Universitat Rovira i Virgili

Introduction to MultiAgent Systems

Activity 1

Final implementation of the practical exercise

Authors: Aleix Toda Mas, Alvaro Romero Diaz, Jordi Riu, Jorge Alexander, Josep Famadas

Supervisor: Dr. Jordi Pujol

Date: 1st January 2018

Table of Content

1. Introduction

A multi-agent system is a system in which different agents live and cooperate between them. These systems are used to solve complex problems or hardly solvable by one unique agent. All these agents can be interpreted as smart entities that live in a common environment.

In this particular case, we deal with a distributed system where the combined behavior of these agents produces a result considered as intelligent. To be more precise, the problem to solve (addressed in the following point) is the metal finding of a grid and the manufacturing of it.

2. Problem statement

This work consists on the implementation of a multi-agent system that efficiently simulates a metal collection center. This task is accomplished by two types of agent:

- Prospectors: this type of agent is responsible for going through the map cells and discovering metals. They must report the type of metal and quantity.
- Diggers: these agents are in charge of collecting the metals. They dig the different metal units that exist in a cell and store them in their own backpack. Where appropriate, they should go to manufacturing sites to exchange metals for "points".

For the coordination of all the agents of the system there are a series of agents that must supervise all the operation of the game:

- System agent: agent in charge of game control and simulation. Its most important task is to continually update the game status.
- Coordinator agent: the link between the system agent and all other agents. It centralizes the orders to be executed in each turn.
- Prospector coordinator agent: agent responsible for coordinating prospectors. It is the means of communication between prospectors and other agents.
- Digger coordinator agent: agent responsible for coordinating diggers. It is the means of communication between diggers and other agents.

Therefore, taking into account all these agents we can set our multi-agent structure as the next network:

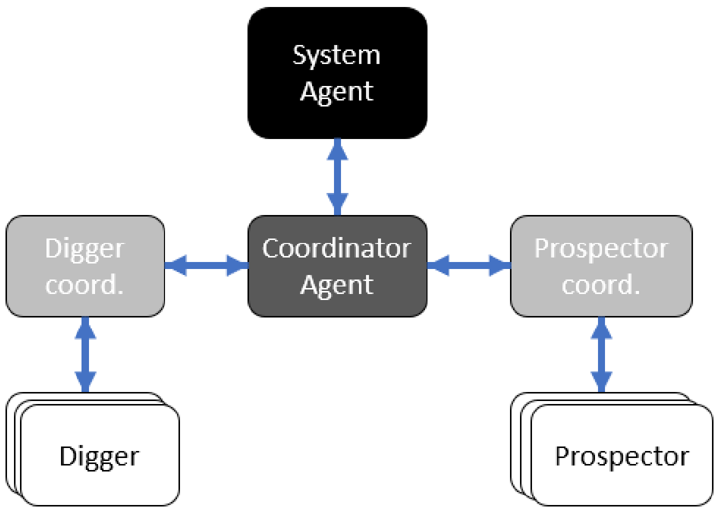

Figure 1: Multi-Agent System architecture

The exchange of metals must be carried out at specific points called manufacturing centers. Depending on the center, the amount of points awarded to each metal varies and, therefore, to maximize the statistics there must be a reasoning that makes every digger go to the right manufacturing center.

To conclude, say that the metal must be randomly generated in the grid to obtain a correct simulation of the requested multi-agent system.

3. Action Flow

The shift starts when the system agent sends the current state's game settings to all the other agents. This flow of information is represented in the next graph:

