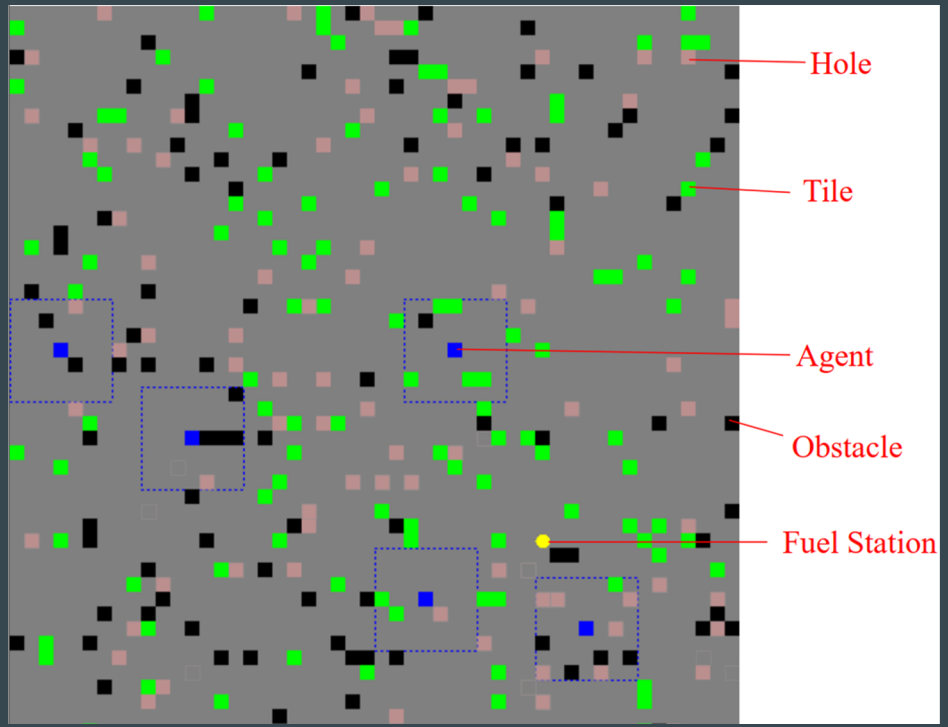
AI6125: Multi-agent System

Group 7: Group Project Presentation

Team Members: Tian Meng
Liu Qixuan
Zeng Zheng
Zeng Tian

Content

- Basic architecture
  - Memory module
  - Communication module
  - Fuel station searching strategy
  - Basic reasoning procedure to score
- Individual strategies
- Test score

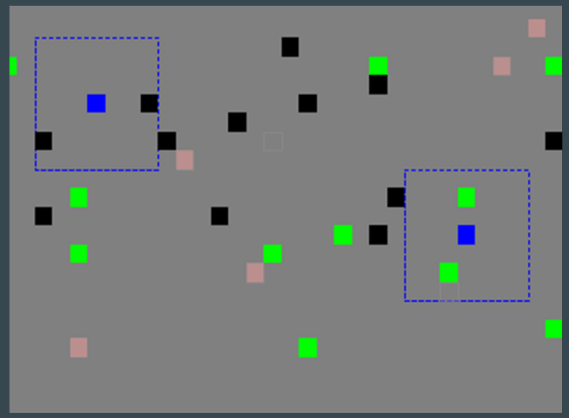


Memory Module

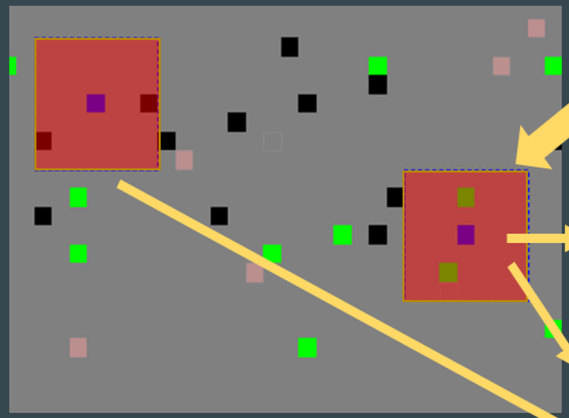
- Motivation
  - To avoid conflicts in acting tasks (e.g. 2 agents pick up the same tile)
  - Also prevent agents from crashing/sticking on void/null Objects

- Design (at each step)
  - Decay memory
    - Traverse all objects in the current memory and remove objects with long lifetime
  - Set all objects within sense range to null
  - Update memory using sensed information from agent self (7*7)
  - Update memory using sensed information from all other agents (7*7)
    - Done by broadcasting sensed info to all agents

