

Blocks World for Teams (BW4T)

PDF version

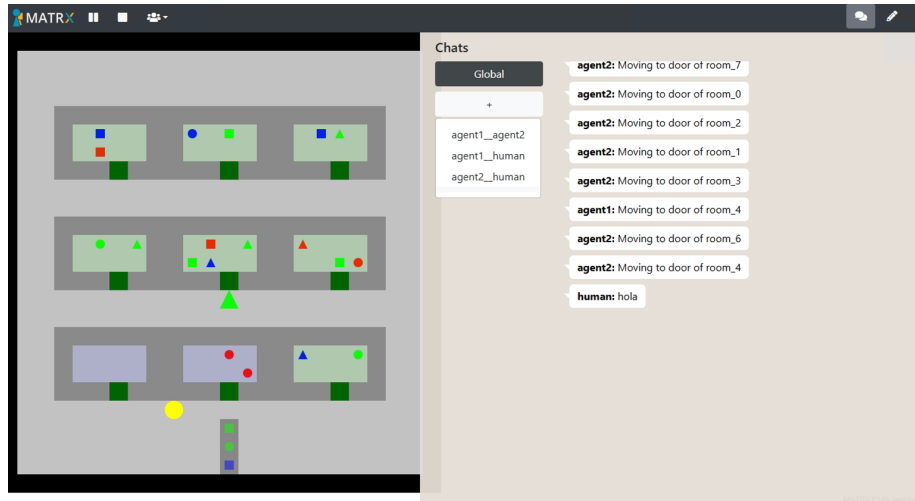


Figure 1: BW4T environment

Blocks World for Teams (**BW4T**) is a testbed EIS environment 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. (source).

The environments works in all Operating Systems as long as it allows to run Python.

Installation

1. Install python 3.10

For macOS, you can install Python 3.10 using Homebrew with the following command:

```
brew install python@3.10
```

For Windows, you can download the installer from the official website. For Ubuntu, you can install Python 3.10 using the following commands:

```
sudo apt update
sudo apt install python3.10
```

Test if the installation was successful by running the following command:

```
python --version
```

2. Download the BW4T environment from here or clone the repository with the following command:

```
git clone https://github.com/rsverhagen94/TU-Delft-Collaborative-AI-Trust
```

3. Create a virtual environment with the following command:

```
python -m venv bw4tenv
```

or you also specify the exact python version with the following command:

```
/opt/homebrew/bin/python3.10 -m venv bw4tenv
```

Activate the virtual environment with the following command:

```
source bw4tenv/bin/activate
```

For Windows, you can activate the virtual environment with the following command:

```
bw4tenv\Scripts\activate
```

to deactivate the virtual environment, run the following command:

```
deactivate
```

4. Install the required packages with the following command:

```
pip install -r requirements.txt
```

Make sure to install `matrx==2.1.2` and replace the `state.py` file (“TU-Delft-Collaborative-AI-Trust_x/venv_py310/lib/python3.10/site-packages/matrx/agents/agent_utils/state.py”) with the one provided in the repository (this file).

5. Run the BW4T environment with the following command:

```
python main.py
```

6. Open the browser and go to the following URL <http://localhost:3000/> to play the game. At the end of the game the logs are saved in `TU-Delft-Collaborative-AI-Trust_x/world_1` directory as timestamped csv files.

7. In-depth overview of the game can be found at <https://tracinsy.ewi.tudelft.nl/pubtrac/BW4T-Matrix-CollaborativeAI/wiki>. The game setup and algorithms can be changed by modifying the files

- `main.py`: set specific game settings
- `BW4TWorld.py` (TU-Delft-Collaborative-AI-Trust/bw4t/BW4TWorld.py): to change the appearance and the properties of the blocks
- `BW4THumanBrain.py` (TU-Delft-Collaborative-AI-Trust/bw4t/BW4THumanBrain.py): to change the human player's behavior
- `BW4TAgentBrain.py` (TU-Delft-Collaborative-AI-Trust/bw4t/BW4TAgentBrain.py): to change the agent's behavior

Task and Usage

The BW4T environment contains a number of objects of different shapes and colors in a number of rooms with doors. Each agent has limited visibility, but can navigate in the environment. The common goal of the agents in the world is to find blocks of desired colors and shapes, and deliver them to the “drop zone” to the right position in the right order. For example, in Figure 3, the goal is to deliver one blue rectangle object, one green circle object, and one green rectangle object, in that order. (content source).

