

BW4T3 Instructions

W. Pasman, M. B. van Riemsdijk,



Contents

1	Introduction	3
1.1	System requirements	3
1.2	Installation	3
2	Running the server	4
2.1	Advanced run settings	4
3	Running the client (Human Controlled Bot)	6
4	Running the client (Agent Controlled Bot)	6
4.1	Advanced run settings	6
5	Restarting, pausing and resuming the system	8
6	Loading and creating new maps	9
6.1	Using the Environment Store	9
6.2	Manual editing	10
7	Using the interface	11
7.1	Sending messages	11
7.2	Click on room or dropzone	12
7.3	Click on blocks	12
7.4	Click on hallway	12
8	Running on multiple computers	12
8.1	Distributed human GUIs	13
9	Programming a BW4T Agent	13
9.1	Testing your agent	14
10	Creating your own Scenario	14
10.1	General use	15
10.2	Configuration Panel	16
10.2.1	New configuration file	16
10.3	Open configuration file	16
10.3.1	Save configuration file	17
10.3.2	Save configuration file as	17
10.3.3	Export configuration to mas2g	17
10.3.4	Exit the Scenario Editor	17
10.4	Entity Panel	17
10.4.1	Create new bot	17
10.4.2	Modify bot	18
10.4.3	Rename bot	18

10.4.4	Change controller type	18
10.4.5	Change the amount of entities of a bot	19
10.4.6	Delete bot	19
10.4.7	Create new e-partner	19
10.4.8	Modify e-partner	19
10.4.9	Rename e-partner	19
10.4.10	Change the amount of entities of an e-partner	19
10.4.11	Delete e-partner	20
10.5	Bot Store	20
10.5.1	Rename bot	20
10.5.2	Change controller type	21
10.5.3	Change the amount of entities	21
10.5.4	GOAL options	21
10.5.5	Bot properties	21
10.5.6	Save modifications	21
10.5.7	Reset modifications	21
10.5.8	Cancel modifications	21
10.6	E-partner Store	22
10.6.1	Rename e-partner	22
10.6.2	Change the amount of entities	22
10.6.3	GOAL options	22
10.6.4	E-partner properties	23
10.6.5	Save modifications	23
10.6.6	Reset modifications	23
10.6.7	Cancel modifications	23
11	Log file	23

1 Introduction

Blocks World for Teams (BW4T) is a testbed for team coordination. BW4T allows for games with human-human, agent-agent and human-agent teams of variable sizes. The goal is to jointly deliver a sequence of colored blocks in a particular order as fast as possible. A complicating factor is that the players cannot see each other.

This document describes how to install Blocks World For Teams (BW4T) for use with GOAL. We briefly comment where necessary how it would work with a different agent system. You can also use BT4T stand-alone but this is not described in this document. BW4T is a client-server system. The server is responsible for the administration, simulation and visualization of the virtual world: it keeps track of robots, rooms, blocks, connected GOAL agents, etc. The server uses Repast, software to simulate virtual environments, to do part of this administration. The client is GOAL, which runs a multi-agent system (MAS) and connects to the server. The agents in the GOAL client get percepts from the server, and send actions to the server. Client and server can run on a different computer. This document describes how to install the BW4T server, configure it, place the BW4T client in GOAL, configure the MAS file, and run the system.

This chapter describes how users can install and use the Blocks World for Teams environment. For developer information please go to the project page on github <https://github.com/eishub/BW4T> and click on the Link to Developer Details.

We use the following names to refer to directories of BW4T:

- <GOAL> refers to the directory where you installed GOAL.
- <SERVER> refers to the directory where you have put the server.jar, documentation and utilities.
- <CLIENT> refers to the directory where you have put the client.jar.

1.1 System requirements

To use BW4T you need Java JDK 7 or higher. The BW4T3 environment has been tested on Windows 7, Windows 8 and OSX. Note that OSX may come with java 6 pre-installed. You need to install java 7. You can then keep java 6 installed as well, but ensure that java 7 is used as the default.

1.2 Installation

You need to install a number of items to run BW4T from GOAL.

1. (if not yet done): Install Java 7 and made it the default Java (such that double clicking on a jar makes it open with Java 7).

2. You can download GOAL from [http : //ii.tudelft.nl/trac/goal](http://ii.tudelft.nl/trac/goal). You can choose for the plugin for Eclipse or the stand-alone SimpleIDE.
3. GOAL provides a sample Multi-agent-system 'BW4T3' that is demonstrating the Blocks World for Teams system.¹
4. You can download the BW4T server from [https : //github.com/eishub/BW4T/releases](https://github.com/eishub/BW4T/releases) and save it in <SERVER>. Select the latest *bw4t-server-X.Y.Z.jar*.
5. You can optionally download the BW4T map editor from [https : //github.com/eishub/BW4T/releases](https://github.com/eishub/BW4T/releases) and save it in <SERVER>. Select the latest *bw4t - environment - store - X.Y.Z.jar*.
6. You can optionally download the BW4T scenario editor from [https : //github.com/eishub/BW4T/releases](https://github.com/eishub/BW4T/releases) and save it in <SERVER>. Select the latest *bw4t - scenario - editor - X.Y.Z.jar*.