Diagram Illustration



$Step_{t-1}$



$Step_t$

Memory Decay

Decay Info at t-1 within sensed range (7*7)

Update Info at t within sensed range (7*7)

Update Info at t within sensed range

Communication Module

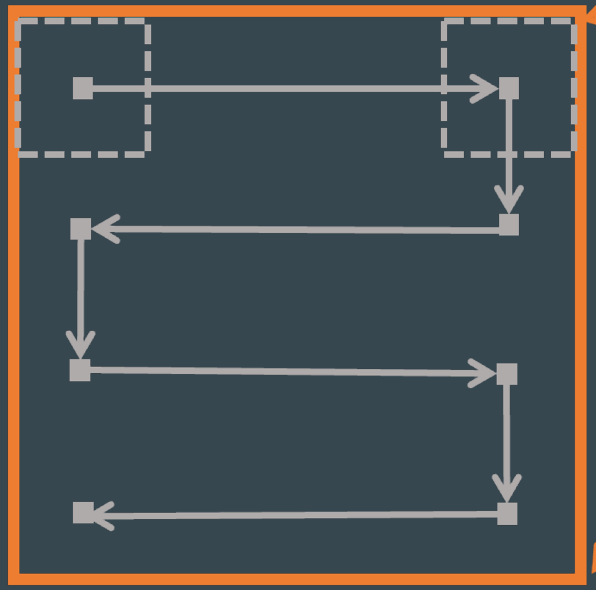
- Motivation
  - To share information among the agents (e.g. the location of fuel station, the sensed objects)

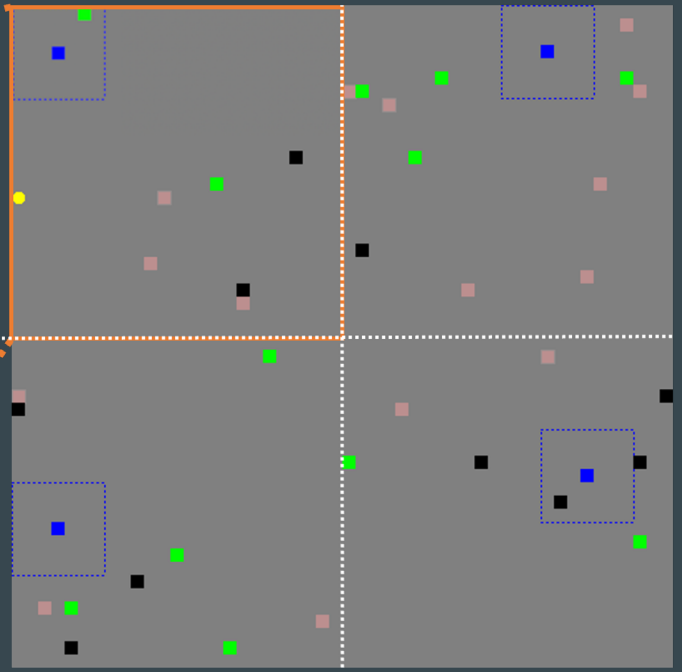
- Design (at each step)
  - Message
    - HashMap<String, Object>
    - Useful tool functions are inside the class (e.g. add the fuel station location and get this location, add sensed objects and get, and add the sending agent's location for other agents refresh the memory by both location and sensed objects).
  - Send messages in the communicate() function
  - Receive messages in the think() function

