Rules of the Dodge ball game

About the game

The game is played in an environment with a rectangular grid of ***n*** x ***n*** spaces. The boundary spaces contain a **`wall.`** Your **`agent`** will be placed on a random empty space. Other spaces may contain other agents, but two agents cannot be on the same space. An agent may move freely, but he must not violate the 'one space - one agent' condition by his movement. A **`ball`** is placed on a random space that does not contain a wall. The agent can **`grab`** the ball to hold it. If the agent holds the ball, he can **`throw`** it on a space that he specifies. This space must not contain a wall. If any of the spaces between the agent and the target space are occupied by an agent, this agent will be hit by the ball with a probability **`p`**. With probability **`1-p,`** he will dodge the ball. Every agent has 3 lives initially. If an agent is hit by the ball, the agent loses one life. If an agent has 0 lives, he is removed from the environment and does not play anymore.

Throwing the ball

A space lies between the thrower's space and the target space if it is intersected by a line that connects the centres of the thrower's space and the target space. Distance of two spaces is the length of this line. When an agent throws the ball, the ball must pass through a series of spaces. If an agent is standing on a space and the ball passes through this space, he will be hit by the ball with a probability ***p***. In case of the mandatory game mode, this probability is always ***100%***.

In case of the competitive game mode (students against each other), the probability ***p*** of hitting an agent with a ball decreases with the distance ***d*** of the affected agent from the agent that throws the ball (the length of one space is 1). If the distance ***d*** is less than 2, the thrower will hit the other agent with probability ***p = 100%***. If the distance ***d*** is greater or equal to 2 but less than 5, the other agent will be hit with probability ***p = (100 - d*10) %***. If the distance is 5 and more, the other agent is not hit even if the ball passes through or lands on his space.

If an agent is hit by the ball, the ball drops on the space where the agent is standing. Therefore if more than one agent stand between the thrower's space and the target space, the closer agent is the first to dodge the ball. Only in case he dodges the ball will the agent standing behind him have to dodge the ball.

The agent cannot hit himself with the ball.

Movement in the environment

The agents may move in the environment: they can go **`left, right, up`** and **`down`**. The agents may not move in the environment if they possess the ball.

Time in the game

The game is a turn-based simulation. The time is divided into steps. In each step, an agent may perform one action: move, grab the ball or throw the ball. Actions of all agents are performed simultaneously, with the exception of the throw-ball action, which is always executed first.

Bumping agents

All agents declare where they want to go (as their action) before their movement is executed. If two agents occupy the same space after the movement, neither of them will move to prevent bumping into the other. If a ball lies between two agents and both agents try to go to this space at the same time, the ball is „kicked out" onto a random place in the environment.

Deadlock prevention

An agent that is on the same space where the ball is must make sure that at most after three turns he will stop sharing the space with the ball. That means that he will either throw the ball or move out of the space. If he does not move out or throw the ball, he loses one live for each time step over the limit.

Game modes

There are two game modes – in the first the agent is playing against computer opponents. These do not move in the environment. If, however, a ball drops onto their square, they throw it at the place where the student's agent is standing.
In the second game mode the student's agent is put into an environment where he has to compete with the agents of all the other students. The winner (the student whose agent scores the most points) gets the A grade without having to go to the exam.

Bugs in the agent programs

If there is a bug in your agent program that prevents the tests from running successfully, you will lose points. If the bug is present in the program for the game mode 1 and the game mode therefore cannot be run successfully, you will not be awarded the assessment. If the bug is present in the game mode 2, your agent shall be removed from the competition for the A grade.

Goal of the game

In case of the single-player game where the player plays against the computer adversaries, the player wins only if he is the only agent alive at the end of the game.
In game mode 2, every agent scores points for hitting other agents (10 points per hit) and for staying alive while the others die (20 points per agent outlived). The winner is the agent with most points at the end of the game. Agents are allowed to cooperate.

End of the game

The game ends either after ***100*(number of agents in the game)*** turns or when only one agent remains in the environment.

Percept of the agents

Every agent sees the whole environment: he sees the playing grid; he sees the walls, positions and names of the other agents and their number of lives. He also sees the ball and knows if the ball is in someone's possession.

Permissible actions of an agent

Moves – are accomplished provided that the goal space is empty:

- **`go-up`** – move one position forward
- **`go-left`** – move one position to the left
- **`go-right`** - move one position to the right
- **`go-down`** - move one position to the right

Manipulating the ball – these are the actions that affect the ball

- **`grab-ball`** – grab the ball if the agents stands on the space where the ball is
- **`(throw-ball x y)`** – throw the ball on the specified square. The agent has to hold the ball to be able to throw it

Structure of agent's percept

The agent's percept is a two-element list, whose first element is the **`agent-body`**. The second element is the perceived game room in a form of a two-dimensional array

representing the contents of the environment. Each space can contain ***nil*** or a list of ***percept-objects***:

- `'agent`
- `'wall`
- `'ball`

Files necessary for a proper work of an agent

1. The modified AIMA code is available on the course web pages. You can either download the whole modified AIMA distribution from the course website if you are using Allegro CL 8.2, or download the AIMA code from its official website and update it with the files from DBFiles.zip.
2. The following files from the AIMA code should be loaded without any changes, only their home directory has to be specified accordingly:

```
(load "e:\\JUI\\AIMA\\utilities\\utilities.lisp")
(load "e:\\JUI\\AIMA\\utilities\\binary-tree.lisp")
(load "e:\\JUI\\AIMA\\utilities\\queue.lisp")
(load "e:\\JUI\\AIMA\\utilities\\cltl2.lisp")
(load "e:\\JUI\\AIMA\\agents\\environments\\basic-env.lisp")
(load "e:\\JUI\\AIMA\\agents\\environments\\grid-env.lisp")
(load "e:\\JUI\\AIMA\\agents\\agents\\agent.lisp")
(load "e:\\JUI\\AIMA\\agents\\algorithms\\grid.lisp")
```

Optionally, you can load the whole AIMA environment by:

1. Editing `aima.lisp` according to instructions at http://aima.eecs.berkeley.edu/lisp/doc/install.html
2.

```
(load "e:\\JUI\\AIMA\\aima.lisp")
(aima-load 'all)
(aima-compile) ;; needs to be run once only if (test all) returns t
(test 'all)    ;; needs to be run once only if returns t
```

3. The following file has to be completed and then loaded

```
(load "e:\\JUI\\AIMA\\agents\\environments\\dodgeball.lisp")
```

Hint: you can start implementing your agent by copying the code for agent `'ask-user-db-agent`. You have to rewrite the program function so that it stops asking for help and starts acting on its own.

4. Test runs can be performed using e.g.

```
(test-agent-mode-1 'your-agent-name)
```

AIMA lisp download and documentation:

http://aima.cs.berkeley.edu/lisp/doc/overview.html

Files to submit

Submit two files – one containing your implementation of the agent for the first game mode, the other an implementation of the agent for the second game mode. Do not re-submit AIMA code.