

Práctica 2122q2

Sergio Alvarez

salvarez@cs.upc.edu

SID2022

<https://kemlg.upc.edu>

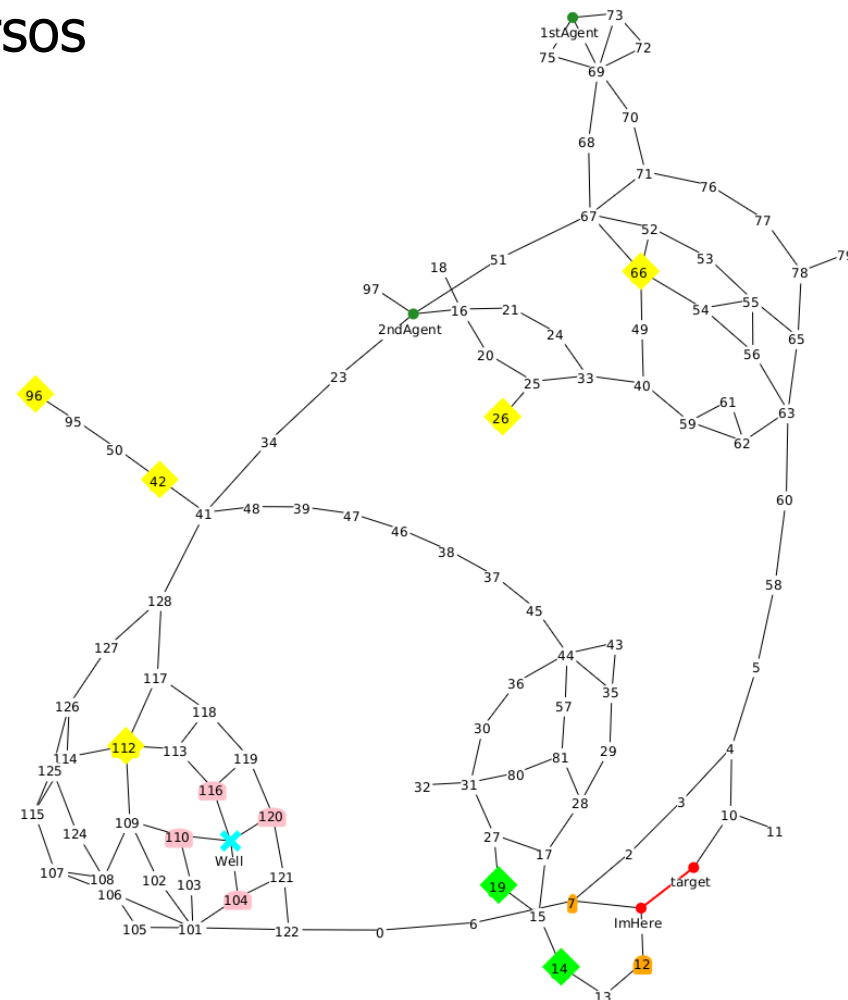


Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA




Resumen

- Entorno basado en la exploración del mapa y la recolección de recursos
 - Parcialmente observable
 - Dinámico
 - Continuo
 - Determinista
 - Accesible
- Diferentes tipos de agente
 - Explorador (*explorer*)
 - Recolector (*collector*)
 - Almacenaje (*tanker*)



Resumen

- Objetivo colectivo (cooperativo):
 - Minimizar recursos remanentes en el mapa
 - Objetivos individuales (competitivos):
 - Explorador: Maximizar conocimiento del mapa y transferencia de conocimiento
 - Recolector: Maximizar recursos recogidos
 - Almacenaje: Maximizar carga de recursos
 - Los nodos sólo aceptan un agente a la vez
 - Tipos de nodo
 - Recursos (oro, diamante)
 - ◆ Nivel de habilidad mínimo para abrir cerradura
 - Obstáculos (pozos)
- 