Fuel Station Searching Strategy

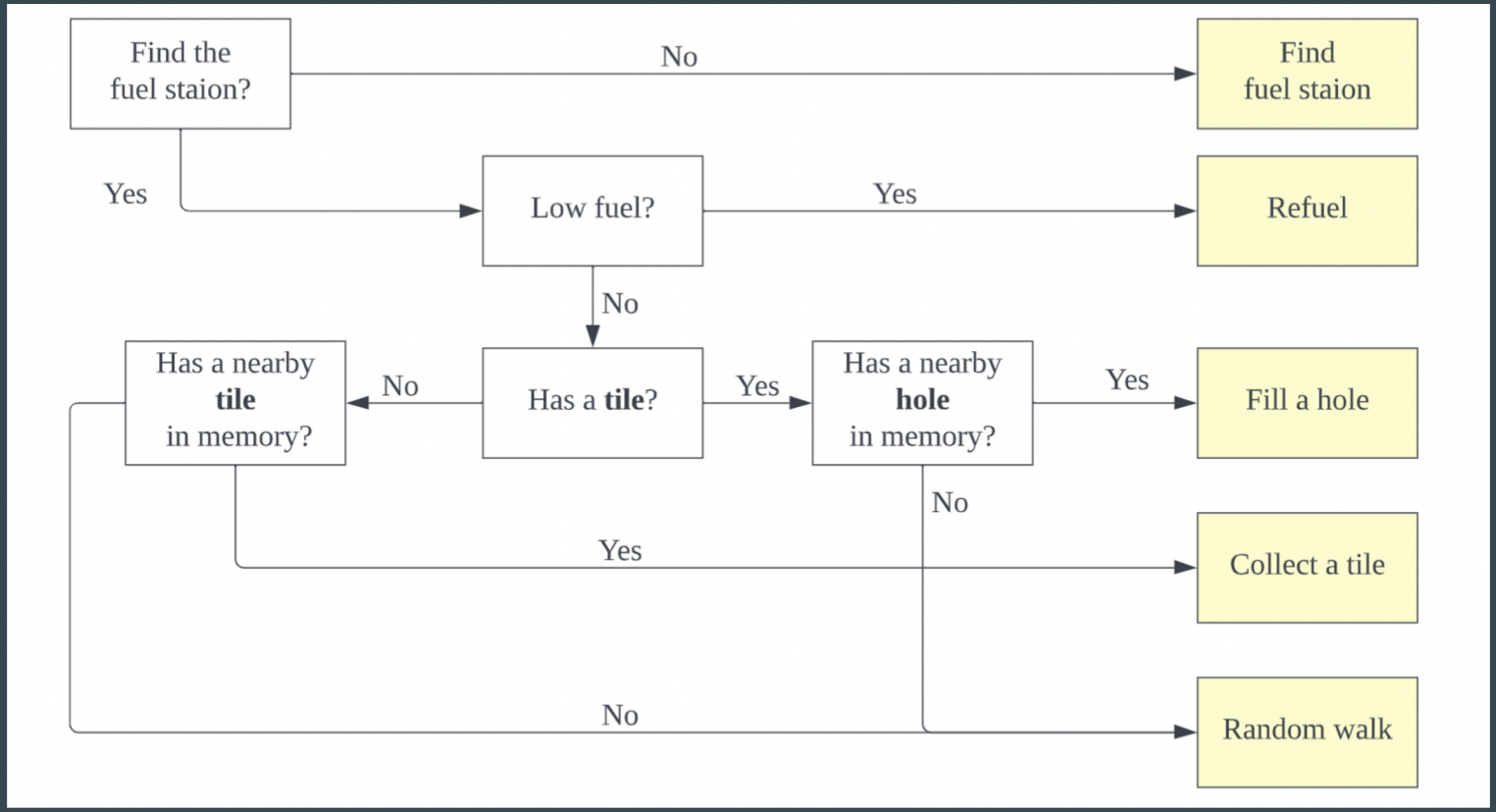
- Each agent is assigned with a small world, and agents search the map parallelly.
- Once one agent finds it, broadcasts it.

Searching route



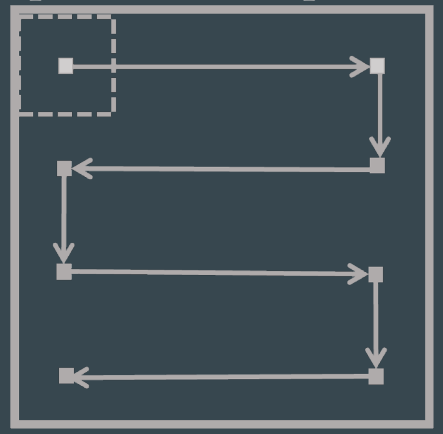


Basic Reasoning Procedure to Score



Agent Architecture 1 - Greedy & Inspector Based MAS

- **Main idea**: three greedy agents scores, and one inspector agent explores the map.
- **Greedy agent: to score**
  - Planning
    - always go to the nearest tile / hole and score.
  - Problem: they always group together.
    - Few resources caused by poor exploration strategy
    - Goal conflict caused by shared memory
- **Inspector agent: to find new resources**
    - To: Avoid issues above.
    - Planning
      - Start inspecting after the fuel station is found.
    - How: Search the map, looking for tiles & holes for teammates.



searching route of Inspector

Diagram: Agent Design

Based on TOURINGMAHINES architecture.