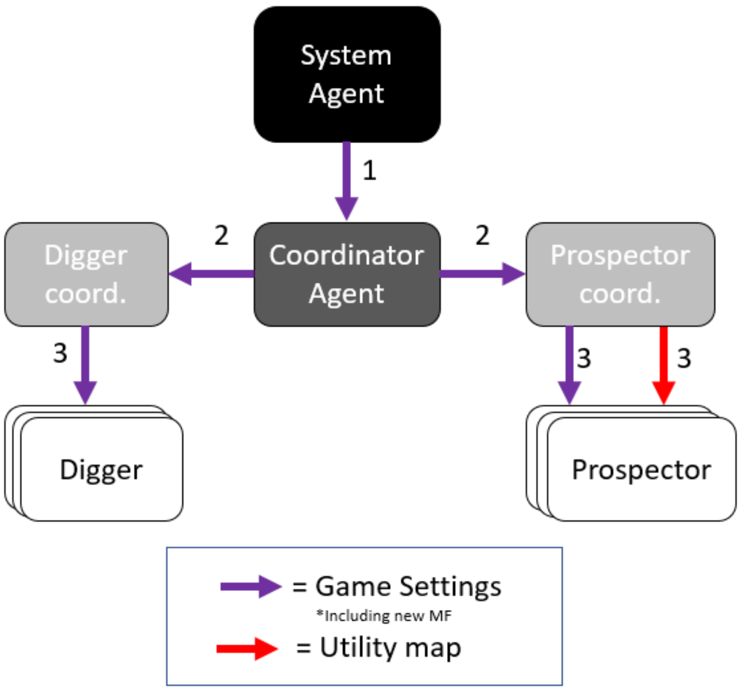

Figure 2: New turn game settings spread

As we can see, the information leaves the system agent and advances through the coordinators agents to all nodes, thus covering all diggers and all prospectors (represented as leaf nodes). Also, the prospector coordinator has the map of utilities implemented and in each turn, he sent it updated to each prospector.

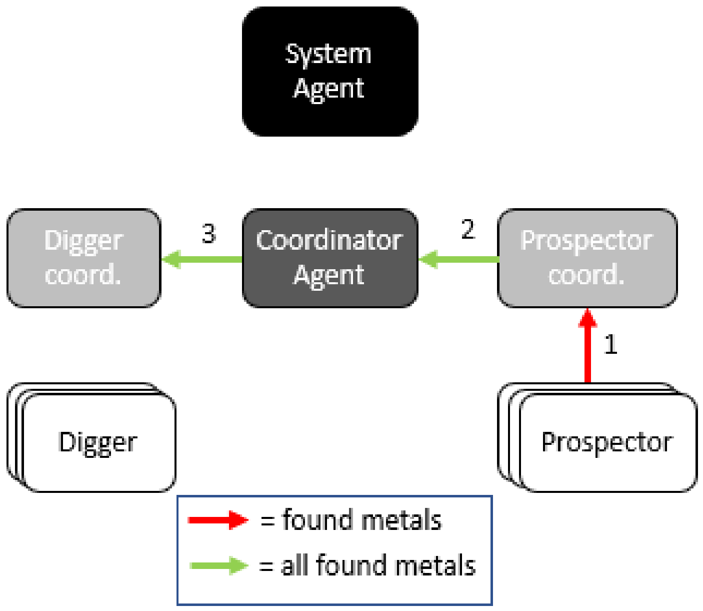

Figure 3: Found metal fields to assign

When all the agents have the map, it's the prospectors' turn. They should discover the metals that exist in its vision range and send them to the prospector coordinator. He collects all the metals and brings them together in a single list to transmit this information through the coordinator to the digger coordinator.

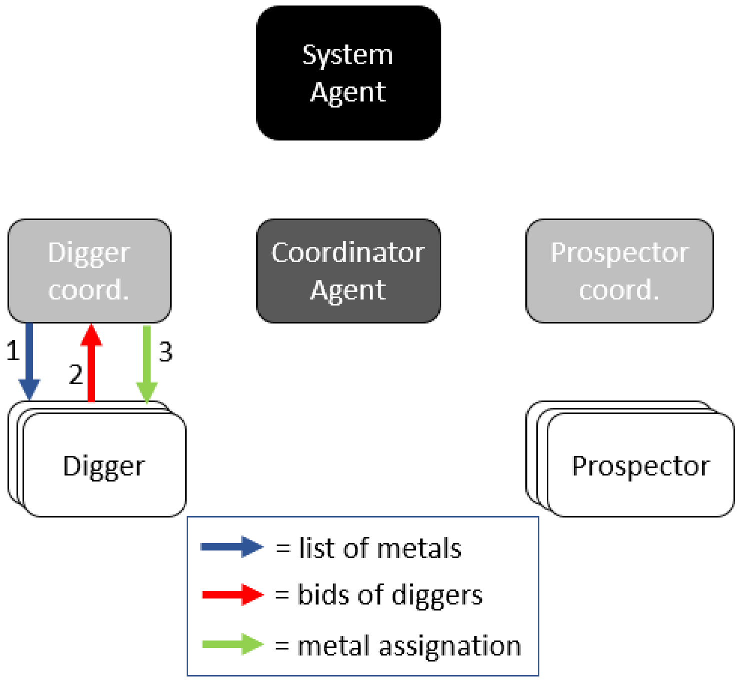

Figure 4: Metal field assignation (Selectivity Protocol)

When the digger coordinator has the list of discovered metals, He transmits it to the diggers. Internally, each one of them calculates a bid price for each metal and send it back to the digger coordinator. Once the digger coordinator has received bids from all diggers, he decides the allocation of the metals and communicates it to the diggers.