2 Running the server

Run the server before running the client, as otherwise the client cannot connect to the server. Start the server by opening (typically: double clicking) the *server.jar* in <SERVER>. This should open the server window (Figure 1). Note that during this opening, two other maps are created:

- Maps: The folder in which all possible maps are put.
- Log: The folder in which all log files will be placed.

2.1 Advanced run settings

The default settings for the server can be changed in the command line of your OS. To change the server ip and/or server port execute the following:

```
java -jar bw4t-server.jar -servip <server ip here> (default: localhost)
-serverport <serverport here> (default: 8000)
```

If you change the *servip* and/or *serverport*, change the client settings correspondingly (see below). The server will now start using the new values.

¹If you run in a different agent system, you can download the client-side environment from [https : //github.com/eishub/BW4T/releases](https://github.com/eishub/BW4T/releases). Select *bw4t - client - X.Y.Z.jar*. This is the environment that should be attached to your agent system. You then will have to write your own agents that can run in the BW4T world.

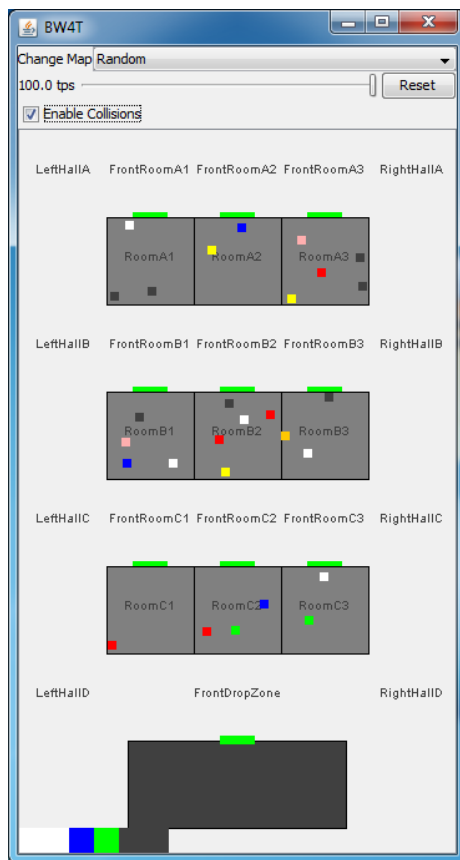


Figure 1: Server window

3 Running the client (Human Controlled Bot)

Before running the client, make sure you have already started the server. Start the client.jar in <CLIENT>. Figure 2 will show up and the bot is automatically added to the server. You can now control the bot by clicking (left or right) at different spots in the Client.

4 Running the client (Agent Controlled Bot)

Before running the client, make sure you have already started the server. Start Eclipse IDE and load your GOAL files. Open the .mas2g file and press the run button. This time, no new window will appear. The agent will control your bot and all evaluations will be shown in the console. In the server window you will see your bot(s) move according to the rules applied in the .goal file(s).

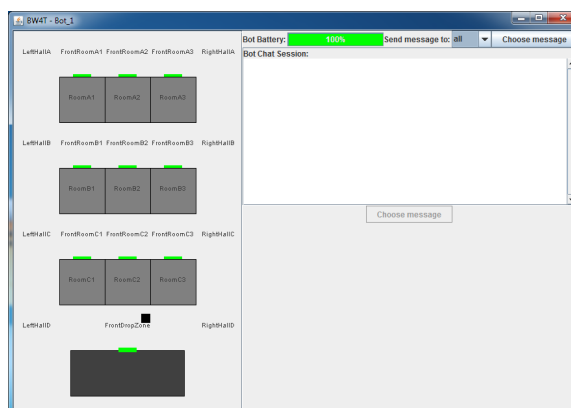


Figure 2: Client window

4.1 Advanced run settings

The following MAS init parameters are available (the default values are shown inside the brackets). In GOAL these settings can be changed from the mas2g file. Most of these parameters are not needed if you use an agent config xml file. The CONFIGFILE option is needed if you use a config file. The HUMANGUI option is needed if you have a human GUI connected with an agent (and the agent should receive percepts as well).

- CLIENTIP("localhost"): client's own IP. Passed to server so server is able to find us with this IP.
- CLIENTPORT("2000"): The port that the client listens to.
- SERVERIP("localhost"): The IP address of the server.
- SERVERPORT("8000"): The Port that the server listens on.
- AGENTCOUNT("0"): the amount of agents (also specified in the launchpolicy section, see below), that the client should load. If the

agentcount is higher than the amount of entities in the map then they won't be loaded. (default: 0)..

- `LAUNCHGUI("true")`: whether to launch a separate GUI for each bot (controlled by an agent or human) can be set to true or false. This GUI shows the environment from the perspective of the bot. (default:false).
- `HUMANCOUNT("1")`: the amount of human players that should be loaded. If the humancount is higher than the amount of entities in the map then they won't be loaded.
- `AGENTCLASS("nl.tudelft.bw4t.client.agent.BW4TAgent")`: The java agent class to load when new entities appear.
- `GOAL("true")`: are we connected with GOAL? This param should be auto detected, it will be set to false if the program is started from commandline.
- `KILL("")`: The key we should try to use to kill the remote server.
- `CONFIGFILE("")`: The file from which the client reads the configuration.
- `GOALHUMAN("false")`: Forces the use of the human GUI with an GOAL agent to translate the commands. You should turn on this option if you have a human GUI but still want the agent to receive the percepts.
- `MAP("")`: The map name to be loaded. If you specify a map, the server will reset to load the new map, which disconnects all entities.
- `LOG("ERROR")`: The log4j log level to be used. Available values: OFF, FATAL, ERROR, WARN, INFODEBUG and ALL.

