Evidence 1. Integrative Activity

Luis Ignacio Gómez López - A01644423

Edgar Osvaldo Navarro García - A01644488

José Eduardo Nájera De Anda - A01644400

Juan Manuel Villalobos Nuño - A01639135

Diego Iván Rodríguez Núñez - A01644772

August 31st, 2025

TC2008B Modeling of Multi-Agent Systems with Computer Graphics

Jesús Israel Hernández Hernández

Oscar Guadalupe Hernández Calderon

Iván Axel Dounce Nava

Description of the problem

In your jewelry store, the employees usually leave a disaster in the display case, leaving every piece of jewelry completely disorganized and in complete disarray. To solve this constant problem without having to open the case and organize the jewels, you decided to buy some organizer robots that could automatize this task. However, the robots were too expensive and you did not have any money left to buy the state-of-the-art multi-agent management software, so you decided to do it yourself.

You now have to design the individual management software, which will be responsible for controlling each individual robot and the multi-agent management software, which will be responsible for coordinating the robots and making them collaborate to reach the final objective of accommodating the jewels by color.

Description of the solution

The implemented solution consists in dividing the area of the jewelry into a grid of positions using the dimensions of the jewelry and the size of the agent. The agent is spawn randomly across the grid, each of them facing a different direction. The agent moves in the grid while searching for a gem, moving towards the closest possible unvisited grid position, giving priority to the direction in which they are facing. Once they decide which position to move, this is marked so that other agents do not move to this position. In case the sensors (collider for sensing the nearest positions) detect a gem, the position of the gem is shared with the other agents so that an agent of the same color of the gem takes it to its destination. Then this position is marked as a blocked position for the agents which are not of the same color as the gem.

When there is a gem of a certain color's position on the shared knowledge, an agent of the same color can reserve it, this makes it as if there is more than one agent of the same color, only one of them gets to the gem. Once the gem gets reached by the agent, the latter moves it into the designated position through the most optimal path. This releases the blocked position so that the agents can move more dynamically through the display case.

Once the agents, one or more, of a certain color have found all their gems, they continue to move throughout the grid, helping the rest of the agents. In case of an error as if the grid has already been fully traversed, but there are still missing gems, the agents activate a random algorithm so they move once again through the entirety of the grid until the missing gems are found.

To avoid collision, through the grid it maintains record of the current position of the agents and the next position of each of our agents marking this position on the grid. Then each agent tries to avoid going to the same grid positions as the rest of the agents; for but when the agents go to a gem diagonally, it was implemented a specific detection system for which in case it detects another agent in front, this agent stops and rotates avoiding the moving agent. For the gems, once a gem of another color is detected, this position is marked as a visited position making the agents avoid this location.

Github link to the project: https://github.com/BlitzExp/FightersJewelrySeekers

Analyze whether there is a strategy that could reduce the time spent, as well as the number of moves made. What would it be? Describe it.

- There could be a strategy in which instead of counting each movement through the grid, the agent would choose a line to move and detect all the gems in that line while also avoiding them, reducing significantly the number of movement especially for larger displays.

- Another strategy which could be used, would be to always recognize where all the agents are, and instead of giving priority to the positions in which the agent is facing, give priority to positions which are further away from the rest of the agents. This would reduce the time spent towards finding all the possible gems.
- Another implementation which could reduce the number of movement and time would be that, once an agent is taking back a gem to the designated area, it takes into consideration the unvisited positions and makes a path using them. Which would reduce the number of unvisited positions while also delivering the gem.