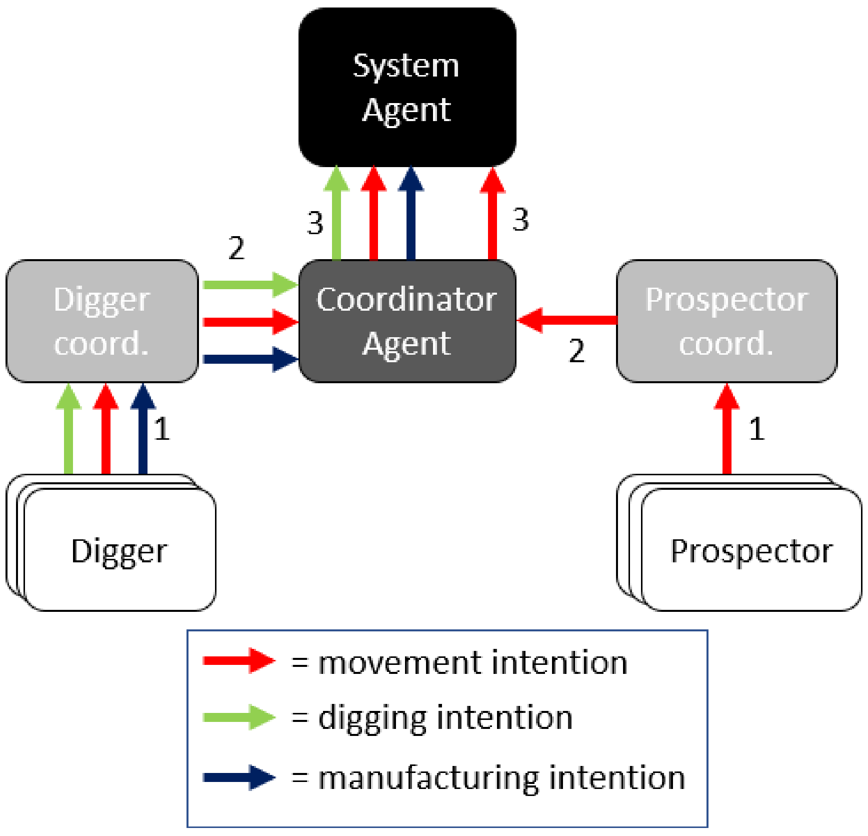

Figure 5: New turn desired actions

Once all these steps have been completed, we can proceed to the requests that are made to the system agent, such as motion requests, digging requests and manufacturing requests (these last two are only made by the diggers).

When all these steps are completed, you can start the next turn of the game by returning to the first point explained in this section.

4. Agents and Behaviors

4.1. System Agent

- **Amount:** 1
- **Mobility:** Static
- **Communication:** Coordinator Agent
- **Function:** This agent acts as the game master. It is responsible for creating the game settings, graphical interface, and the rest of the agents.

  Also, it is who will perform all modifications in the game, who will add new metal fields and who applies the movements of digger and prospector agents.

  Once all game, GUI and agents are properly initialized, it sets the **RequestResponseBehaviour**.

  It has 2 main functions:

  - **checkTurnChanges**: This function could be clearly explained by its sequential steps:
    1. Set new metal fields detected to visible: set all metal fields detected by prospector agents to visible in the map.
    2. Set up diggers working: using the list of digging requests, set the digger cell with digger agent working (block movement through it) and remove one metal unit from the digging metal field.
    3. Free cells where digger agents have finished working
    4. Movements checking: all requested movements are checked and if possible, also updated in the map.
    5. Update manufacturing centers (rewards): all requested metal units to manufacture are changed for the corresponding reward.
    6. Substitute the old map with the new checked map (next turn map)
  - **decrementStep**: This function decrement one step (subtract one step to the current step number). If there are no remaining turns, game is ended by killing the system agent.

    Game map and statistics panel are updated.

  Furthermore, System Agent performs all statistical calculations that will be periodically updated in the statistical sheet in the GUI.