The number of agents is specified at two places in the mas2g file. First, the agentcount and humancount specify the number of entities of the corresponding type that should be created in the environment. Second, the launchpolicy specifies which and how many agents should be connected to these entities. Make sure that the agentcount and humancount in the initialization parameters are in line with the launch policy section in the mas2g file. Furthermore, the number of agents should not exceed the maximum number of agents defined on the map (see section "Loading and creating new maps" below). If you use BW4T from a batch runner, you may want to reset the server after each run. This is done by specifying a map in the mas file init parameters.

5 Restarting, pausing and resuming the system

If you are running from SimpleIDE, you can pause and resume the system at any time by clicking the pause or step buttons. When running from Eclipse, to be able to pause the system, don't run the mas2g file as described above. This time, open the mas2g file and click on the debug button (the small bug icon).

Eclipse will ask you whether you would like to go to the debug screen (see figure 3), click yes and watch your screen adjust. Now you can see the running bots listed at the left side, and the current goals, beliefs and knowledge of the selected bot on the right side. Beneath the list the goal file belonging to the bot is shown and at the bottom of the screen, all debug actions are being run. See figure 4. To pause the system, select the bot to be paused and click on the pause button. To resume, select the bot and click resume. To restart the system, do the following

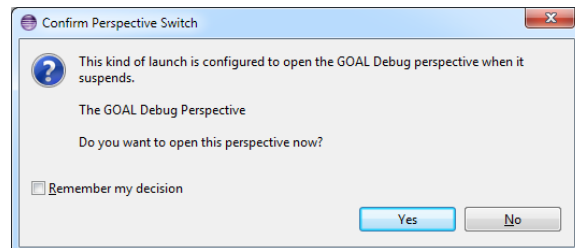


Figure 3: Choose yes

1. In eclipse, kill the MAS by clicking the red box (stop button) at the right of the console window.
2. In the server window, press Reset, or choose a new map from the Change Map menu.
3. Run the MAS in Eclipse as described above.

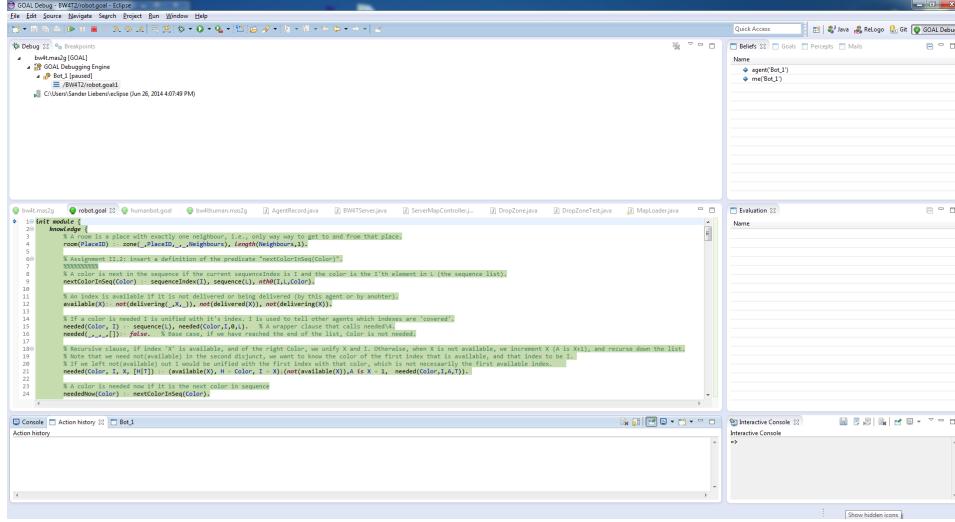


Figure 4: Debug mode

6 Loading and creating new maps

6.1 Using the Environment Store

The Map Editor is a tool for editing maps for the BW4T server. Double click the jar file `environment-store.jar` to start the map editor. After starting up, the map size dialog (Figure 5) appears, which can be used as follows. Here you can choose the amount of rows and columns you want the environment to consist of. Each cell in the “table” can be used to fulfill one piece of the map. This could either be a room, a corridor, a dropzone, a charging zone, start zone or a blockade. By pressing the colours at the bottom you can generate the sequence to be completed. This can also be done by pressing the according numbers on the keyboard.

