**Assignment 4 Q&A**

**Question: Why Turtles are not moving?**

Answer: If none of turtles are moving when you run your program, try to move your turtle1 then they will start moving.

**Question: Does each of T turtle is supposed to move in a square?**

Answer: Each of T turtles are not supposed to move in a square. Each T turtle is supposed to move differently showing different behavior.

For example,

* T1 will move back and forth in a fixed boundary. For this turtle you need to focus on "distance" factor (attribute) to identify its behavior pattern. For this turtle, Rote learning to remember the fixed boundary can be an approach you can use.
* T2 will move back and forth as well but not necessarily in a fixed boundary. However, if you observe the "distance" factor, you will notice that T2 moves based on certain pattern. For T2, think about using generalization or induction.
* T3 will move by two factors "distance" and "rotation". So your program needs to find out the move pattern by two attributes "distance" and "rotation angle" using a learning method like decision tree or simple probability calculation (see the rock-paper-scissor example in lecture).
* Remember that the main objective of this assignment is to use any learning method (discussed in class such as Rote learning, generalization or induction, and supervised learning such as decision tree or perceptron learning) to learn the T turtle move patterns and capture all T turtles.
* You don't have to worry about finding out X turtles' move pattern. **For X turtles, just try to avoid being captured by X turtles**. Note that one of X turtles may chase after turtle1 when turtle1 stays on the same location for too long time.

There are many sophisticated machine algorithms developed by computer scientists in the past. Even ID3 decision tree induction algorithm is not simple. Those algorithms are all important and useful in solving complex problems. However, for assignment#4, you don't have to implement any complicated machine learning algorithm to accomplish the mission. To think about a machine learning method you need to use, just remember the principle idea of machine learning or the following definition of machine learning:

"Any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population”.

In fact this is the basic learning method we all use.

**Q1: We've noticed since working on the code that the X turtles are much closer to the spawn position of turtle1 than they were in the last assignment. This is because your code for spawning the turtles goes from (0 + 2, 0 + 2) to (9 + 2, 9+2) and turtle1 now spawns in the middle of the map, instead of at (0, 0). It does still appear that none of the X turtles will spawn on turtle1 (as far as we have seen so far anyway), just that they might only be a distance of 1 away. Is this intended?**

A1: No I didn't intended but if spawning X turtles being too close to turtle1 is going to be a problem, I can remove the restrictions for Turtles locations.

**Q2: We noticed the three patterns: moving back and forth, moving up and down, and moving in a straight line (back and forth), rotating 90 degrees, moving in a straight line (back and forth), then rotating 90 degrees once again. Will we be expected to learn any other patterns?**

A2: Yes you should definitely be able to find some patterns so that you can use Rote learning, generalization (including basic mathematical induction), or any supervised learning. Particularly, you may use the Rote learning for the first pattern, generalization or induction for the second pattern, and supervised learning based on two factors (or attributes) for the third pattern. However, these patterns may change a little bit, which means that you don't want to hard code the patterns. Instead make your turtle learn the patterns itself so that it can adapt to such slight changes in turtles' move. This sounds hard and challenging but you will find it that it is easier than you think once you understand the basic concepts and methods of machine learning.

**You use the following ways to build and run hw4test.cpp:**

**(1) using one package for both hw4test.cpp and your program**

(a) Copy "hw4test.cpp" to your package directory that contains your program.

(b) Add two lines "add\_executable(hw4test hw4test.cpp)" and "target\_link\_libraries(hw4test ${catkin\_LIBRARIES})" to CMakeLists.txt file.

(c) go to your workspace directory and do "catkin\_make" to build both programs.

(d) open a terminal, move to your workspace directory

(e) type "source devel/setup.bash"

(f) type "rosrun yourpackagename hw4test"

(g) open another terminal and "rosrun yourpackagename yourprogramname".

Of course, before you do steps (f) and (g) you have to run roscore and turtlesim window by "rosrun turtlesim turtlesim\_node".

**(2) using a separate package for hw4test.cpp**

(a) create a package directory for hw4test

(b) copy hw4test.cpp to the package directory

(c) change the CMakelists.txt file appropriately as specified in (1)

(d) go to your workspace directory and do "catkin\_make"

(e) the rest steps are identifical to steps (d) - (g) specified in (1).

***I think using one package for both your program and hw4test.cpp will be easier***. If you put your program under yourpackagedirectory/src, you just need to change the program name in CMakeLists.txt file.