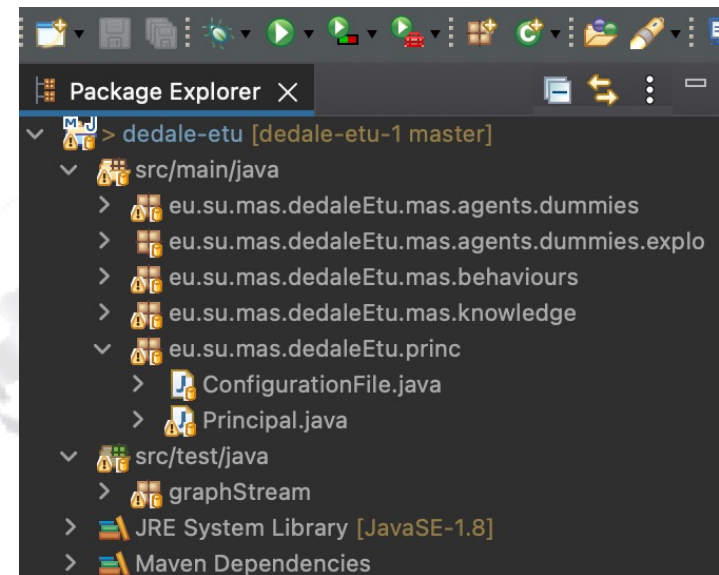


Entorno

- Proyecto Dedale (LIP6 Research Group, Computer Science, Sorbonne University)
 - Usado en el curso de máster de Fundamentos de Sistemas Multi-Agente
 - <https://dedale.gitlab.io/>
 - Usa JADE como plataforma
- Enlaces:
 - Instalación del proyecto:
<https://dedale.gitlab.io/page/tutorial/install/>
 - Configuración de los agentes:
<https://dedale.gitlab.io/page/tutorial/deployagents/>
 - Configuración del entorno:
<https://dedale.gitlab.io/page/tutorial/configureenv/>
 - Javadoc: <https://dedale.gitlab.io/API/index.html>

Cómo empezar

- En Eclipse, descargar el proyecto via Git
 - File -> Import -> Git -> Projects from Git -> Clone URI
 - Introducir la URL: <https://gitlab.com/dedale/dedale-etu.git>
 - Aceptar todas las opciones por defecto, excepto (opcionalmente) el directorio local donde se guardará
- Esperar a que descargue dependencias e indexe
- Buscar la clase Principal
- Ejecutar (botón verde de "play")
- Aparecerán varias ventanas:
 - JADE GUI
 - JADE Sniffer
 - Entorno Dedale



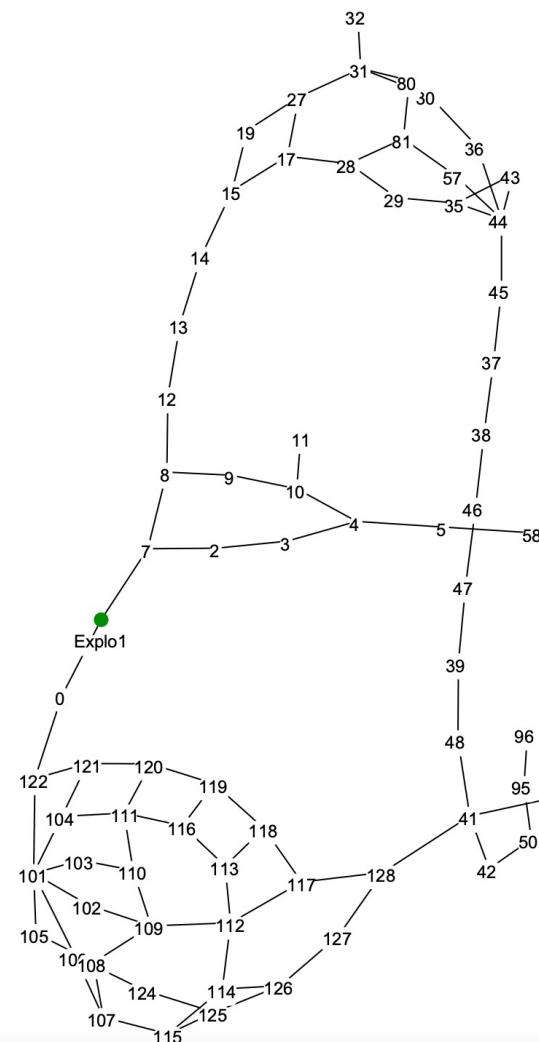
Cómo empezar

- Por defecto la plataforma se inicia con un agente de la clase `DummyMovingAgent`, que se mueve de manera aleatoria
- Si pulsamos ENTER en la consola:
 - Hace el siguiente movimiento aleatorio
 - Muestra las percepciones disponibles desde su nodo actual

```

Explo1 -- myCurrentPosition is: 6
Explo1 -- list of observables: [<6, []>, <0, []>, <7, []>]
Press enter in the console to allow the agent Explo1 to
execute its next move