- **Behaviors:**
    - **RequestResponseBehaviour:** It triggers when the Coordinator Agent sends a “GET_MAP” request to the System. Its only function is to send to the requester the first turn GameSettings, which will start the game.
    - **ChooseActionSA:** It triggers when it receives a “CHOOSE_ACTION” message (from the Coordinator Agent) which will always contain a CompleteMessage object.

      CompleteMessage object contains:

        1. Movement requests
        2. Digging requests
        3. Manufacturing requests
        4. MetalFieldList (New turn discovered metals)

      The behavior calls **checkTurnChanges** SystemAgent method and once checked map (next turn map) is recieved, it also calls the **decrementStep** SystemAgent method.

      Finally, it sends back to the Coordinator Agent the new turn GameSettings with the updated map.

4.2. Coordinator Agent

- **Amount:** 1
- **Mobility:** Static
- **Communication:** System Agent, Digger Coordinator Agent, Prospector Coordinator Agent.
- **Function:** This agent is the general coordinator of the Multi Agent System. It acts as link of the system agent and the system, meaning that every single communication that any agent wants to send to the system has to go through it. It is also responsible for communicating the diggers side with the prospectors’ side.
- **Behaviors:**
    - **RequesterBehaviour:** It triggers when the Coordinator Agent is created and sends a “GET_MAP” request to the System Agent. When it receives the initial

GameSettings it resends them to the Digger Coordinator and the Prospector Coordinator.

- **ActionHandlingCA:** It triggers when the Coordinator Agent receives a message with language set to "DIG_ACTION". It will act depending on the content of the message:
  - Metal Field List (from Prospector Coordinator): It merges the current found metals with the new ones received and send the resulting metal field list to the Digger Coordinator.
  - Digging Message List (from Digger Coordinator): It adds the petition to a Complete Message with all the petitions, if the 4 petitions have been added, sends the Complete Message to the System Agent.
  - Moving Message List (from Digger Coordinator): It adds the petition to a Complete Message with all the petitions, if the 4 petitions have been added, sends the Complete Message to the System Agent.
  - Digging Message List (from Prospector Coordinator): It adds the petition to a Complete Message with all the petitions, if the 4 petitions have been added, sends the Complete Message to the System Agent.
  - Manufacturing Message List (from Digger Coordinator): It adds the petition to a Complete Message with all the petitions, if the 4 petitions have been added, sends the Complete Message to the System Agent.

- **MapHandlingCA:** It triggers when the Coordinator Agent receives a "NEW_MAP" message from the System Agent. It updates the complete metal field list with the turn changes and then sends the new turn game settings to the Digger Coordinator and Prospector Coordinator.

4.3. Digger Coordinator Agent

- **Amount:** 1
- **Mobility:** Static
- **Communication:** Coordinator Agent, Digger Agents (all of them).
- **Function:** This agent is responsible for the coordination of the digging side. It receives from the Coordinator Agent the new turn game and the metal fields found in the new turn by the prospectors and assigns them to the diggers through the protocol seen in Figure 4.

- **Behaviors:**
    - **MapHandlingDCA:** It triggers when the Digger Coordinator Agent receives a "GET_MAP" message from the Coordinator Agent and acts depending on the content of the message:
        - (new turn) Game Settings: It simply updates its own current Game Settings and send them to all the diggers.
        - MetalFieldList: Sends the MetalFieldList to the diggers so they can "bid" (the "bids" will be handled in SelectivityVotingDCA).
    - **SelectivityVotingDCA:** It triggers when the Digger Coordinator receives a "SELECTIVITY" message from any of the Diggers containing their "bids" for the current MetalFieldList. If it has received all the "bids", makes the assignation and sends a message to the Diggers with it.
    - **ChooseActionDCA:** It triggers when the Digger Coordinator receives a "CHOOSE_ACTION" message from any of the Diggers containing their petition to move, dig or manufacture. Once it has received the petition from all the Diggers, put them together and send them to the Coordinator Agent.

4.4. Prospector Coordinator Agent

- **Amount:** 1
- **Mobility:** Static
- **Communication:** Coordinator Agent, Prospector Agents (all of them).
- **Function:** This agent is responsible of the coordination of the prospecting side. It receives the metal fields found in the new turn by the prospector agents and sends them to the digging side through the Coordinator Agent. It is also responsible of the organization of the prospectors' movement.
- **Behaviors:**
    - **MapHandlingPC:** It triggers when the Prospector Coordinator receives an "INFORM" message and acts depending on its content:
        - (new turn) Game Settings (from Coordinator): It updates its current game settings, computes the new utility map for the Prospectors and send it to them.