After running the game (`python main.py`), the game can be played by opening the browser and going to the following URL `http://localhost:3000/`. The game can be ended by closing the browser window. The logs are saved in `TU-Delft-Collaborative-AI-Trust_x/world_1` directory as timestamped csv files. At the start the landing page will look like the one shown in the figure below (Figure 2).

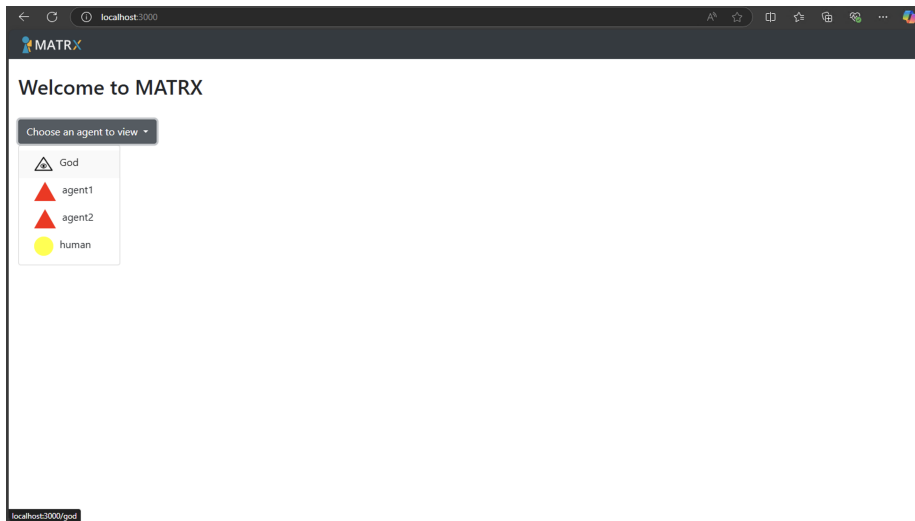


Figure 2: Start page of BW4T after launch

The game can be started by clicking the triangle play button on the top bar. With the god view the game looks like the one shown in the figure below (Figure 3), it shows the complete environment including all the agents and objects.

The human view shows the view of the agent in the environment, it looks like the one shown in the figure below (Figure 4). Human can control character can be controlled using

Keyboard Letter	Effect
W, A, S, D	Move agent

Keyboard Letter	Effect
Q	Grab an object
E	Drop object
R	Open a door
F	Close a door

Human view only allows to see the agent's view and not the complete environment.
Human can only see the objects in the room by ONLY entering the room.



Figure 3: God view

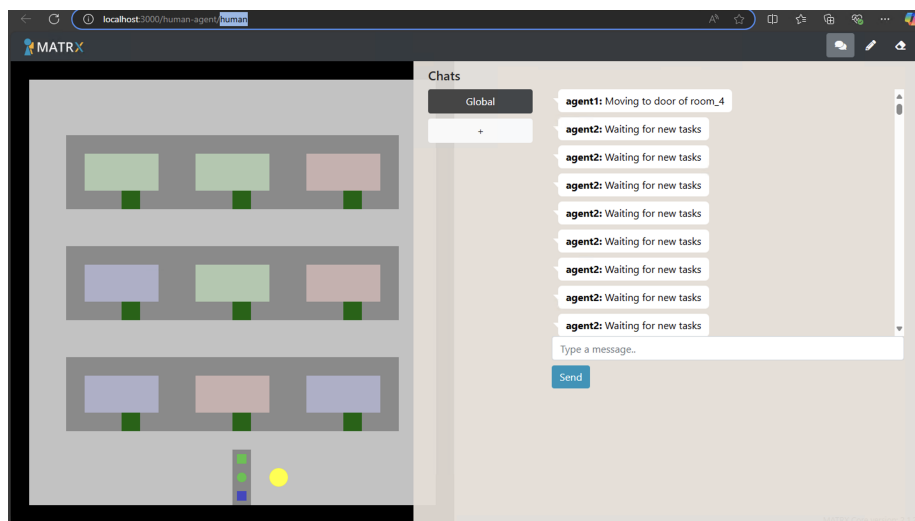


Figure 4: Human view