```



Funciones disponibles para el agente Dedale

- `String getCurrentPosition()`
 - Devuelve la posición actual del agente
- `List<Couple<String,List<Couple<Observation,Integer>>>> observe()`
 - Devuelve el conjunto de percepciones disponibles
- `boolean moveTo(String myDestination)`
 - Si el nodo `myDestination` es accesible desde la posición actual, mueve el agente a este nodo
- `void sendMessage(ACLMessage msg)`
 - Envía un mensaje, si el destinatario está en rango
 - **IMPORTANTE: Este es el único método válido para la comunicación entre agentes situados en el entorno**



Funciones disponibles para el agente Dedale

- `String getMyTreasureType()`
 - Devuelve el tipo de recurso que el agente puede recoger
- `Set <Couple<Observation,Integer> getMyExpertise()`
 - Devuelve los niveles de habilidad del agente
- `int getBackPackFreeSpace()`
 - Devuelve el espacio libre del agente para recoger recursos
- `boolean openLock(Observation o)`
 - Abrir la cerradura del recurso del nodo actual, siempre y cuando el agente tenga suficiente nivel de habilidad
- `int pick()`
 - Recoger la máxima cantidad de recurso posible del nodo actual (mínimo entre capacidad del agente y recursos remanentes), siempre y cuando el agente pueda recoger este tipo de recurso
- `boolean EmptyMyBackPack(String agentSiloName)`
 - Descargar los recursos transportados por el agente en el agente de almacenamiento correspondiente al nombre dado, siempre y cuando esté en un nodo vecino



Configuración de los agentes del entorno

- Fichero resources/agentExplo
 - Primera línea: nombre del fichero con la configuración del mapa
 - Resto de líneas: un agente por cada línea, con parámetros separados por ":"
 - ◆ Tipo del agente (agentExplo/agentCollect/agentTanker)
 - ◆ Nombre del agente
 - ◆ Rango de comunicación (número de nodos de "radio")
 - ◆ Posición inicial (número del nodo o free para aleatorio)
 - ◆ Capacidad inicial y máxima de transporte de oro
 - ◆ Capacidad inicial y máxima de transporte de diamante
 - ◆ Radio de detección (distancia en nodos a la que se puede percibir a este agente por parte de los demás: *stench*)
 - ◆ Nivel de habilidad para abrir cerraduras
 - ◆ Fuerza del agente (irrelevante)

Configuración de los agentes del entorno

- Fichero Principal.java
 - Hay que mapear los nombres de agente del fichero agentExplo con clases de agente (extends AbstractDedaleAgent)
 - Tenéis varios ejemplos comentados en esta clase

```

//1) Get the container where the agent will appear (the default should be always fine)
c = containerList.get(ConfigurationFile.LOCAL_CONTAINER2_NAME);

//2) Give the name of your agent, MUST be the same as the one given in the entities file.
agentName="Explo1";

//3) If you want to give specific parameters to your agent, add them here
Object [] entityParametersExplo1={"My parameters"};

//4) Give the class name of your agent to let the system instantiate it
ag=createNewDedaleAgent(c, agentName,DummyMovingAgent.class.getName(),
                        entityParametersExplo1);
agentList.add(ag);
  
```



Entrega de la práctica

- El enunciado completo (con rúbrica) se colgará en el Racó antes del 2 de abril y la entrega será el 3 de junio
- Cada grupo deberá escoger 2 tipos de agente e implementarlos en una misma clase de agente
 - Los grupos de 4 deben implementar los 3 tipos
- Dos agentes a entregar:
 - 1 agente puramente reactivo situado en el entorno, herencia de AbstractDedaleAgent y capaz de comunicarse con los demás agentes situados
 - 1 agente BDI que no puede comunicarse con ningún agente excepto para recibir percepciones de y enviar acciones al agente situado, y controla:
 - ◆ Los objetivos del agente
 - ◆ Las creencias del agente, opcionalmente mediante ontologías
 - ◆ Los métodos de coordinación con otros agentes



Knowledge Engineering and Machine Learning Group UNIVERSITAT POLITÈCNICA DE CATALUNYA

Sergio Alvarez
salvarez@cs.upc.edu

2022