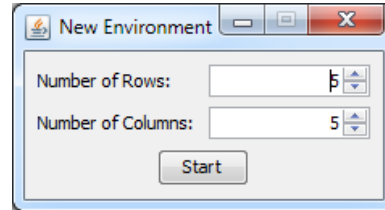


Figure 5: Choose the amount of columns and rows

In the Map editor GUI above you can add blocks to rooms. To add blocks to rooms, right click in the cell, adjust the type of the zone to Room. Now you can add blocks the way you did with the sequence. Right clicking the cell again lets you choose at which side of the room you want the door. When you’re done, press save. By pressing Tools in the menu bar, you are able to randomize zones, blocks and sequences. WARNING: Randomizing these

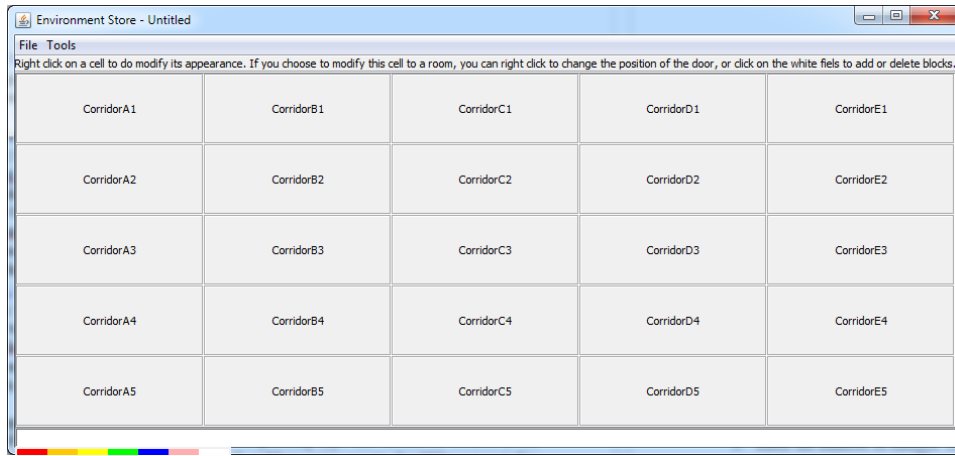


Figure 6: Create a map

will not always guarantee that the sequence can be completed. By pressing File in the menu bar, you are able to watch a preview of your map, and you are able to save the map. Save the map in the `<SERVER> /maps` folder ¹.

6.2 Manual editing

It is also possible to manually edit a map as it is a plain text XML file.

1. Copy an existing map file in the `<SERVER> /maps` folder to a new file ¹.
2. Open the copied map with a text editor. You can then edit the colors in the rooms. Each `<blocks>COL< /blocks>` line inside a `<zones>` of type ROOM adds another block to the room. The currently available colors are BLUE, ORANGE, RED, WHITE, GREEN, YELLOW AND PINK. A room has place for at most 10 blocks.
3. You can change the number of rooms by adding or removing `<zones>` of type ROOM and position the rooms correctly on the map by editing the `<x>` `<y>` `<width>` and `<height>` inside the room's `<boundingbox>`. Also you need to add a `<door>` properly positioned on the border of the room.
4. You can edit the goal sequence by adding `<sequence>` items to the map, with colors as mentioned above.
5. You can prevent multiple entities to enter corridor zones by putting true in the item `<oneBotPerCorridorZone>` in the map.

¹This map directory appears after the first run of the server.

6. You can add entities to the environment by creating more `<entities>` items. Make sure they have a unique `<name>` and that their start position is in a different `<zone>` if you have `<oneBotPerCorridorZone>` set to true.
7. You can let the server pick a random sequence of a given length by setting the `<randomSequence>` to a positive value. These random blocks are added to the existing sequence items and the random blocks are placed randomly on the map.
8. You can let the server pick random extra blocks to be placed in the rooms on the map by putting a positive number in the `<randomBlocks>` in the map. Note that this addition is on top of all blocks that are already placed in the map; so normally you leave the room zones empty when using this option.
9. You can set per-zone visibility of the zone namelabel in the server map renderer, using a `<renderOptions>` `<labelVisible>` false `</labelVisible>` `</renderOptions>` block to disable visibility.
10. Save the map in the `<SERVER>` `/maps` directory.
11. Edit the map initialization parameter for the server to your new map file. See the section on customizing the server settings. If you now start the server and your new map should be loaded.

7 Using the interface

To use the interface, the user clicks with the left (occasionally the right) mouse button in the GUI. Depending on where the user clicks, different menus appear. The user then picks the appropriate action from the menu to execute that action. Below the possibilities are explained.

7.1 Sending messages

By clicking on the dropdown menu, the user can choose to which bot he or she will send a request or question. “all” sends the message to all bots. Next to the dropdown menu, a button is placed with choose message. This button contains the messages that are most likely to be sent. Left clicking this button will create a list from which the user can choose a message. When clicking in the “Bot Chat Session”, gives a number of standard answers: yes, no, don’t know, ok, etc. It is not possible to use free text because the non-human agents can only process these pre-specified messages. The specification document gives more details on how messages are processed by non-human agents.

7.2 Click on room or dropzone

By clicking on a room, the user can order his bot to go to that room, tell everybody something about that room or ask something about that room.

7.3 Click on blocks

By clicking on a block (particularly, those below the drop zone), the user can tell everybody something about that block, or ask others for information about that block.

7.4 Click on hallway

By clicking on a hallway, the user can point to an exact (X,Y) location to go to. Also it is possible to tell all others roughly where one is in the hallway.

8 Running on multiple computers

If you want to run BW4T on multiple computers you should designate one of these computer as the server. On this computer you can start the server as described in the first section of this guide. You must use RMI messaging (check the GOAL Run menu) to allow other GOAL runtimes to connect . You need to check a few things in the MAS that you use here:

- specify a map such that the environment resets when you start up the MAS
- make sure that the map that is used has enough entities to accommodate all agents in all computers that want to connect
- make sure that the entities get the proper type, by specifying the proper agentcount and humancount.

