueño espera por él. Cuando la medicina es entregada, ambos vuelven a su comportamiento habitual.

7 Conclusión

A lo largo de este proyecto hemos trabajado en una serie de soluciones a diferentes desafíos con el objetivo de implementar los requisitos especificados en los diferentes puntos de esta memoria. A pesar de los ya mencionados fallos, hemos conseguido crear un proyecto principalmente funcional y hemos logrado ofrecer una solución válida a buena parte de los problemas planteados, como puede ser la implementación del movimiento de los agentes de forma que siempre encuentren el camino más óptimo para llegar a su destino, el añadir un elemento al entorno completamente funcional o la simulación del comportamiento real que tendría una persona representado a través del agente Owner.