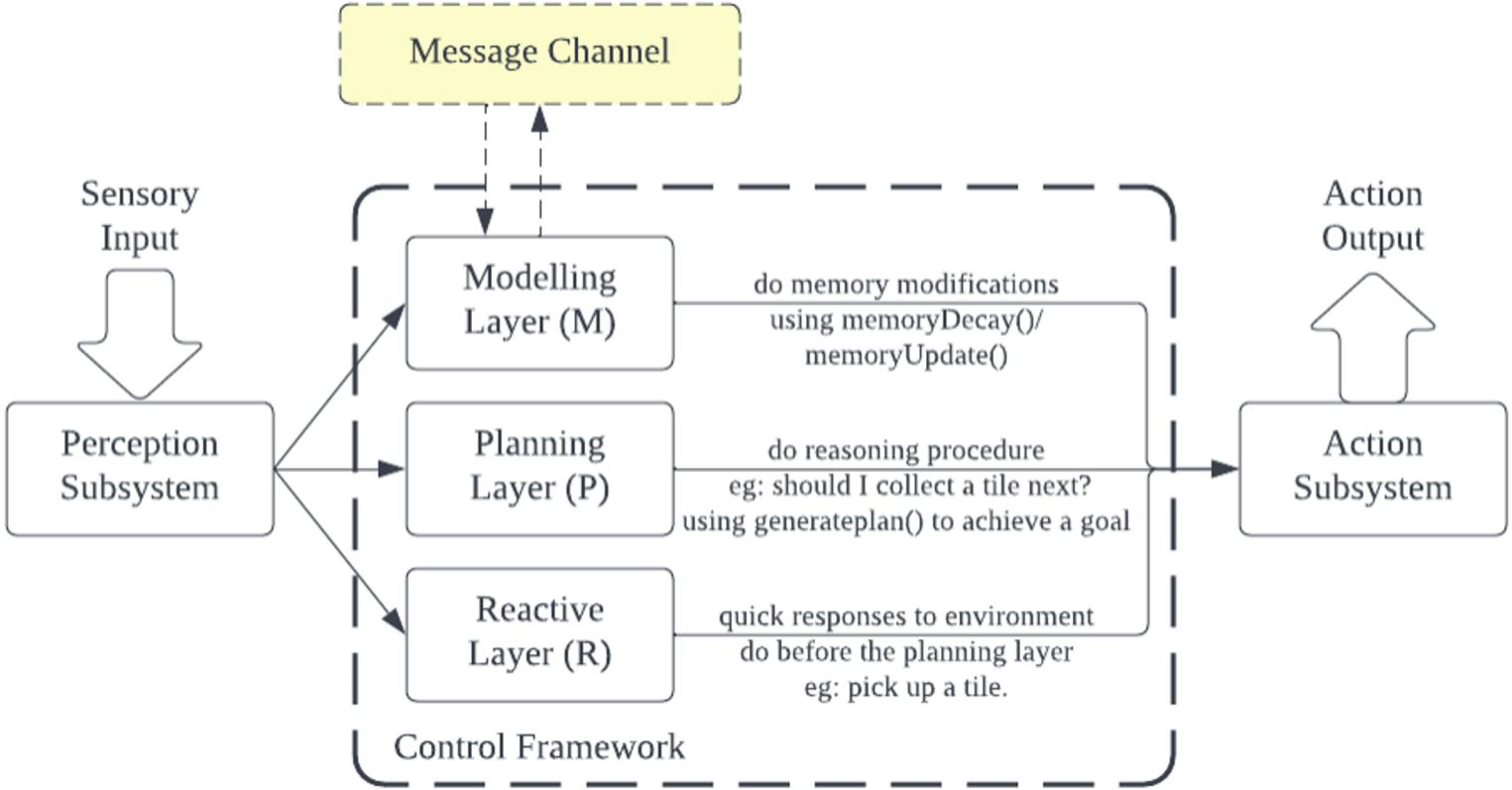
