**Assignment 4 - Machine Learning w/ Turtlesim**

**Due: 5/21/17 By: 11:55pm**

**a) Team Name: Cyberdyne Industries**

Mohammad Hassnain, mohammed-17@csu.fullerton.edu (25% contribution)

Stephen Chan, stephen.chan9@yahoo.com (25% contribution)

Reyniel Maglian, renrenmaglian@gmail.com (25% contribution)

Vincent Campbell, vrcampbell@csu.fullerton.edu (25% contribution)

**b) Description of the machine learning used to accomplish our goal:**

Since each turtle has a randomized movement pattern we decided to capture the target turtles in no particular order. We used a do-while loop in order to find the target turtles while they existed. We took into account the danger\_tolerance variable used in the beginning of the program to ensure that the turtles were separated from the villain turtles. Our turtle will know the distance between each target turtle whenever moveGoal is called.We implemented maneuvers to ensure that the turtle stops when a villain turtle is close to the main turtle by using a danger tolerance variable to alert the main turtle when it is .5 away from a villain turtle. Our algorithm will always check for the villain turtles at each of the locations and the algorithm will stop when all target turtles are captured.

**c) Pseudocode (not source code):**

**main()**

*Tturtles = max number of target turtles*

*Do{*

*//should move our turtle1 the distance to the first target*

*Call moveGoal()*

*if(turtle1 position = target turtle position){*

*Kill the target turtle*

*}*

*//if the target has moved*

*Else*

*{*

*while(turtle1 position != target turtle position){*

*Call moveGoal()*

*}*

*}*

*Tturles, aka the current number of target turtles, - 1*

*}while(tturtles != 0)*

**movegoal()**

*// To check if Turtle1 is a safe distance from villian turtles*

*If (( getDistance(main\_turtle, villain\_turtle1)>(D\_TOL)) &&*

*(getDistance(main\_turtle, villain\_turtle2) >(D\_TOL)) &&*

*( getDistance(main\_turtle, villain\_turtle3)>(D\_TOL))*

*)*

*{*

*moveGoal(main\_turtle,target\_turtle)*

*}*

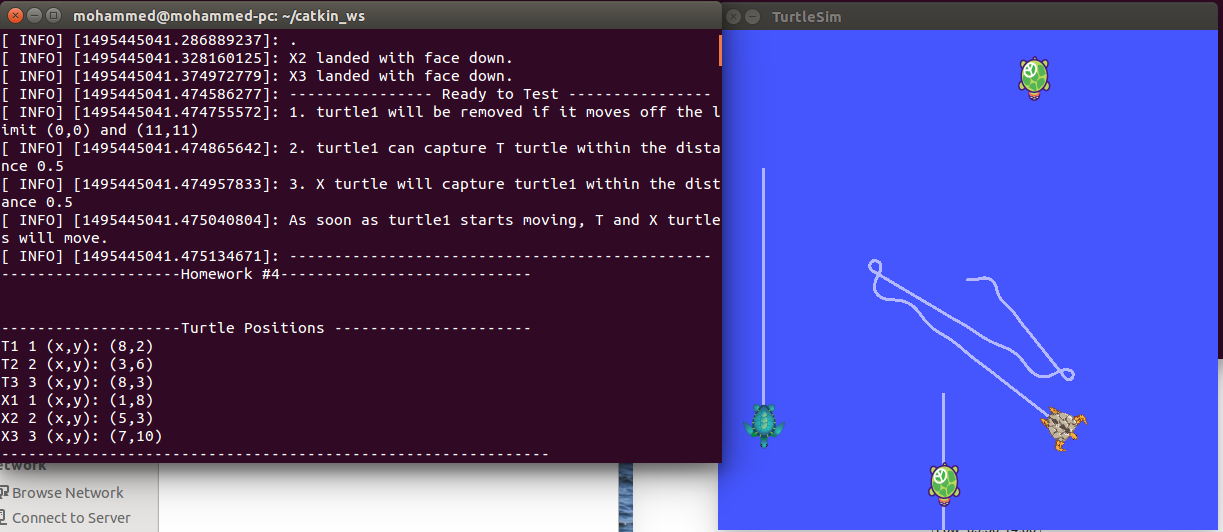
*Else{*

*Stop turtle*

*Rotate turtle*

*Move turtle short distance*

*}*



**d) Description of moving patterns learned from each T turtle:**

Although each turtle doesn’t have an assigned movement pattern, we were able to figure out the movement of the T turtles. There are going to be 4 turtles that move up and down in a line, 1 turtle that moves in straight lines after rotating at a couple stops, and 1 turtle that doesn’t move at all. The window of opportunity to catch the turtles we figured is when the turtles are stopped in their patterns. During the turtlesim run, the turtles in random eventually stop in their tracks for a certain amount of time and then resume their patterns. We figured it was best to catch the turtles when they aren’t moving or else would just be in a constant chase with the turtles.

**Resources Used:**

-Edu.gaitech.hk

<https://github.com/aniskoubaa/gaitech_edu/blob/master/src/turtlesim/cleaning_app/robot_cleaner.cpp> (code from the video)

**YouTube resources:**

ROS Tutorial 4.2: Moving in a Straight Line (Turtlesim Cleaner)

Covers robot\_cleaner.cpp setup and execution

<https://www.youtube.com/watch?v=PGZMlzBlMmw>

ROS Tutorial 4.3: Rotation Left/Right (Turtlesim Cleaner)

<https://www.youtube.com/watch?v=Ddqwq2WXFEk>

ROS Tutorial 4.4: Go-To-Goal Location (Turtlesim Cleaner)

<https://www.youtube.com/watch?v=Qh15Nol5htM>