

# Rules of the Last Touch game

2013

## Overview

The game is played in an environment where there is a hexagonal grid of  $n \times n$  cells, each potentially containing objects. The boundary cells contain wall, some cells contain obstacle. The game is played by multiple agents of different types, which represent the persons playing the Touch-last game. The initial placement of agents and obstacles is chosen randomly. The first agent starts the game as the last touched.

## Permissible actions of an agent

**Moves** are processed provided that the target cell is empty:

- **north** – move one cell to the north in the direction which the agent is facing.
- **northwest** – move one cell to the north-west of the direction which the agent is facing.
- **northeast** – move one cell to the north-east of the direction which the agent is facing.

**Possible complementary move actions**

- All move actions can be specified as  $n$ -times repeated in the form (action  $n$ ), where  $n$  is a positive integer. This action corresponds to the basic action being repeated exactly  $n$ -times at one step. It will be stopped when an obstacle is reached.

**Turns** are actions that change the direction which the agent is facing:

- **turnleft** – turn by  $60^\circ$  to the left.
- **turnright** – turn by  $60^\circ$  to the right.

**Touch**

- **pass** means touching the other agent and passing it the ‘baba’. Pass is possible only if the touching agent is next to another agent and is facing in its direction.

**Stopping the game**

- **stop** – if there is an unreachable agent.

## Order of agent actions

The first agent to perform its action is always that having the last touch. Actions of the remaining agents are performed in the same relative order in which they were performed last time. Table 1 illustrates this behaviour using four agents (the suffix  $b$  denotes the last touched agent).

First	Second	Third	Fourth	Note
$1b$	2	3	4	assume 1 touched 2
$2b$	1	3	4	assume 2 touched 4
$4b$	2	1	3	assume 4 touched 1
$1b$	4	2	3	etc.

Table 1: Example of changing order of the agents action execution.

## Structure of agent's percept

The agent's percept is a variable-length list, whose first element is the agent-body. The second element is either **bump** or a list containing one element, which is the contents of the cell one step to the north of the direction, in which the agent is facing. The  $(n + 1)$ -th element is a list containing  $n$  elements, which correspond to the positions visible from the agent inside the angle  $120^\circ$  in the direction where the agent is facing. Content of the cell is represented as follows:

- **nil** – the cell is empty.
- **dark** – the cell cannot be seen by the agent, because it is blocked out by an obstacle (or wall).
- **(baba x y)** – the cell is occupied by the last touched agent facing in direction **(x y)**.
- **(agent x y)** – the cell is occupied by an agent facing in the direction **(x y)**.
- **wall** – agent is looking at a wall.
- **obstacle** – agent is looking at an obstacle.

Note that direction **(x y)** is arranged. You can't add it directly to the location by e.g. function **xy-add** to get the . Instead, use the prepared function **xy-update**.

Visibility of cells in percept is determined by the Breadth-first search algorithm run from the bottom and left to right. Cell added to the algorithm queue by a visible cell will also be visible, while the cell enqueued by a **dark** cell will be **dark**. Try the game to see how it works.

## Supplementary information

The structure representing an agent's body includes e.g. the following fields:

- **(object-loc agent-body)** – actual location of the agent.

- `(tl-agent6-body-grid agent-body)` – grid where the agent can construct its own representation of the state of world.
- `(tl-agent6-body-ma-babu? agent-body)` – T if the agent has been touched last.

## How to run the game

1. The directory in the second line of the file `run.lisp` must be changed to match the project root directory, e.g.

```
(cd "~/agentsim")
```

2. Agent test runs can then be performed using

```
(run-gui (make-tl-world6))
```

## ask-user-agent key bindings

You can use keys *Q*, *W* and *E* to move north-west, north and north-east. Keys *A* and *D* turn the agent to the left and to the right. Pressing *P* causes the agent to **pass**, *X* stops the game.

## List of useful functions

Following functions can be found in file `hex/utilities.lisp` and should be examined before usage:

- `update-grid`
- `xy-update`
- `tnorth`
- `tnorthwest`

- tnortheast
- tsouth
- tsouthwest
- tsoutheast