The other computers will then function as client. Create a MAS file for each of these, and configure this MAS as follows (see also bw4thuman.mas2g in the GOALagents directory of GOAL):

- set the serverip and serverport initialization parameters to the ones that the server is listening on (default for the server is localhost and port 8000).
- Set the humancount and agentcount parameter on each client to reflect how many human or agent players that client should load.
- Use humanbot.goal for human agents

- use `env = <CLIENT> /client.jar`. in the environment section. Do not connect to an already running environment in another MAS. This is because BW4TClient creates GUIs for humanbots, on the machine where it is running.
- Check that the launchpolicy picks up the proper entity type, so use 'human' if you want to attach to human entities etc.

8.1 Distributed human GUIs

This section describes how to run a set of distributed human GUIs such that they all communicate through GOAL. Before running a set of distributed Human GUIs with Eclipse, make sure that the server is installed on one machine. Furthermore, make sure that the server map can contain a sufficient number of entities. To run a set of distributed Human GUIs with GOAL, do the following:

- Start the server
- For each machine where you want to have a human GUI:
 1. Start Eclipse IDE
 2. Open the MAS file of the `bw4thuman.mas2g`
 3. adjust the `serverip` to correctly point to the machine ip number where the server runs
 4. Start the MAS

9 Programming a BW4T Agent

To program your own BW4T agent, use the same actions as specified in the `demorobot`. You can choose to change the pre- and post conditions of each action.

Percepts are retrieved automatically by GOAL, see the percept specifications for what percepts you can expect.

In order for messages received by other GOAL agents to appear on the GUI of your agent, you should add the following lines to your GOAL agent's code:

1. Add the following line to your knowledge base:
`#import "messageTranslation".`
2. Add the following line at the end of your goal file.
`#import "message.mod2g".`

3. Add the following line to the start of your event module:
`if bel(true) then message.`
 This will make sure that the message module that was just imported will be run first when entering the event module.
4. Your own message handling code should be performed after this line and should delete any messages after handling them. Otherwise they will be continuously posted to the GUI as they are not deleted by the message module. If you don't do any message handling yet you can add the following line below the one provided in step 3 to delete all received messages:
`forall bel(received(Agt,Msg)) do delete(received(Agt,Msg)).`

9.1 Testing your agent

In order to test your agent you should edit the `bw4t.mas2g` file as follows.

- Add your agent to the `agentfiles` list.
- Add a line to the `launchpolicy` (copy the line of the `demorobot` and replace `'demorobot'` with the name of your agent).

Make sure that you set `GOAL` to launch the desired number of your agent. Also make sure that the initialization parameter of `agentcount` is not set to 0 as then only humanbots will be loaded.

Note that besides in the `bw4t.mas2g` file, the number of agents is also specified in the map that you use. This number determines the maximum amount of bots that can appear on the map. If the maximum number of agents to be launched as specified in the `bw4t.mas2g` file is bigger than the number of agents specified in the map, some of the agents will not appear on the map and not be part of the team.

10 Creating your own Scenario

This part describes how to make use of the Scenario Editor (Figure 7). When you open the editor, you see two main parts:

1. The Configuration panel, which is used to create, open and save configuration files.
2. The Entity panel, where a list of bots and a list of e-partners are being displayed. Here you can create, modify, rename and delete bots and e-partners.

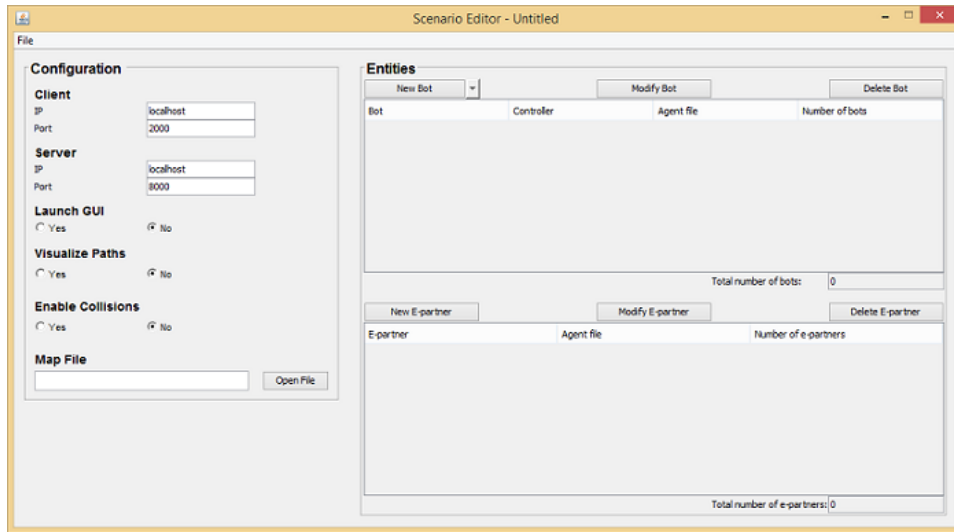


Figure 7: Scenario Editor

10.1 General use

At the top of the editor you can see a *File* menu. If you click on that, you get the options to create a new configuration, open a configuration, save your configuration, export your configuration and to exit the editor.

