AgentWorld
COEN 166/266

Overview
AgentWorld is a simulated natural environment, populated with competing agents, vegetation, and predators. Agents can consume vegetation (when it is blooming) to increase their energy level. An agent's energy level will also decrease over time based on their activity.

Invocation
All of the sources and executables for AgentWorld are available in a shared directory on the DC (design center) linux machines: `/scratch/COEN266/AgentWorld`. To start using AgentWorld, copy these files into a local directory in your own account. There is no need to rebuild the software from source unless you want to install it on a system outside of the DC.

When the "`simworld`" executable is run, it looks for student agents in subdirectories of the `Agents` directory. Each of the environments in the `Environments` directory will be run in succession, populated with all of the student agents. When a simulation is complete, a history of the simulation is added to the `Results` directory.

Environment
The environment is comprised of an grid whose size is unknown to the agents. Each square in the environment contains no more than one of the following items at any time:
* Student Agent: Yourself, or another student
* Vegetation: The resources your agent will eat to increase its energy level
* Predator Agent: An environmentally-controlled agent that wanders the terrain looking for Student Agents to attack. Each turn a predator may move up to one space, and if it is in a space adjacent to a student agent it may choose to attack that agent.
* Boundaries: The edge of the landscape.

The following attributes are specified in the environment file:
* Size of the world (X and Y dimensions)
* Duration of the simulation, in turns
* Starting energy level of the agents
* For each type of vegetation:
  - The likelihood of it occurring in a given square
  - The incubation period (number of turns after it is eaten before it will bloom again)
  - The bloom patten (how much energy it will be worth after each turn in which it is in bloom)
* For each type of predator:
  - How much damage it can inflict on an agent (in terms of energy lost)
  - How many turns the predator will survive
  - How many instances of the predator will be in the environment at any given time

Actuators
Your agent will be able to perform one of the following actions:

STAY: Your agent's position and orientation will not change. It will, however, lose 1 point of energy.

TURN-{RIGHT,LEFT,AROUND}: Your agent's orientation will change to the (relative) left, right, or opposite of its current orientation. It will lose 2 points of energy.

MOVE-{PASSIVE,AGGRESSIVE}-{1,2,3}: Your agent will be charged 10, 30, or 60 energy points to move 1, 2, or 3 squares in a turn. When more than one agent attempts to move into the same square, the following occurs:
PASSIVE: If all agents moved passively, they will all yield and nobody will enter the square
AGGRESSIVE: If one or more agents moved aggressively, all agents that moved passively will yield and only be charged the cost of the distance actually moved. If more than one agent moved aggressively, only the strongest agent will enter and all aggressive agents will lose an additional 30 points for each opposing agent that attempted to move aggressively into the same location (e.g., if 4 agents attempt to move aggressively into the same square they will each lose 90 points). In the case of a tie for strongest, nobody will move.

EAT-{PASSIVE,AGGRESSIVE}: Your agent will be charged 5 points of energy. If there is vegetation in the single square immediately in front of the agent, and no other agent is also trying to eat from the same vegetation, then all of the energy from that vegetation will be added to your agent's score. If more than one agent is trying to eat from the vegetation at the same time, then the outcome will depend on whether the agents were eating passively or aggressively:
PASSIVE: If none of the agents are aggressive, the total energy from the vegetation is divided equally amongst them.
AGGRESSIVE: If one or more agents are aggressively eating, all passive agents will allow them to eat and will be charged 5 points. The aggressive agents will be charged 30 points each for each opposing agent that was also eating aggressively, and only the strongest will get the entire energy from the vegetation.

Sensors

Your agent will be able to "see" at a 45 degree angle to each side in front of themselves, up to 5 squares in the distance. Your agent will know how much energy it has remaining, and the energy value of vegetation within its sight.

Interface

Your program must be loadable from a main.scm file. If you have other files, they should be loaded from main.scm.

On entering a new environment, the function "`initialize-agent`" will be called with no parameters. The agent will then have 5 seconds in which to initialize its state to prepare to enter the new environment. Your agent must return a string (although the actual value of the string will be ignored by the environment).

Every subsequent turn will then cause a "`choose-action`" function call to be issued to your agent. This call will have three parameters: an integer value representing the energy level of your agent, a list of the events that occurred during the last turn, and a list representing the discernible environment. The environment list will contain five sublists of lengths 3, 5, 7, 9, and 11, representing the contents of spaces up to 5 positions away from the agent in the direction it is facing.

The list of events from the last turn may include the following values:
`(moved` ***`spaces`*****`)`**: Your agent moved 'spaces' positions
`(ate` ***`id e-delta`*****`)`**: Your agent got 'e-delta' energy from eating vegetation 'id'.
`(fought` ***`id e-delta`*****`)`**: Your agent had a change in energy of 'e-delta' from fighting with agent 'id'.
`(attacked-by` ***`id e-delta`*****`)`**: Your agent had a change in energy of 'e-delta' from attacks by predator 'id'.

The environment percepts will be one of the following values:
`empty`: represents a vacant square

barrier: Represents a location that cannot be occupied
(agent *id strength direction*): represents a competing agent, with a unique identifier, an integer value representing the log base 2 of its energy level, and a value representing the orientation of the agent, which will be one of the following values:

- away: the agent is facing the same direction as your agent (away from your agent)
- towards: the agent is facing in the opposite direction as your agent (towards your agent)
- right: the agent is looking to the right of your agent
- left: the agent is looking to the left of your agent

(predator *id*): represents a predatory agent, with a unique identifier
(vegetation *id bloom*): represents a plant, with a unique identifier and an integer value representing the amount of energy available from eating this plant

Your agent **must return a string** representing one of the actions listed in the 'Actuators' section. It is expected that your agent will be able to make a decision within 1 second of the function call. Agents that crash or take significantly longer the 1 second to make their decision may be considered dead and eliminated from the environment. Agents that return an invalid string will be considered to STAY.

If you are finding the interface confusing, take a look at the sample agent provided in the Agents/Random directory. Then, run the simulation with that agent and look at the log file. It will show you the current state of the environment, the choose-action commands sent to the agent, and the string returned from the agent for each.