Adversary Task Simulation Model

Version 1.0 - 06/10/2018

Overview:

This simulation can be used to model a swarm of agents, some informed (leaders), and the others, followers who simply obey reynolds rules and their programed communication model. In addition to the swarm of interest, there are adversary agents which are programed to attack the swarm. If an agent comes too close to an adversary agent, it is said to be eaten, and changes colors from white to grey. The intended objective of this simulation task is for all of the agents to reach the goal area without being eaten, where they are safe from the adversaries and are rewarded for reaching it.

When the simulation begins, each leader agent has information about the goal area, its distance and location relative to the goal area, and its distance and location relative to other non-informed agents. The controller for the leader agents is powered by a neural network which learns based on the error function's output after each round. Learning (training) of the leaders of the swarm only occurs when the "TRAIN" parameter is specified. When the "TEST" parameter is specified, the swarm will not learn, but simply execute the behavior it has either learned or started out with (depending on how the model is being used).

Please Note:

- After each change in the code for the model, ./compile must be executed at the command line for these changes to be reflected in the next simulation execution.
- In the includes.h header file, visualization can be set to 0 or 1, enabling or disabling visual rendering of the simulation. For training, it is recommended to run the simulation with visualization set to 0, as the model runs much faster this way.
- A space must follow each command line argument, except the final one

Command line argument parameters in order from first to last:

1. Location of the executable model: **./bin/main** (while in the home directory of the project)
2. Operation to be performed by the model: **"TRAIN"** or **"TEST"** (including quotes)
3. Number of leader agents: (number of leader agents must be less than total agents)
4. Number of total agents
5. Number of epochs for which the simulation will iterate over
6. A placeholder 1 from previous use and development

Command line example to train the model with 4 leaders, 10 agents, and 100 epochs:

```
./bin/main “TRAIN” 4 10 100 1
```

Command line example to test the model with 4 leaders, 10 agents, and 100 epochs:

```
./bin/main “TEST” 4 10 100 1
```