On the left side of the editor you can configure a scenario as you like. The Client IP, the Client Port, the Server IP and the Server Port should already contain the default values. You can change them if you need to. You can also indicate whether you want to open a GUI, visualise the paths the bots take or enable collisions. If you have a map file you want to use, you can import it by selecting the *Open file* button at the right section.

On the right side of the editor you can create, modify, rename and delete bots and e-partners. There also are a list of bots and a list of e-partners that are created, and below each list there is an indication of how many bots or e-partners are created in total.

Linked to the Scenario Editor are the Bot Store and the E-partner Store, which will also be discussed in this document.

10.2 Configuration Panel

A picture of the configuration panel can be found in figure 8.

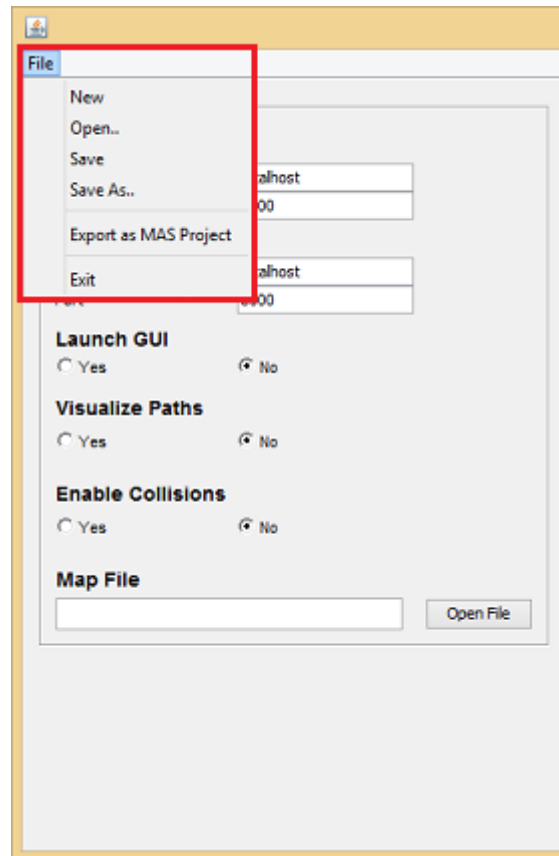


Figure 8: Configuration Panel

10.2.1 New configuration file

To create a new configuration file, select *File* → *New* in the menu bar. You will now get a new configuration with the default values. Creating a new configuration will reset all previous changes you have made, so make sure you save your current configuration first if you want to keep it.

10.3 Open configuration file

To open an existing configuration file, select *File* → *Open* in the menu bar. A window will pop up where you can select the folder where the configuration file is saved to. Once you have selected the right folder, select the file and click the *Open* button.

10.3.1 Save configuration file

To save your configuration file, select *File* → *Save* in the menu bar. If your file hasn't been saved before, a window will pop up where you can select the folder you want to save your configuration file to and enter the file name you want to use. Once you have selected the right folder and entered the file name, click the *Save* button.

10.3.2 Save configuration file as

To save your configuration in a new file, select *File* → *Save As* in the menu bar. A window will pop up where you can select the folder you want to save your configuration file to and enter the file name you want to use. Once you have selected the right folder and entered the file name, click the *Save* button.

10.3.3 Export configuration to mas2g

To export your configuration to mas2g, you have to have saved your configuration first. Once you have done that, select *File* → *Export as MAS project* in the menu bar. A window will pop up where you can select the folder you want to export the configuration to and the file name you want to use. Once you have selected the right folder and entered the file name, click the *Export MAS project* button.

10.3.4 Exit the Scenario Editor

If you want to close the Scenario Editor, select *File* → *Exit* in the menu bar. Closing the Scenario Editor will not save any changes you have made, so make sure you save your configuration first if you want to keep it.

10.4 Entity Panel

The entity panel exists of 2 parts; the top part shows the bot options and the bottom part shows the e-partner options. A picture of the entity panel can be found in figure 9.

10.4.1 Create new bot

To create a new bot, click the *New Bot* button. A new bot will appear in the bot list.

If you want to create a standard bot (which already has default values), click on the arrow next to the *New Bot* button. A menu will drop down where you can select one of the standard bots. If you click on one of the standard bots, the bot will appear in the bot list.