          Utility calculation is the most important action that prospector coordinator has to perform as it will have the greatest impact on all prospectors' movements, with the corresponding impact on finding new metal. This function is explained in detail in the previous activity. **applyUtility** function perform the following steps:

1. FieldCells utility update and reset Path Cells
2. Reset fieldCells inside visual field utility
3. PathCells simple utility set: as deeply explained in the previous Activity, it sets the utility of all path cells based on the surrounding field cells utility value.
4. Penalize cells with prospector agents
5. PathCells propagation utility update: a window of a cell and all its surrounding cells (square of 9 cells in total) with not uniform weights is passed over all path cells, making an unequally (based on weights in the window) utility average of the cells inside the window.

   This action is performed 4 times, 1 time for each direction where one direction is from the 0, 0 coordinates to xMax, yMax and the other 3 directions are the remaining coordinates combinations.

   The utility resultant map is the average of the 4 maps obtaining with the 4 coordinates combination sweeps.
6. Set cells with digger agents working with negative utility

- MetalFieldList (from any Prospector): It adds the new-found metals to its list, checks if all Prospectors have sent their new metal field list, and, if so, sends it to the Coordinator Agent.
- MovingMessage (from any Prospector): It adds the moving petition to a list and, when it has received the petition of every Prospector, it sends the moving message list to the Coordinator Agent.

4.5. Digger Agent

- **Amount:** Depends on the game settings (In the provided case: 8)
- **Mobility:** Can move through the map.
- **Communication:** Digger Coordinator Agent, [Prospector Agent] (Just in case they form a coalition to follow it).
- **Function:** These agents are the ones in charge of extracting metals from the metal fields and carry them to a manufacturing center. They are assigned a metal field to go depending on their "bids" on the Selectivity "Voting" method explained in this agent's behaviors.

- **Behaviors:**
    - **MapHandlingDA:** It triggers then the Digger receives a "GET_MAP" message from the Digger Coordinator containing the new turn Game Settings. It just updates the Digger current position.
    - **SelectivityVotingDA:** It triggers when the Digger receives a "SELECTIVITY" message from the Digger Coordinator containing the current found Metal Field List. It computes the "bid" for every of the Metal Fields and sends the list of "bids" to the Digger Coordinator.
    - **ChooseActionDA:** It triggers when the Digger receives a "CHOOSE_ACTION" message from the Digger Coordinator containing its assignation after the Selectivity process has finished. Acts in function of the assignation and its current state:
        - No metal assigned + Carrying Metal + Touching a valid Manufacturing Center: Sends a manufacturing message to the Digger Coordinator.
        - No metal assigned + Carrying Metal + NOT Touching a valid Manufacturing Center: Computes the best manufacturing center to go and also computes the movement to go there. Sends a movement message to the Digger Coordinator.
        - No metal assigned + NOT Carrying Metal: The digger has to follow its nearest prospector. It starts a ContractNet protocol to find the best prospector.
        - Metal assigned: Computes the distance to the assigned metal. If it is touching the metal, sends a Digging message to the Digger Coordinator.

Else, computes the best movement towards the metal field and sends a Movement message to the Digger Coordinator.

- **ContractNetDA:** It is a ContractNetInitiator behavior which triggers when the Digger has not been assigned any metal field and is not carrying metal. It sends requests to all the prospectors which will respond the request with their position and the Digger will select the "winning" prospector and tell it to him.

4.6. Prospector Agent

- **Amount:** Depends on the game settings (In the provided case: 4)
- **Mobility:** Can move through the map. Uses the utility map to decide where to move. An additional parameter has been added to improve the movement efficiency. This new parameter is the momentum, and as its name points out, momentum increment its value with each turn that the agent holds the same direction. This parameter has shown to improve prospectors path, increasing the probabilities of going to new map areas and avoiding previously visited paths. When the agent finds a field cell in the momentum direction, momentum has no effect, so the prospector will decide what will be its next movement based only on utility map.
- **Communication:** Prospector Coordinator Agent, [Digger Agent] (Just in case they form a coalition to be followed by some of them).
- **Function:** These agents are the ones in charge of finding metals. They move through the map and try to find them. They send the metal found each turn to the Prospector Coordinator.