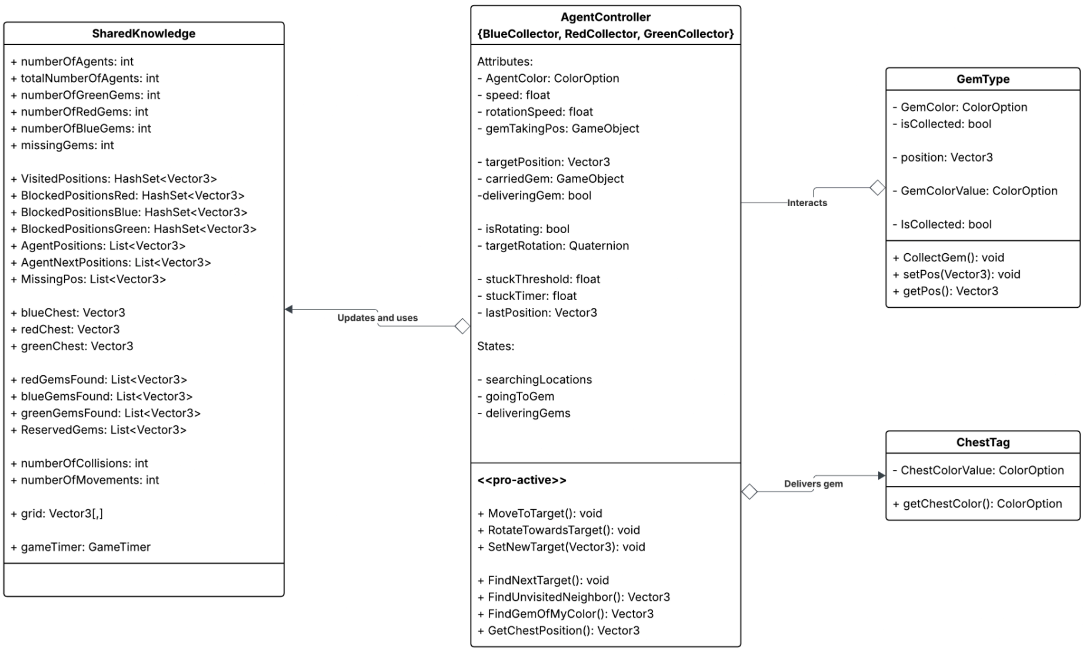
Design description and justification

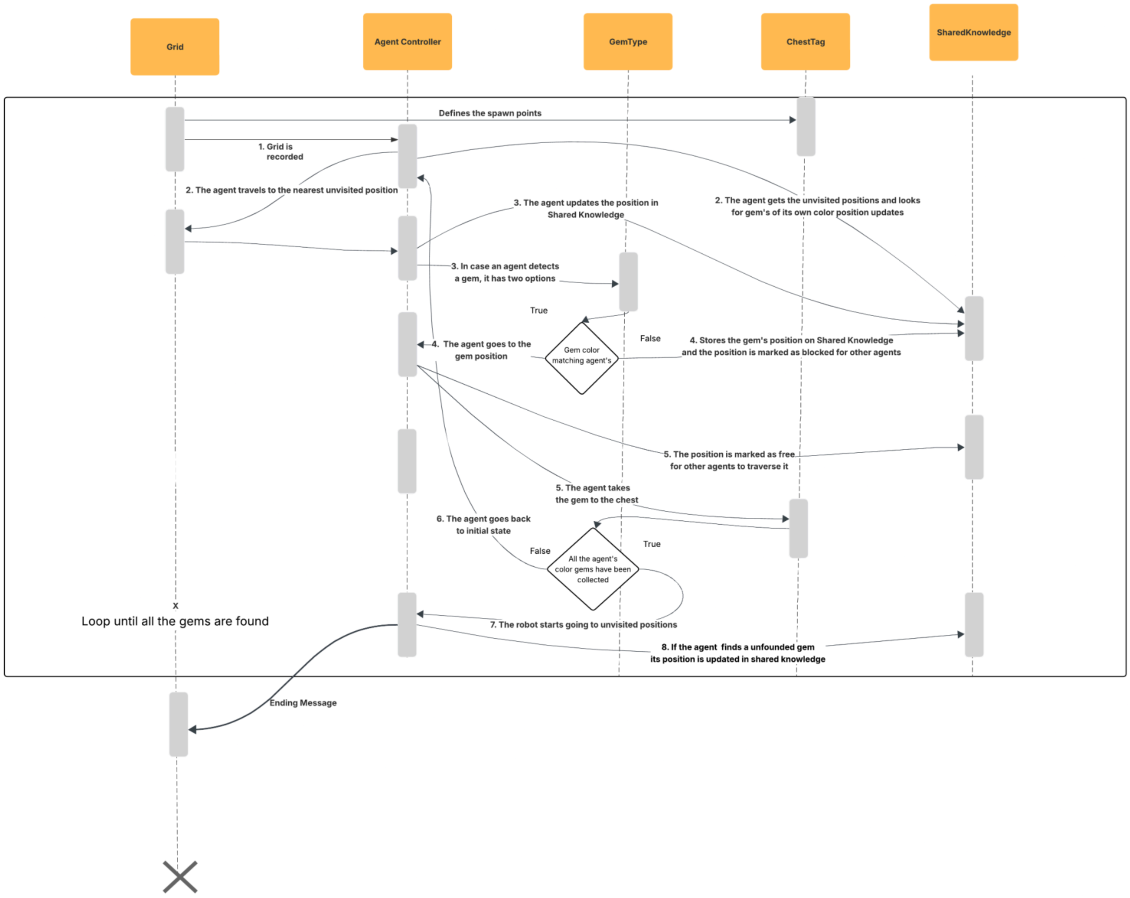
For starters, it was decided to use a grid to divide the "playable" area, as it will make the algorithms for traversing it easier to implement and faster as a whole. Using a grid is the easiest way to keep track of the positions that the agents have already traversed, the ones unvisited, the ones where the recollection points are and the ones where unpicked gems stay at. This also makes it easier to define the protocol for spawn points and robots initial path, as they are spawned facing a certain way, which makes it easier for them all to go to different positions, covering a greater area of the map and avoiding collisions in the early stages.