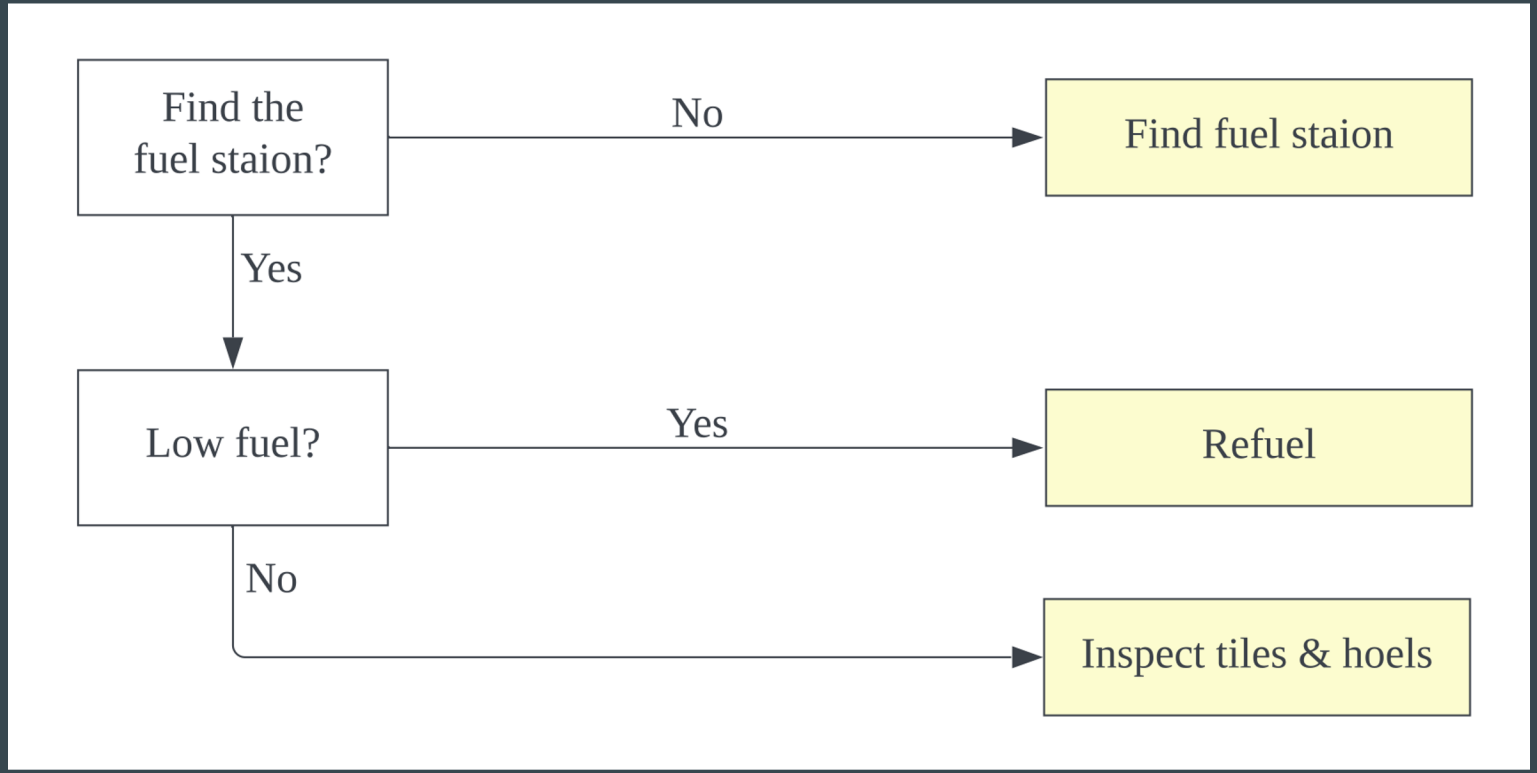