- **Behaviors:**
  - **MapHandling:** It triggers when the Prospector receives an "INFORM" message from the Prospector Coordinator containing the new utility map. It updates the Prospector position, search if there is any metal field, and builds a MetalFieldList with the metal fields found. Then, it sends the list to the Prospector Coordinator.
  - **ContractNetPA:** It triggers when receives a "FIPA_CONTRACT_NET" "CPF" message from a Digger. It sends back to this digger its current position, and waits for him to tell if he will follow the Prospector.

5. Work Changes

5.1. Digger Coordinator Gold / Digger Coordinator Silver

Since the beginning of the word we talked about adding two sub-Digger Coordinators, one for gold and another for silver. The function of them would be just to take some of the actions that the Digger Coordinator would perform in order to reduce its computational load.

However, during the implementation of the project, we have realized that the full computational load the Multi Agent System is not enough to be worth to add these two agents and the implementation complexity it would carry.

That is why we have finally decided not to implement them.

5.2. Prospector agent

In this case, as previously explained, we have added the momentum variable. This new variable has shown to be very simply to apply and providing good performance improving.

5.3. Utility Computing

Although it was explained in the last delivery, we have not taken into account the distance between prospector agents when computing the utility due to the fact that we already achieve good results and this implementation would just increase the code complexity.

5.4. Digging

In the previous delivery we designed a coalition formation method for digger agents, so that they would cooperate if assigned to the same metal field. However, since the metal fields are generated much faster than they are dug, it is very unlikely for that situation to happen.

5.5. Parameters Optimization

We also thought about including some learning abilities for the agents to optimize the parameters used in the weighted sum of the “selectivity bidding” and the manufacturing center selection. Unfortunately, we did not have enough time to implement that.

5.6. Manufacturing

In the implementation of the project, the digger agents always prioritize mining over manufacturing, i.e. if they have a metal field assigned and have free slots they will not even consider selecting a manufacturing center.

6. Performance Measurements

One of the most important things when designing our system is that, despite the fact that it is heterogenous, all agents will be cooperative, with the aim to achieve the best value of the following evaluation parameters given a limited number of turns:

- **Benefits:** amount of points received.
- **Manufactured metal:** current amount of each metal already manufactured.
- **Average benefit for unit of metal:** amount of points received per metal unit, regardless of the metal's type.
- **Average time for discovering metal:** the amount of time spent from metal appearance until a prospector discovers it.
- **Average time for digging metal:** the amount of time spent from metal discovery until the first digger gets to that point.
- **Ratio of discovered metal:** the ratio of the metal that is already discovered by prospectors from the total.
- **Ratio of collected metal:** the ratio of the metal that is already collected by diggers, including that in the digger and that already disposed in manufacturing centers.

7. Results

In order to check the performance of our Multi-Agent System we were provided a default settings files which initial configuration map can be seen in *Figure 6*.