As for the movement, it was decided that every robot must move to the nearest unvisited position, as this would make it more probable to visit every part of the grid in the fastest possible way. Once a gem of a certain color is found, the agent of the same color would immediately go to that position without following the grid; this way, an obstacle for all the other agents is removed as fast as possible and so, accelerating the grid traversion as it gives more freedom for agents to move. The gem is then taken to the designated recollection point as fast as possible.

To avoid obstacles, it was decided that every robot would mark the next position and the current position in which they are. Doing it this way, each agent knows in every moment which cells are occupied and avoids them. In the case they are about to collide with another agent, a system was implemented in which if they detect the agent in front of them, this agent stops its movement and rotates. Doing it this way, it gives enough time for the other agent to continue its path and avoid the collision.

Class diagrams and agent protocol





Link to diagrams:

https://lucid.app/lucidchart/951dde6e-5ca3-473f-a3cb-201dd8cdf3de/edit?viewport_loc=-9061%2C-790%2C4558%2C2627%2C0_0&invitationId=inv_46a37972-924b-4dd2-804d-63f8b02eaeb5

Results and analysis

It is important to take into consideration that the number of movements is higher than other solutions because the agent moves through a grid, which means each movement is a grid space.

Grid Size

- **3 Agents in a 5 X 5 display case, searching for 3 gems**

Time Taken (s)	Number of Collisions	Number of Movements
2	5	14
1	5	10
1	6	14
1	8	11
1	6	13

Video Demonstration

https://drive.google.com/file/d/1YDg9NKOxiDs01Dfy6nfyw9P21r1JONlW/view?usp=sharing

- **3 Agents in a 10 X 10 display case, searching for 3 gems**

Time Taken (s)	Number of Collisions	Number of Movements
4	5	39
5	9	40
8	14	60
4	7	40
4	12	39

Video Demonstration

https://drive.google.com/file/d/1ZH0r6n8Py93XKRsv5lQ0Ox_457KGzq12/view?usp=sharing

- **3 Agents in a 15 X 15 display case, searching for 3 gems**

Time Taken (s)	Number of Collisions	Number of Movements
8	9	72
11	13	102
13	18	121

11	17	80
11	14	102

Video Demonstration

https://drive.google.com/file/d/10U-vuTdr7_6EB_mdMOdHHycQroq-c_wx/view?usp=sharing