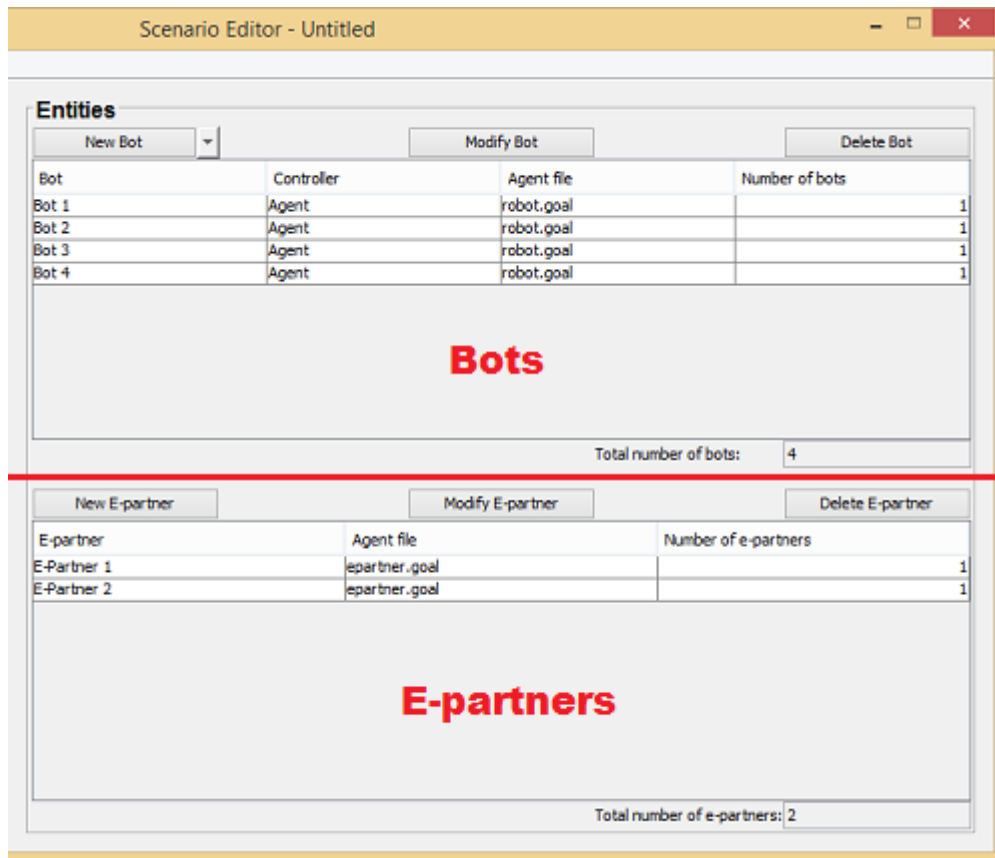


Figure 9: Entity Panel

10.4.2 Modify bot

To modify a bot, select the bot you want to modify in the bot list and click the *Modify Bot* button. The Bot Store will open in a new window where you can modify the properties of the bot.

10.4.3 Rename bot

To rename a bot, you can either use the *Modify Bot* button to rename in the Bot Store, or you can rename in the Scenario Editor. To rename in the Scenario Editor, select the bot you want to rename in the list with bots. Double click the its name and enter a new name.

10.4.4 Change controller type

To change how a bot is controlled, you can either use the *Modify Bot* button to change it in the Bot Store, or you can change it in the Scenario

Editor. To change the controller type in the Scenario Editor, select the bot you want to change and click on its controller type. You should now be able to choose the controller type.

10.4.5 Change the amount of entities of a bot

To change how many entities of a type of bot you want, you can either use the *Modify Bot* button to change it in the Bot Store, or you can change it in the Scenario Editor. To change the entity amount in the Scenario Editor, select the bot that you want to change the amount of. Now double click the amount given, and enter a new number.

10.4.6 Delete bot

To delete a bot, select the bot you want to delete in the list with bots and click the *Delete Bot* button.

10.4.7 Create new e-partner

To create a new e-partner, click the *New E-partner* button. A new e-partner will appear in the e-partner list.

10.4.8 Modify e-partner

To modify an e-partner, select the e-partner you want to modify in the e-partner list and click the *Modify E-partner* button. The E-partner Store will open in a new window where you can modify the properties of the e-partner.

10.4.9 Rename e-partner

To rename an e-partner, you can either use the *Modify E – partner* button to rename in the E-partner Store, or you can rename in the Scenario Editor. To rename in the Scenario Editor, select the e-partner you want to rename in the list with e-partners. Double click its name and enter a new name.

10.4.10 Change the amount of entities of an e-partner

To change how many entities of a type of e-partner you want, you can either use the *Modify E – partner* button to change it in the E-partner Store, or you can change it in the Scenario Editor. To change the amount in the Scenario Editor, select the e-partner that you want to change the amount of. Now double click the amount given, and enter a new number.

10.4.11 Delete e-partner

To delete an e-partner, select the e-partner you want to delete in the e-partner list and click the *Delete E-partner* button.

10.5 Bot Store

A picture of the Bot Store can be found in figure 10.

