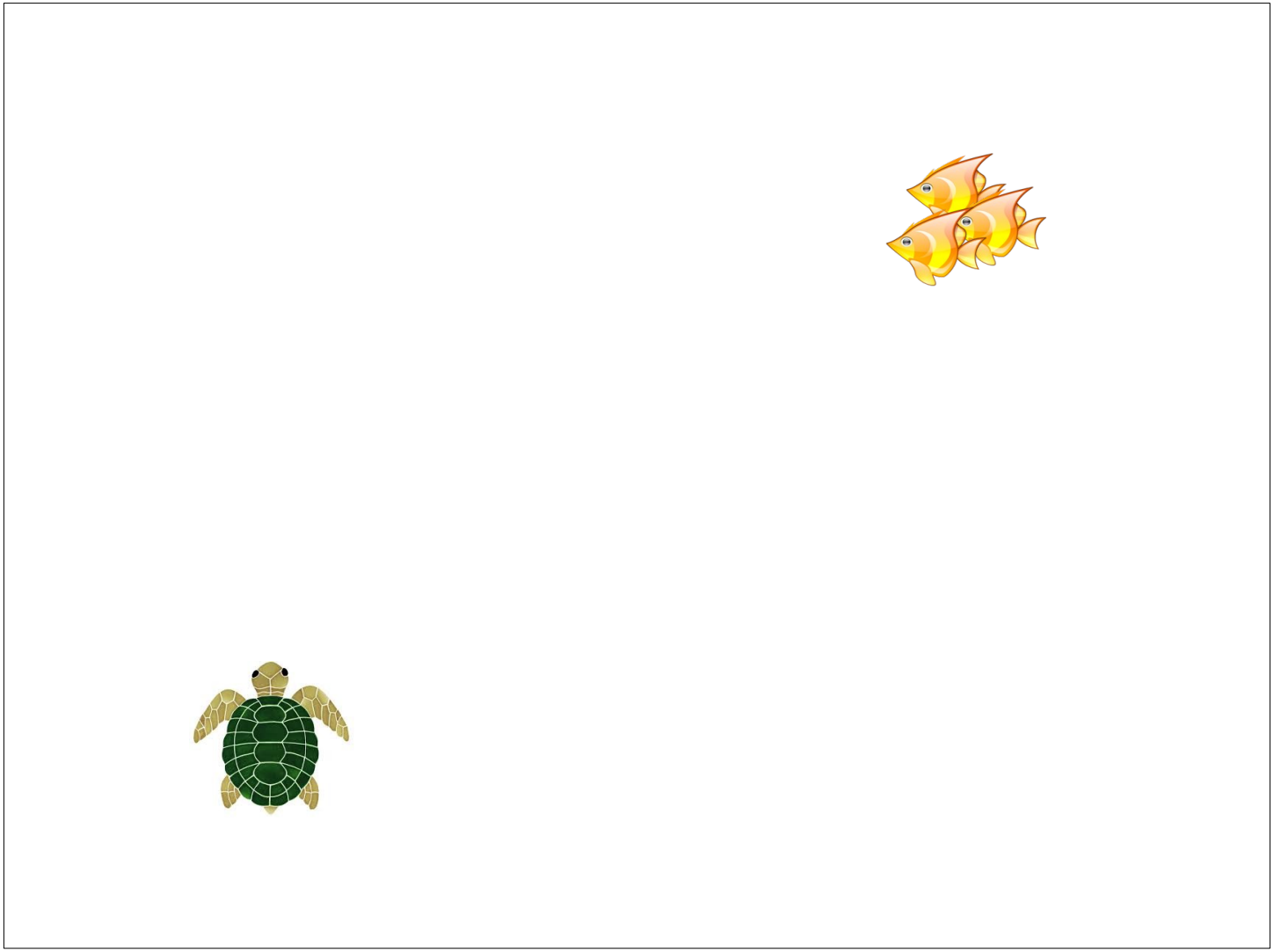
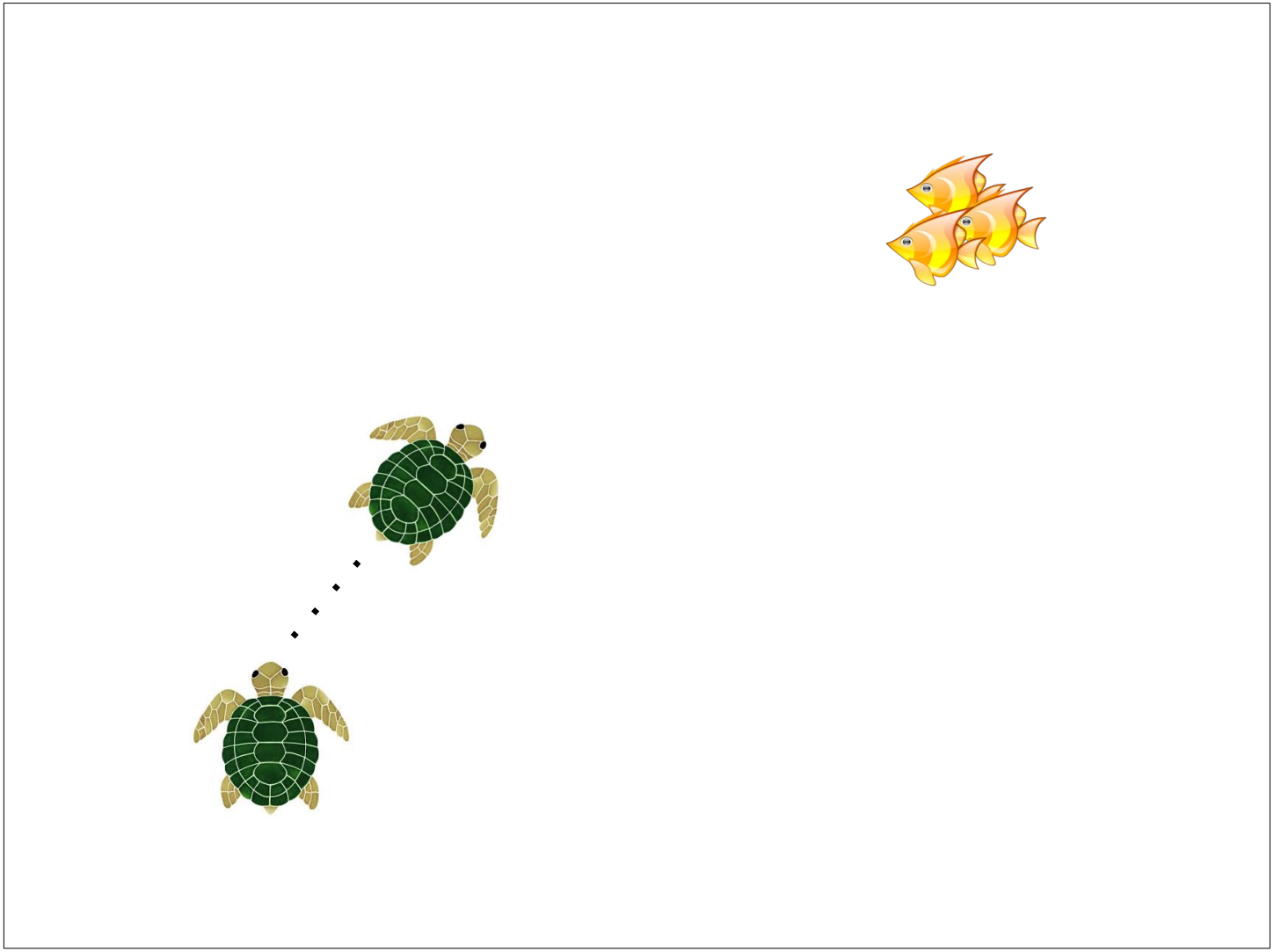