- **3 Agents in a 20 X 20 display case, searching for 3 gems**

Time Taken (s)	Number of Collisions	Number of Movements
15	8	152
26	17	220
25	13	201
31	17	254
19	11	174

Video Demonstration

https://drive.google.com/file/d/1505k6D8QdBUSv-NFY9ffoq2JNa033kb4/view?usp=sharing

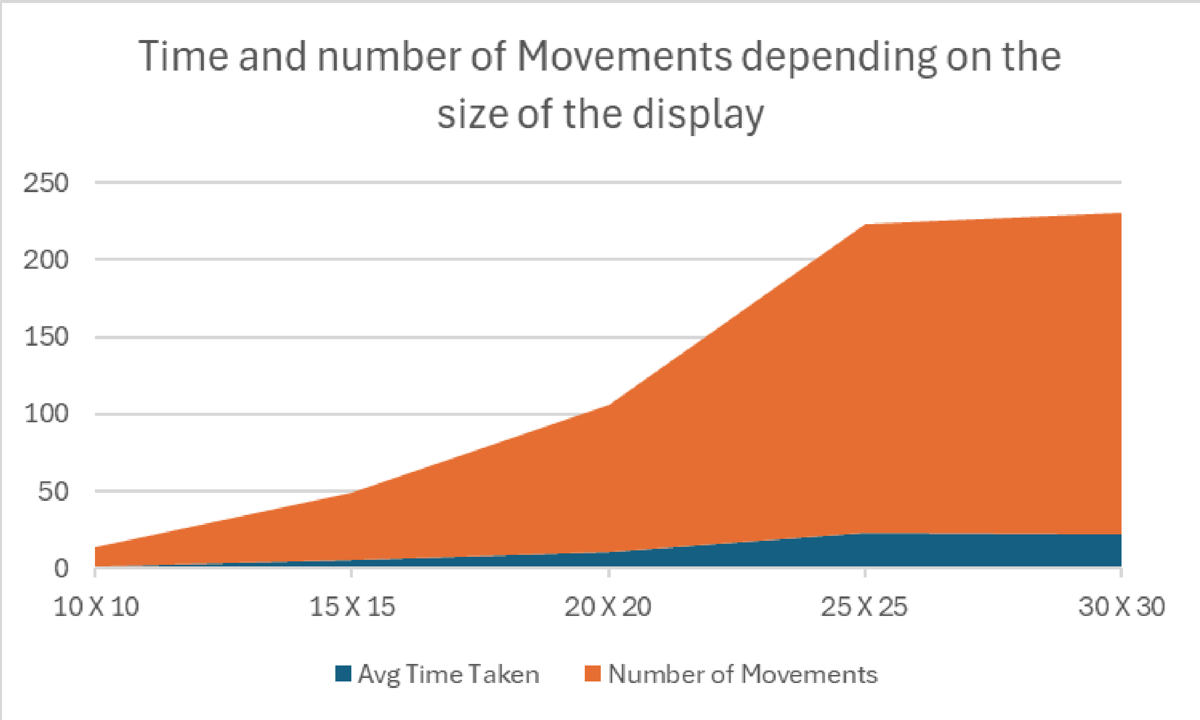
- **3 Agents in a 25 X 25 display case, searching for 3 gems**

Time Taken (s)	Number of Collisions	Number of Movements
16	16	147
19	10	195
10	5	74
36	18	352
27	9	276

Video Demonstration

https://drive.google.com/file/d/1VJ0yasRzM_FIADTB-03z6YnNTzrELvV_/view?usp=sharing

Data Analysis



Observing the data recollected and graph presented before, we can observe that the scale of the display case will have a great impact on the time taken to recollect all the gems and the number of movements. At the start of the graph, going from the display of 10 x 10 to the display of 25 x 25, it can be observed almost a exponential growth of the number of movements. On the other hand, because of our implementation, we can observe that the time taken to complete these simulations is increasing a little but not in a significant amount of time.