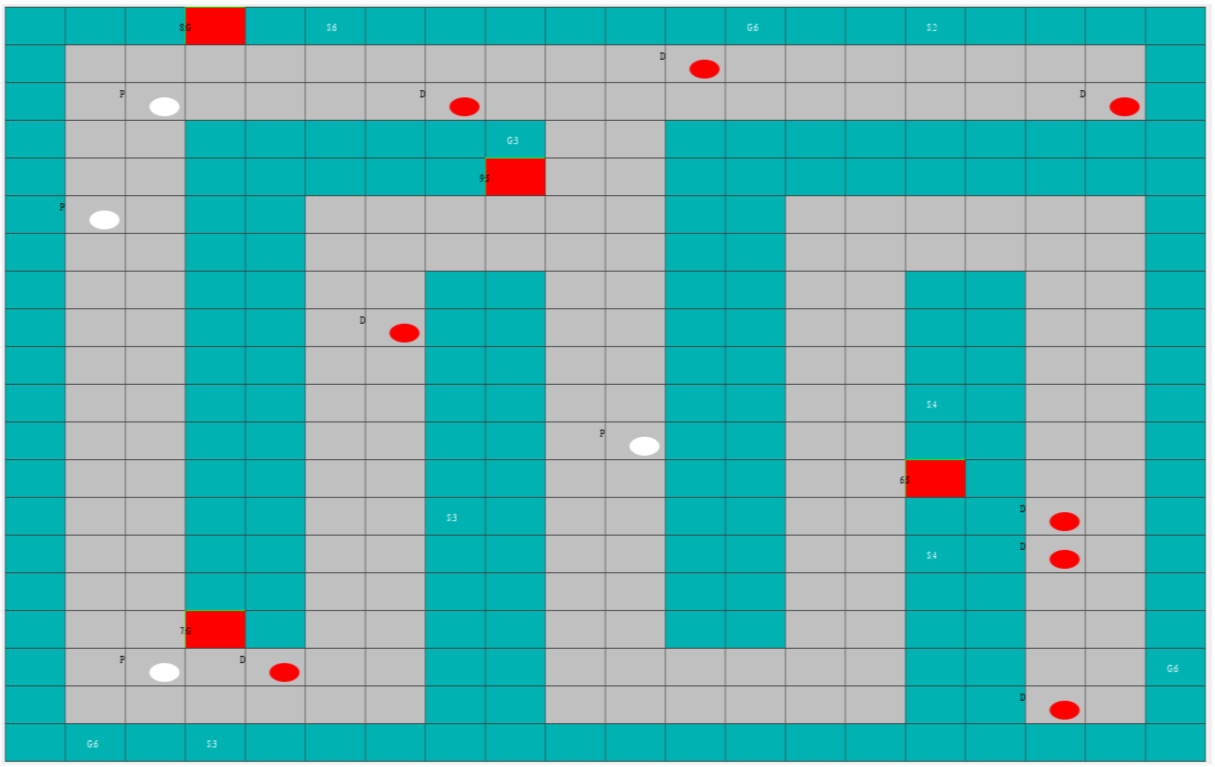

Figure 6: Initial configuration map

The results obtained after 600 turns are the following:

	Execution 1	*Execution 2*	*Execution 3*	*Execution 4*
Benefits	2296	2369	1500	1588
Manufactured Gold	140	156	74	123
Manufactured Silver	138	130	108	74
Average benefit for unit of metal	8.26	8.28	8.24	8.06
Average time for discovering a metal	53.16	67.15	34.18	28.56
Average time for digging a metal	74.86	91.42	88.75	79.57
Ratio of discovered metal	0.94	1.00	0.96	0.95
Ratio of collected metal	0.31	0.31	0.26	0.26

Taking into account that the metal generation is not deterministic, the second and third evaluation metrics are not relevant.

Also, seeing that the average value of the 4 manufacturing centers is 7.5 points per metal unit, and the Average benefit for unit of metal in our implementations is always higher than 8, we can conclude that the diggers are correctly prioritizing the higher reward centers.

If we take a look at the Ratio of discovered metals, we see that the Prospectors are able to reach every single cell of the map because the metric is close to 1 or even 1.

On the other hand, the Ratio of collected metal is so low because the diggers are not able to dig and manufacture metals faster than the new ones are created by the System Agent.

Another relevant phenomenon to point out of the execution is that, as seen in Figures 7 and 8, when the 600 turns have expired, the Field Cells surrounding a Manufacturing Center are of its opposite metal type. This shows that the diggers present an emergent behavior which consists on prioritizing the metal fields near the manufacturing centers in which they will manufacture the dug metal.