Diagram: Inspector Agent Reasoning Procedure

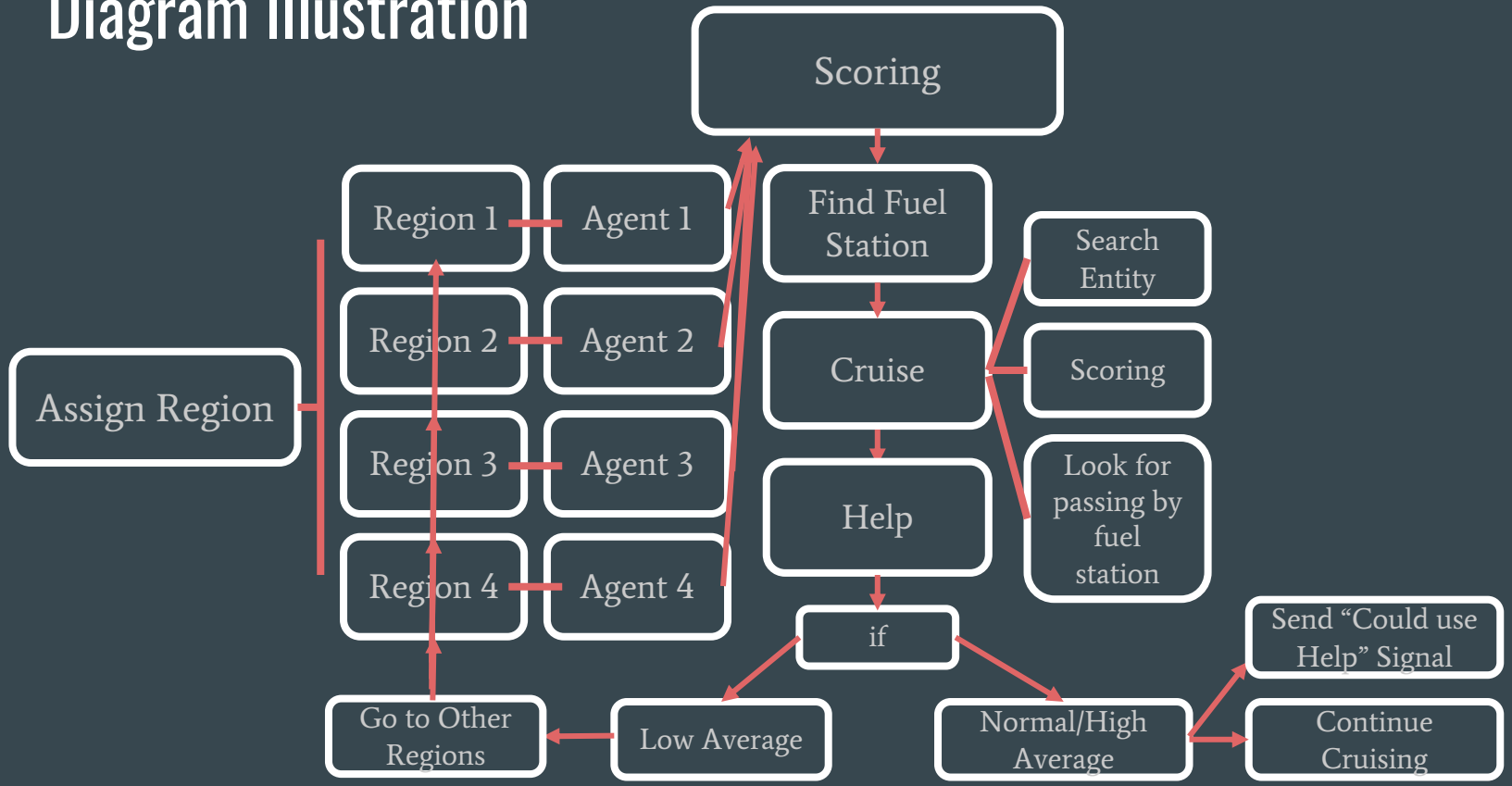


Agent Architecture 2 - Region Based MAS

- Main idea
  - Each agent works on its own assigned region and help others if spare

- Overview design
  - Assign Region - Divide the entire map into 4 regions
    - Each region is of size of (xDim/2, yDim/2)
    - DesignatedPath MODE - Each agent moves to designated region
  - Scoring - Each agent tries to score in its designated region
    - FindFuelStation MODE - Search for fuel Station
    - Cruise MODE - Search for Entity, Scoring, Look for passing by fuel station
  - Helping - If agent is sapre in its region
    - Help MODE - If agent's average score in current region is low, help other regions