Number of gems

It is important to consider that, as it was mentioned previously, while searching, they move by 1 space of the grid; on the other hand, while they are going for a gem or they are returning it to the destination, it only counts this movement as one unit.

- **3 Agents in a 15 X 15 display case, searching for 3 gems**

Time Taken (s)	Number of Collisions	Number of Movements
8	9	72
11	13	102
13	18	121
11	17	80
11	14	102

Video Demonstration

https://drive.google.com/file/d/10U-vuTdr7_6EB_mdMOdHHycQroq-c_wx/view?usp=sharing

- **3 Agents in a 15 X 15 display case, searching for 6 gems**

Time Taken (s)	Number of Collisions	Number of Movements
13	19	88
8	10	35
15	15	129
11	15	58
15	22	108

Video Demonstration

https://drive.google.com/file/d/1Ne4G1JQyDo5k_lcQphVopMzpf45GGgxg/view?usp=sharing

- **3 Agents in a 15 X 15 display case, searching for 9 gems**

Time Taken (s)	Number of Collisions	Number of Movements
14	14	87
13	18	65
14	17	95

14	22	77
20	30	133

Video of Demonstration

https://drive.google.com/file/d/1vFz3fWmHbjeFPYwv38C4AgWIgv-74-wc/view?usp=sharing

- **3 Agents in a 15 X 15 display case, searching for 12 gems**

Time Taken (s)	Number of Collisions	Number of Movements
20	32	112
27	40	122
20	34	108
17	28	84
22	30	110

Video of Demonstration

https://drive.google.com/file/d/1zzsqyshPNDyYS2dZpdVzn4Bu5ADX8nVj/view?usp=sharing

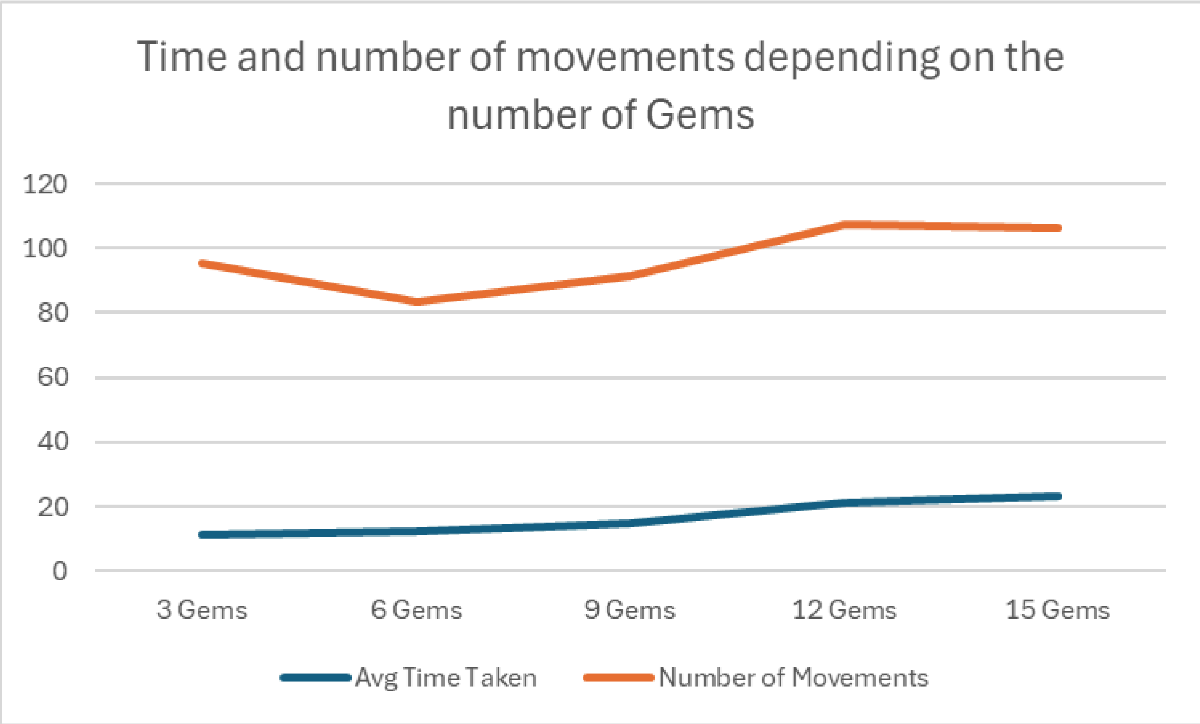
- **3 Agents in a 15 X 15 display case, searching for 15 gems**