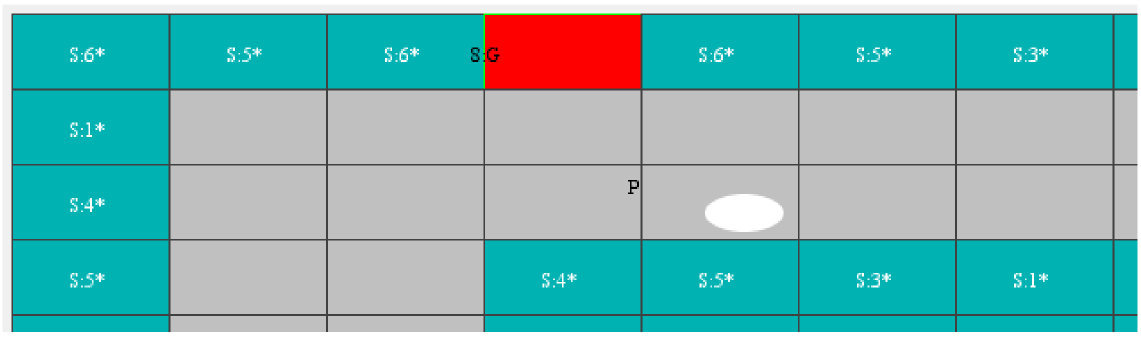

Figure 7: Final turn cells around a gold manufacturing center

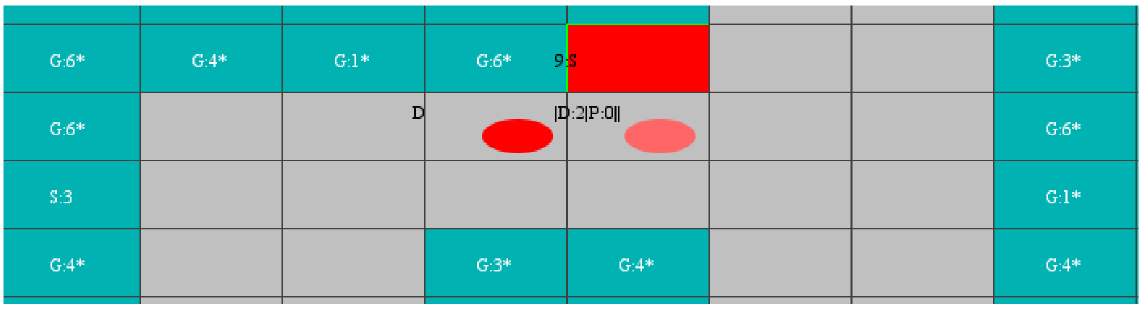

Figure 8: Final turn cells around a silver manufacturing center

8. Conclusions and Future Work

This section covers the main conclusions that we reached thanks to developing the multi-agent system and the future work.

- Working with a system in which there are different agents with different objectives we have understood how to work with it.
- We learnt how to connect different agents thanks to behaviors provided by JADE [4].
- We implemented as much as possible our design of the system, understanding in each moment when we should change it and why.
- We improve our ability with object-oriented programming. In addition, some members of the group did not how to work with JAVA and they took advantage of it learning this programming language.
- Thanks to the use of JADE we have correctly understood this software platform.

As future works to improve this work, can be added the following:

- Maybe the diggers' movement could be more optimal, so it could be improved adding changes.
- We would like to have our prospectors more separated between them in order to cover all the map easily. We deal with it and we almost reach it, having these agents only problems when they are in the same cell.
- We could also add more learning abilities to have an optimal selectivity method or to decide to which manufacturing center should a digger go and when.

9. Bibliography

[1] M. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley, 2009.

[2] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley, 2007.

[3] (Fundation for Intelligent Physical Agents) FIPA, "FIPA Contract Net Interaction Protocol Specification," *Architecture*, no. SC00029H, p. 9, 2002.

[4] "JADE DOC. API." [Online]. Available: http://jade.tilab.com/doc/api/.

[5] "Stack overflow" [Online]. Available: https://stackoverflow.com/.

10. ePortfolio

TEAM JAVADIATORS - Alvaro, Aleix, Josep, Jordi, Jorge					
	Date	**Time**	**Place**	**Participants**	**Notes**
First meeting	01/01/2018	19:00 -20:00	Skype meeting	Whole team	• Work analisis. Decided to go to second call. Action points • Whole team: Enjoy the exams, see you in 22th
Second meeting	22/01/2018	18:00 - 21:00	Skype meeting	Whole team	• First analisys of the full code. • Decided to let 2 days for everyone to read it and understand the behavior and how do they work Action points • Everyone should analyse the code.
Third meeting	25/01/2018	19:00 - 21:00	Skype meeting	Whole team	• Divide work - Jordi & Josep: Digger and DiggerCoordinator. - Alvaro & Jorge: Prospector and Prospector Coordinator. - Aleix: System agent. Action points • Start working on each group part.
Fourth meeting	26/01/2018	18:00 - 21:00	Skype meeting	Whole team	• Merge the work from each group and work into the coordinator agent to do it. Action points • Next two days work together and try to have the code finished by Sunday night.

Fifth meeting	28/01/2018	23:00 - 00:00	Skype meeting	Whole team	• Analyze bugs and errors of the code and start fixing them. Action points • The found errors have to be fixed.
Sixth meeting	30/01/2018	23:00 - 00:00	Skype meeting	Whole team	• Analize possible improvements and set the report template. Action points • Make the improvements, write the report, and make the PPT.
Seventh meeting	01/02/2018	00:00 - 01:00	Skype meeting	Whole team	• Delivery.