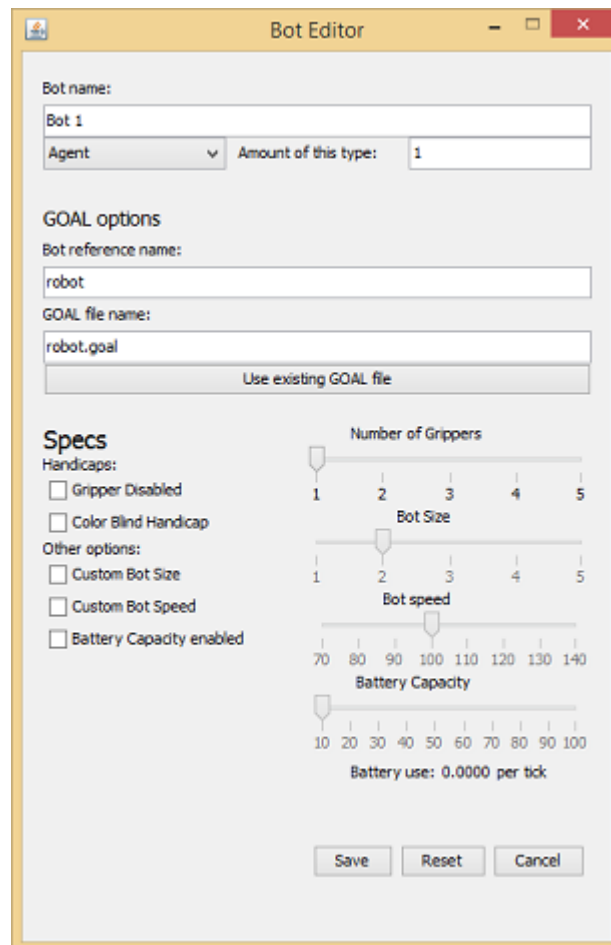


Figure 10: Bot Store

10.5.1 Rename bot

To rename a bot, click on the field that contains its current name. Now you are able to edit the current name or you can enter a new name.

10.5.2 Change controller type

To change how a bot is controller, you can click on its current controller type. Now you are able to choose the controller type.

10.5.3 Change the amount of entities

To change how many entities of the bot you want, you can click on the field that contains the current number of entities. Now you are able to edit the amount.

10.5.4 GOAL options

You can enter the GOAL options for your bot under the *GOAL options* section. Here you can indicate what its reference name in GOAL should be, and you can select a GOAL agent file which will control the bot. To select the GOAL agent file, you can click on the *Use existing GOAL file* button. A window will pop up where you can select the folder where the GOAL file is saved to. Once you have selected the right folder, select the file and click the *Open* button.

10.5.5 Bot properties

You can find the available bot properties under the *Properties* section. To change what properties you want the e-partner to have, you can select or deselect the checkbox next to the property description. Once you have enabled a property, you can use the sliders to change the value of that property.

10.5.6 Save modifications

If you want to save the changes you have made to the bot, you can click on the *Save* button. The Bot Store window will close, and your changes will have been saved.

10.5.7 Reset modifications

If you want to reset the changes you have made to the bot, you can click on the *Reset* button. Your changes will have been reverted to the last saved configuration.

10.5.8 Cancel modifications

If you want to cancel modifying the bot without saving any changes, you can click on the *Cancel* button. The Bot Store window will close.

10.6 E-partner Store

A picture of the E-partner Store can be found in figure 11.

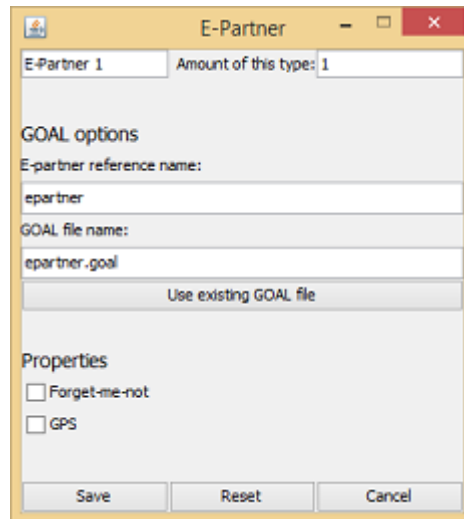


Figure 11: E-partner Store

10.6.1 Rename e-partner

To rename an e-partner, click on the field that contains its current name. Now you are able to edit the current name or you can enter a new name.

10.6.2 Change the amount of entities

To change how many entities of the e-partner you want, you can click on the field that contains the current number of entities. Now you are able to edit the amount.

10.6.3 GOAL options

You can enter the GOAL options for your e-partner under the *GOAL options* section. Here you can indicate what its reference name in GOAL should be, and you can select a GOAL agent file which will control the e-partner. To select the GOAL agent file, you can click on the *Use existing GOAL file* button. A window will pop up where you can select the folder where the GOAL file is saved to. Once you have selected the right folder, select the file and click the *Open* button.

10.6.4 E-partner properties

You can find the available e-partner properties under the *Properties* section. To change what properties you want the e-partner to have, you can select or deselect the checkbox next to the property description.

10.6.5 Save modifications

If you want to save the changes you have made to the e-partner, you can click on the *Save* button. The E-partner Store window will close, and your changes will have been saved.

10.6.6 Reset modifications

If you want to reset the changes you have made to the e-partner, you can click on the *Reset* button. Your changes will have been reverted to the last saved configuration.

10.6.7 Cancel modifications

If you want to cancel modifying the e-partner without saving any changes, you can click on the *Cancel* button. The E-partner Store window will close.

11 Log file

Repast logs the following for each run into a file. The filename is “BW4TXXXX.log” where XXX is the date and time to make the filename unique. The file is saved in the <SERVER> /log directory. It contains:

1. sequence: goal sequence (which block colors are to be dropped)
2. room: initial blocks per room
3. action: log of each action of a bot, with timestamp
4. total time: total time to complete task. Begin time is determined by first incoming action. End time is determined by the last block of the sequence dropped.
5. agentsummary: for each agent:
 - the bot type containing its handicaps
 - # correct drops in dropzone
 - # incorrect drops in dropzone
 - total time of standing still
 - # messages to other agents

- # rooms entered

Logs are written to the file as soon as possible, so that information won't get lost when the system is being killed before the end of the sequence is reached.