Time Taken (s)	Number of Collisions	Number of Movements
19	28	89
22	37	90
18	24	75
33	40	153
24	32	124

Video of Demonstration

https://drive.google.com/file/d/1ZrR1zcY6VAdVW3hnMxSqOzXUThfdzYO5/view?usp=sharing

Data Analysis



In this case, it was tested how the number of gems could influence the time and movements taken by the agents. This was tested with 3 agents in a 15 x 15 display case; for this we can observe that the time taken to obtain the gems was increasing almost linearly. On the other hand, for the number of movements, this can look a bit strange, but it is because when the agent moves to one gem after being localized, this counts as one unique movement. On the other hand, moving one square of the grid also counts as one movement, so what ends up happening in this case is that the agents, when going to a gem and returning it to the chest, end up finding more gems, which reduces the number of movements in the grid.

Number of Agents

- **3 Agents in a 25 X 25 display case, searching for 12 gems**

Time Taken (s)	Number of Collisions	Number of Movements

44	35	297
49	36	301
63	39	358
49	33	334
34	30	197

Video of Demonstration

https://drive.google.com/file/d/1wjFTh2cW1sb9OKSAJ3qQJYypUEHuK2xL/view?usp=sharing

- **4 Agents in a 25 X 25 display case, searching for 12 gems**

Time Taken (s)	Number of Collisions	Number of Movements
39	49	245
41	45	247
46	64	344
48	51	228
36	49	236

Video of Demonstration

https://drive.google.com/file/d/1COePuztsM6uNBVPq18ZCt4HeoFl0fxOn/view?usp=sharing

- **5 Agents in a 25 X 25 display case, searching for 12 gems**

Time Taken (s)	Number of Collisions	Number of Movements
32	51	298
35	47	333
39	50	370
29	52	249
24	39	208

Video of Demonstration

https://drive.google.com/file/d/1wp-PdY4AzbvcI_avIHer_8OOE0yI6EjK/view?usp=sharing

- **6 Agents in a 25 X 25 display case, searching for 12 gems**

Time Taken (s)	Number of Collisions	Number of Movements
21	51	255
32	77	368
30	59	319
29	57	368
20	37	176

Video of Demonstration

https://drive.google.com/file/d/1LTT_TcA4GDOgTus6tbTHAQQ8b8h4zftJ/view?usp=sharing