Diagram Illustration



Agent Architecture 3 - Anti Goal Confliction and Utility

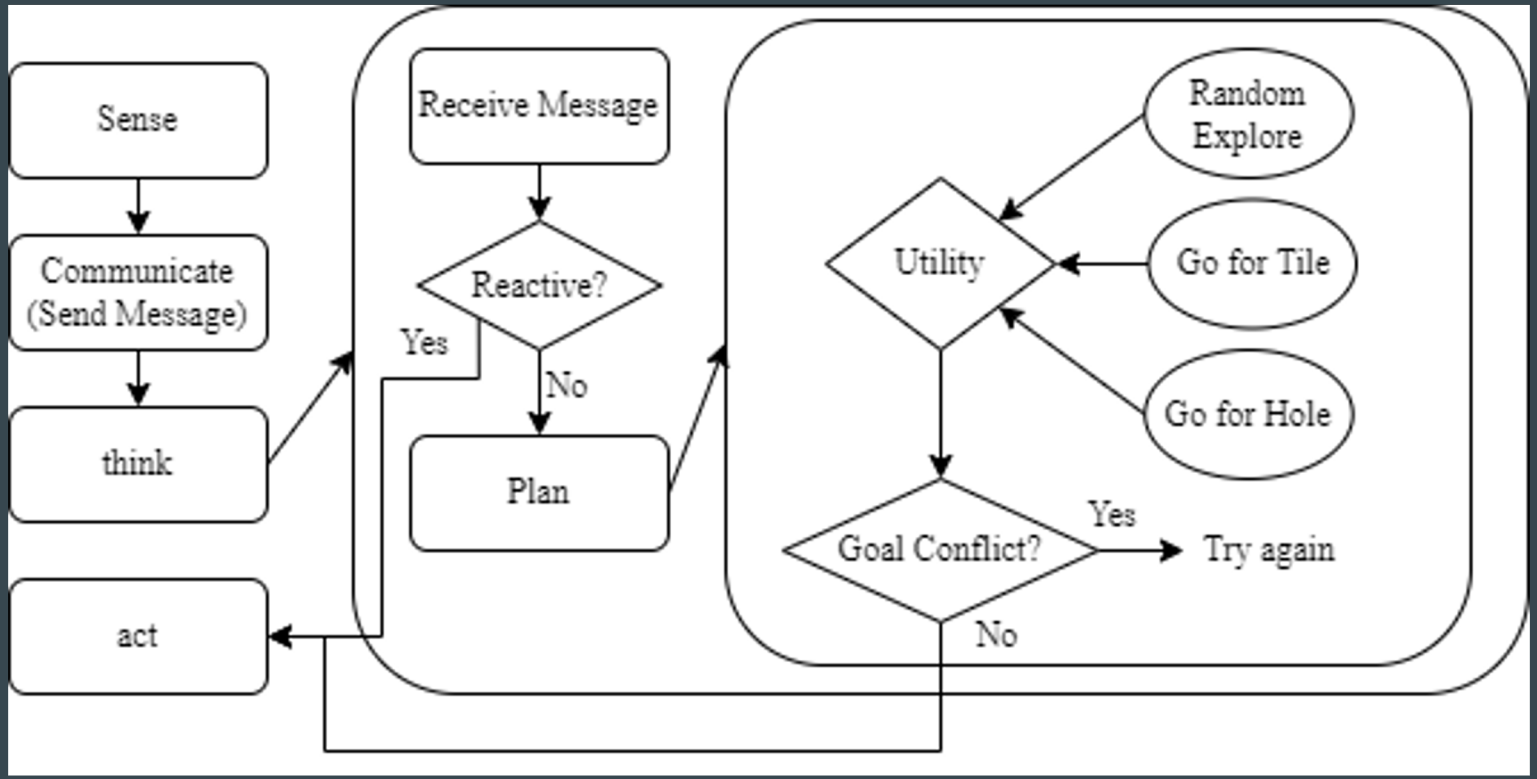
Main Idea

- Every agent are benevolent
- Sharing information about the agents' position, sensed objects, and the goals
  - When agents have a conflict goal, the goal's original owner has the highest priority to continue to own, otherwise it will be occupied according to the time order
- Keep balance between the reactive and goal-oriented behaviour
  - When facing special condition, do the reactive behaviour
  - When facing usual condition, do the goal-oriented behaviour
- Choose the best goal among the random location, the tile, and the hole by using utility functions
  - Utility functions take Parameters into consideration

Other Features

- Overall Strategy: Search Fuel Station → Explore
  - Divide the world into 2 by 2 large grid. For each grid, assign an agent to perform greedy search. After the fuel station is found, do the exploring things.

Flow Chart


Sense
Communicate (Send Message)
think
act
Receive Message
Reactive?
Yes
No
Plan
Random Explore
Utility
Go for Tile
Go for Hole
Goal Conflict?
Yes
Try again
No

