|  |
| --- |
| **University of Southern Denmark – Course AM24** |
| Multi-Agent Systems |
| Planet Exploration |
| **Group 3** |
| **Martin Moghadam, Kalle Grafström, Morten Knudsen** |
| **12-08-2010** |

|  |
| --- |
|  |

Table of Contents

[Introduction 1](#_Toc269389331)

[Design 2](#_Toc269389332)

[Constraints and Parameters 3](#_Toc269389333)

[Queues and Jobs 3](#_Toc269389334)

[Messages 3](#_Toc269389335)

[Base 3](#_Toc269389336)

[Explorer 3](#_Toc269389337)

[Perception and Warping 4](#_Toc269389338)

[Send to Transporter 4](#_Toc269389339)

[Transporter 4](#_Toc269389340)

[Messaging and Communication 5](#_Toc269389341)

[Experiments 5](#_Toc269389342)

[Discussion and Further Development 5](#_Toc269389343)

[Conclusion 5](#_Toc269389344)

[Appendix 5](#_Toc269389345)

# Introduction

A project created for the summer course of AM24 – Multi-Agent Systems, using Madkit and Turtlekit, developed with Java (Java JRE6).

The project involves multiple agents for planet exploration with robots; ore is harvested from the planet and stored in bases, using explorers to find the ore, and transporters to move the ore to the base. The explorers and transporters have a limited amount of energy available, when energy is almost depleted they return to base to recharge. Energy is consumed by actions; move, send message, perceiving the environment. If a robot depletes the energy before it can recharge the robot dies. The explorers have a limited perception scope and can only detect ore that is nearby. The transporters can pick up the ore when they are at the same position. Each base has a limited capacity of ore, and each transporter can carry a limited amount of ore. Robots return to base when; the ore has filled the base, or the time has run out.

The table below shows the parameters of the project and the associated symbols which are used throughout the report.

|  |  |
| --- | --- |
| Base Capacity | C |
| Ore Density | D |
| RobotEnergy | E |
| Grid Size | G |
| Mode | M |
| Number of Bases | N |
| RobotPerceptionScope | P |
| RobotCommunication Scope | I |
| Robot Memory Size | S |
| Max Simulation Time | T |
| Transporter Ore Capacity | W |
| Explorer Amount | X |
| Transporter Amount | Y |

The motivation of the project was:

* Becoming familiar with Madkit and TurtleKit, understand the tools and examining the online documentation and examples.
* Learning to develop Multi-agent systems and creating communication between the agents.
* Creating smart agents that are not just reactive, but proactive and cooperative.
* Testing, experimenting and improving the agents, and examining solutions of the other groups to see the different possibilities and gain experience.

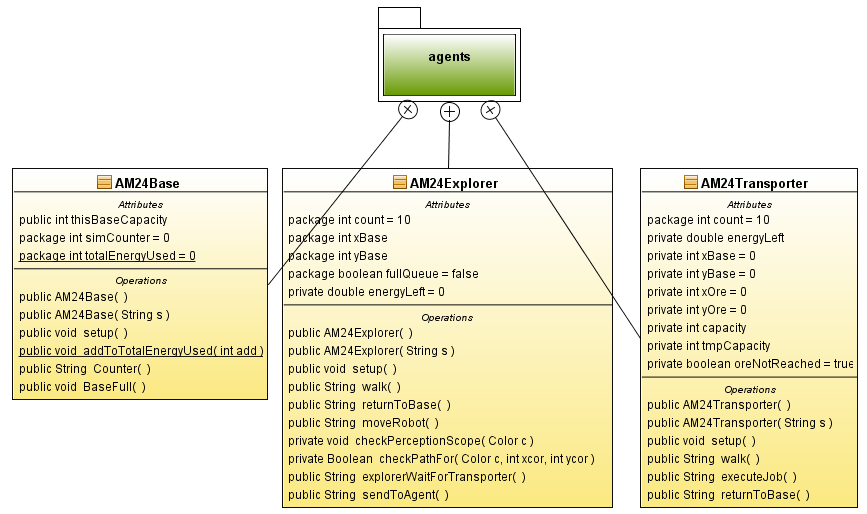
The report documents the development and experiments, then discusses the results, examines further development possibilities and draws a conclusion.

# Design

The project consist of three packages; launcher, util, agents.



* The launcher package contains the initialization of the project;
  + Creating the ore as pink patches on the grid.
  + Adding the simulation agents.
* The util package contains the utility classes;
  + The constraints and parameters.
  + Queue for managing jobs.
  + Messaging system for communication between the agents.
* The agent package contains the agents;
  + The base agent.
  + The explorer agent.
  + The transporter agent.



The class diagram shows the agents and the agent package, which is examined in the following section.

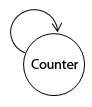
## Constraints and Parameters

## Queues and Jobs

## Messages

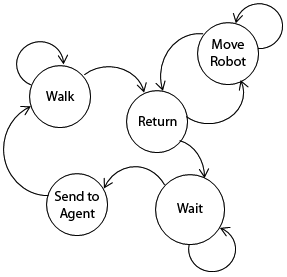
# Base

* Total Energy Used
* Counter
* Base Full



# Explorer

The state diagram shows the different state of the explorer.



* Walk; the explorer walk through the environment
  + As the explorer move it checks the perception scope for ore
* Return; When queue is full of job the explorer returns to base
  + The explorer move robot
* Wait; The explorer waits for a transporter agent and then;
  + Sends the queue of job to the agent
* The explorer then continues to walk

## Walking

The explorer moves forward in a random heading, and after moving in that direction for a period according to a counter the direction is changed to a new random heading. The pattern is a lot like are termite, and the termite demo from Turtlekit was examined for inspiration. When the explorer moves it check the perception scope for ore.

## Perception

The explorer checks for ore by examining the patches within the range of perception, the explorer iterates though the different patches within its perception and checks if the color is that of the ore. When ore has been found a job is created and added to the queue for harvesting.

## Warp

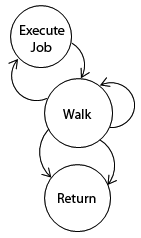
When warp is enabled finding ore also warp around the planet, if the explorer is located near the edge of the grid the perception can see the other side of the planet when it is within range.

## Energy and Time Cost

## Send to Transporter

When the queue is full and the transporter is at the base; the jobs in the queue are polled and sent to a transporter agent, which is selected at random.

# Transporter



## Messaging and Communication

# Experiments

|  |  |  |
| --- | --- | --- |
| Base Capacity | C | 200 |
| Ore Density | D | 0.05 Uniform distribution |
| RobotEnergy | E | 5000 |
| Grid Size | G | 200x200 |
| Mode | M | Cooperative |
| Number of Bases | N | 1 |
| RobotPerceptionScope | P | 5x5 |
| RobotCommunication Scope | I | 11x11 |
| Robot Memory Size | S | 15 |
| Max Simulation Time | T | 10000 |
| Transporter Ore Capacity | W | 8 |
| Explorer Amount | X | 10 |
| Transporter Amount | Y | 10 |

# Discussion and Further Development

# Conclusion