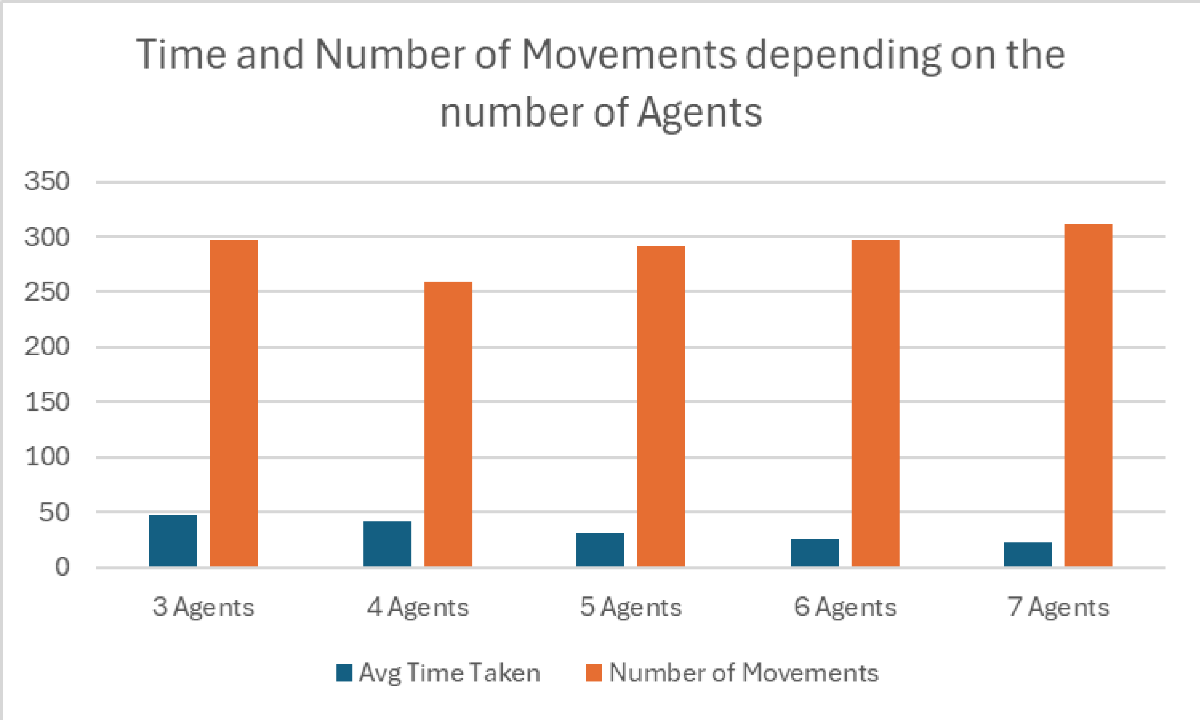
- **7 Agents in a 25 X 25 display case, searching for 12 gems**

Time Taken (s)	Number of Collisions	Number of Movements
30	60	402
19	51	205
21	61	280
20	51	318
27	67	350

Video of Demonstration

https://drive.google.com/file/d/1fbWwvMABad8VIF6PNMZqQwtBFJKRGKie/view?usp=sharing

Data Analysis



In this graph it can be observed how the number of agents can improve the time taken for organizing the same jewelry stand. The graph shows that a greater number of agents assigned to complete a task would reduce the amount of time taken. This happens because the agents are constantly sharing information with one another, which helps them find all the gems. Also, because of the number of agents, the agents can organize themselves better and complete the tasks faster. It can also be seen that the number of movements almost stayed the same in all simulations. This happens because the grid that the agents need to explore is always the same, which means that although there are more agents to search, they need to search the same area and perform a similar number of tasks.

Conclusions

With this project, we successfully demonstrated that a grid-based, multi-agent system can reliably organize jewels. The approach ensured full coverage of the display case, allowing agents to avoid collisions and collaborate effectively, even though it requires more movements than some more optimized methods. Possible improvements include scanning lines instead of individual cells, prioritizing less crowded areas, or combining delivery and exploration actions. These strategies could reduce both time and movements, particularly in larger grids. Overall, the system achieved its main goal while providing clear directions for future optimization and scalability.

It can also be concluded that we successfully created an environment that could be used to test and develop solutions for organizing the display case. In this case, it was tested whether the size of the display mattered, the number of agents, and the number of gems. In the case of the size of the display case, it can be said that the bigger the display, the more movements and time needed to find and organize the jewels. On the other hand, for the case of the number of agents, although the number of movements needed did not decrease, the more agents there were, the less time it took to organize the display. And in the case of the number of gems, it can be said that the more gems there are, the more time it will take to organize them, but on the other hand, it can reduce the number of movements depending on the quantity.

From this experience, we learned to create models in 3D in order to make simulations more graphically realistic, and we learned to implement and create simulations using Unity 3D. From running the simulation, we learned of the importance of making them and testing them; in this case, we could determine which is the best number of agents for different display cases, for example. Which in real life could mean preventing problems and saving money. Through the use of agents, we learn the importance of sharing information between agents, cooperation, and the implementation of autonomous agents, which are able to fulfill a specific task.