Agent Architecture 4: Request & Response

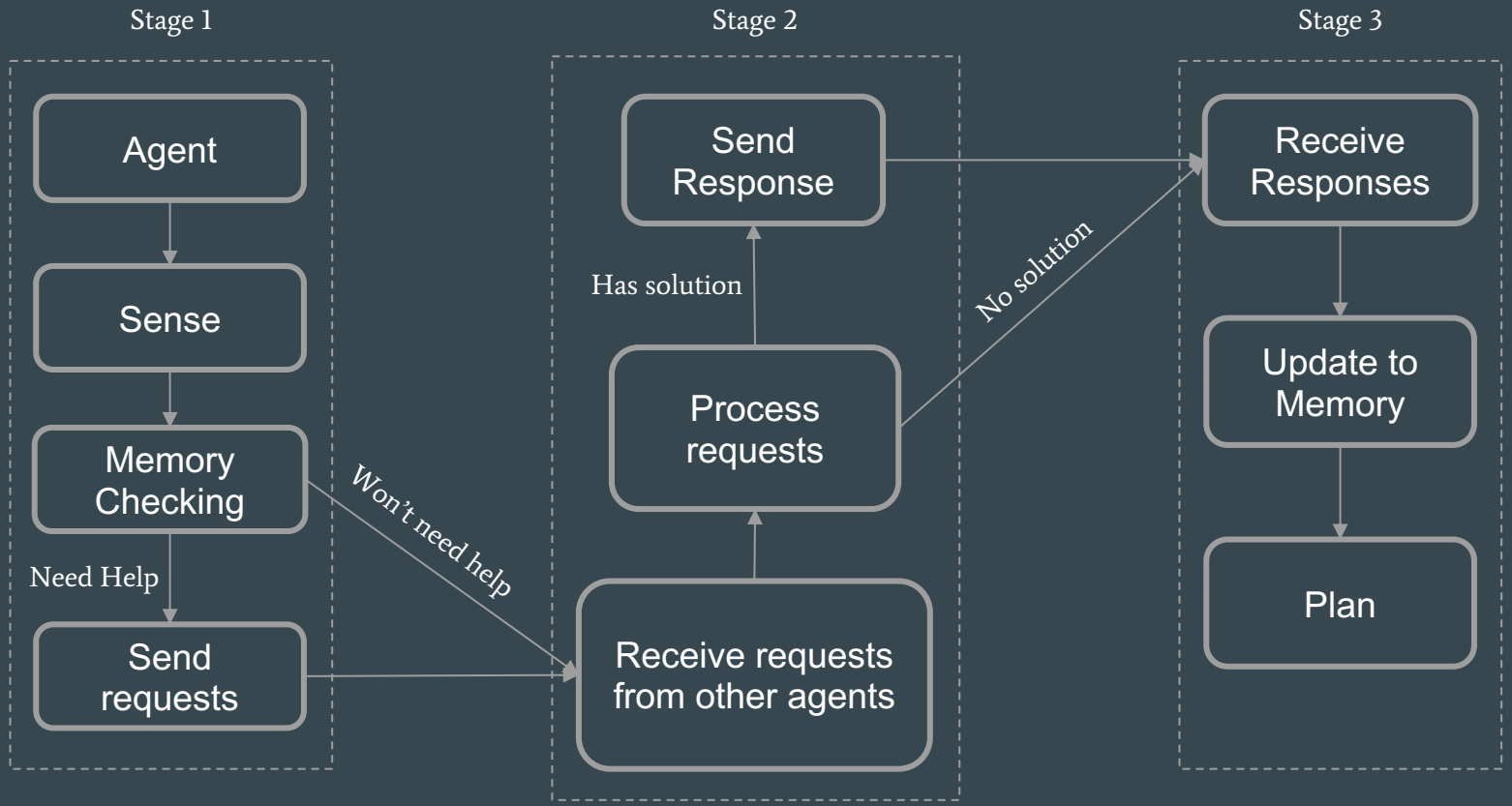
Main Idea

- Every agent are benevolent
- Agents looks for any requirements in the message, and try to give a response.
    - For example, when an agent has carried 3 tiles but it doesn't have available hole information in its memory, it will send a request.
    - Other agents will take the request and try to reply with a solution.

Other Features

- Deal With Goal Conflict
    - When agents have a same goal, only the closest one to the destination will preserve the goal.
- Search Fuel Station
    - Divide the board into 2 by 2 large grid.
    - For each grid, assign an agent to perform greedy search.

Flow Chart


Stage 1
Agent
Sense
Memory Checking
Need Help
Send requests
Won't need help
Stage 2
Send Response
Has solution
Process requests
Receive requests from other agents
No solution
Stage 3
Receive Responses
Update to Memory
Plan

Performance (Four agents, average final score)

Design	Configuration 50 x 50	Configuration 80 x 80
Design 1 - Greedy & Inspector Based MAS	271.0	305.0
Design 2 - Region Based MAS	113.5	273.8
Design 3 - Anti Goal Confliction and Utility	256.5	499.2
Design 4 - Request & Response	378.3	540.0

Performance (Four agents, total average score)

Configuration	Design 4
50 x 50	378.3
80 x 80	540.0
*70 x 70 parameter 1 setting	253.4
*60 x 120 parameter 2 setting	475.0

DEMO

Thank you!