Homework 4: Multi-Agent System

Reactive Paradigm Example

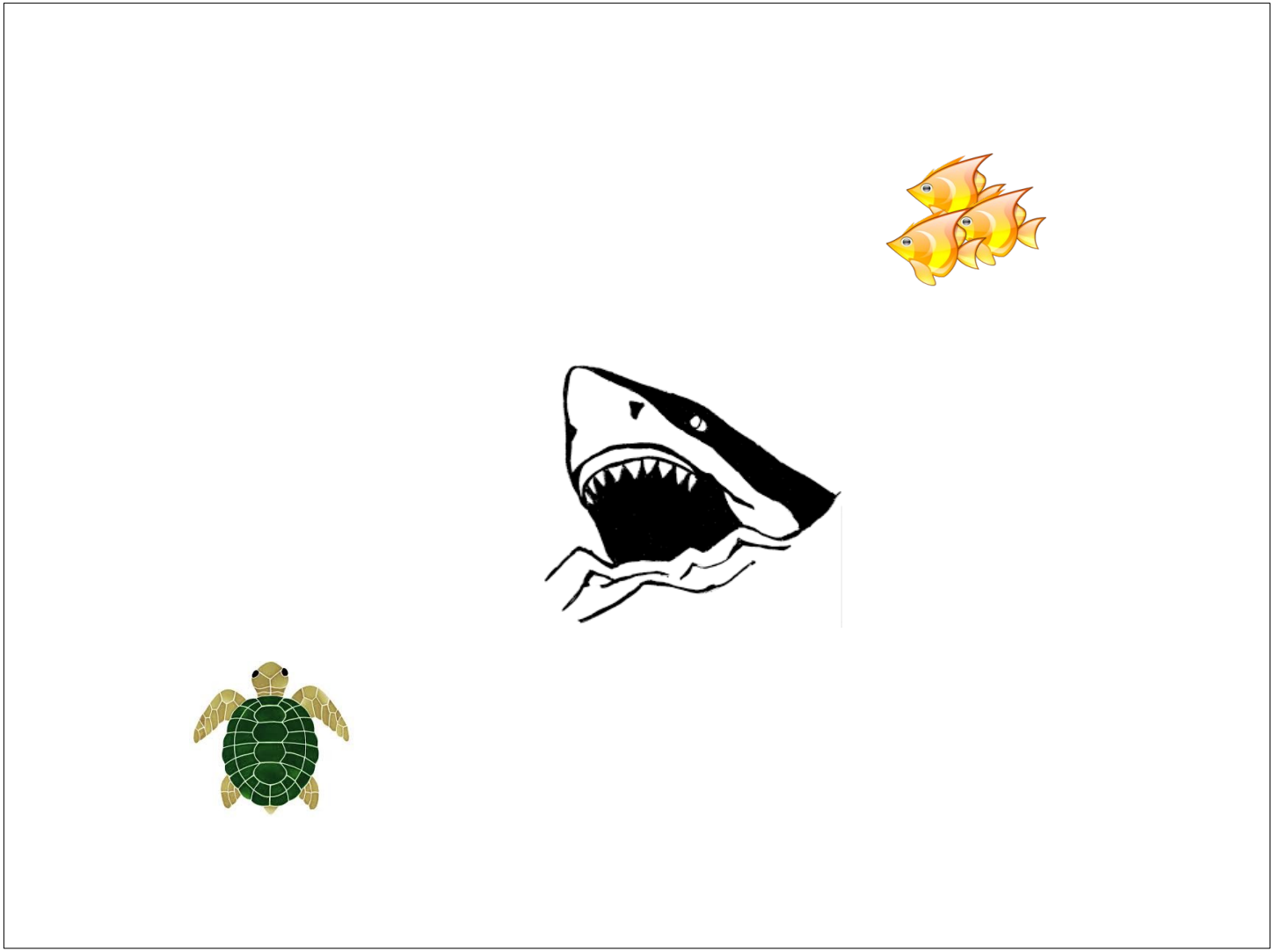
Reactive Paradigm Example



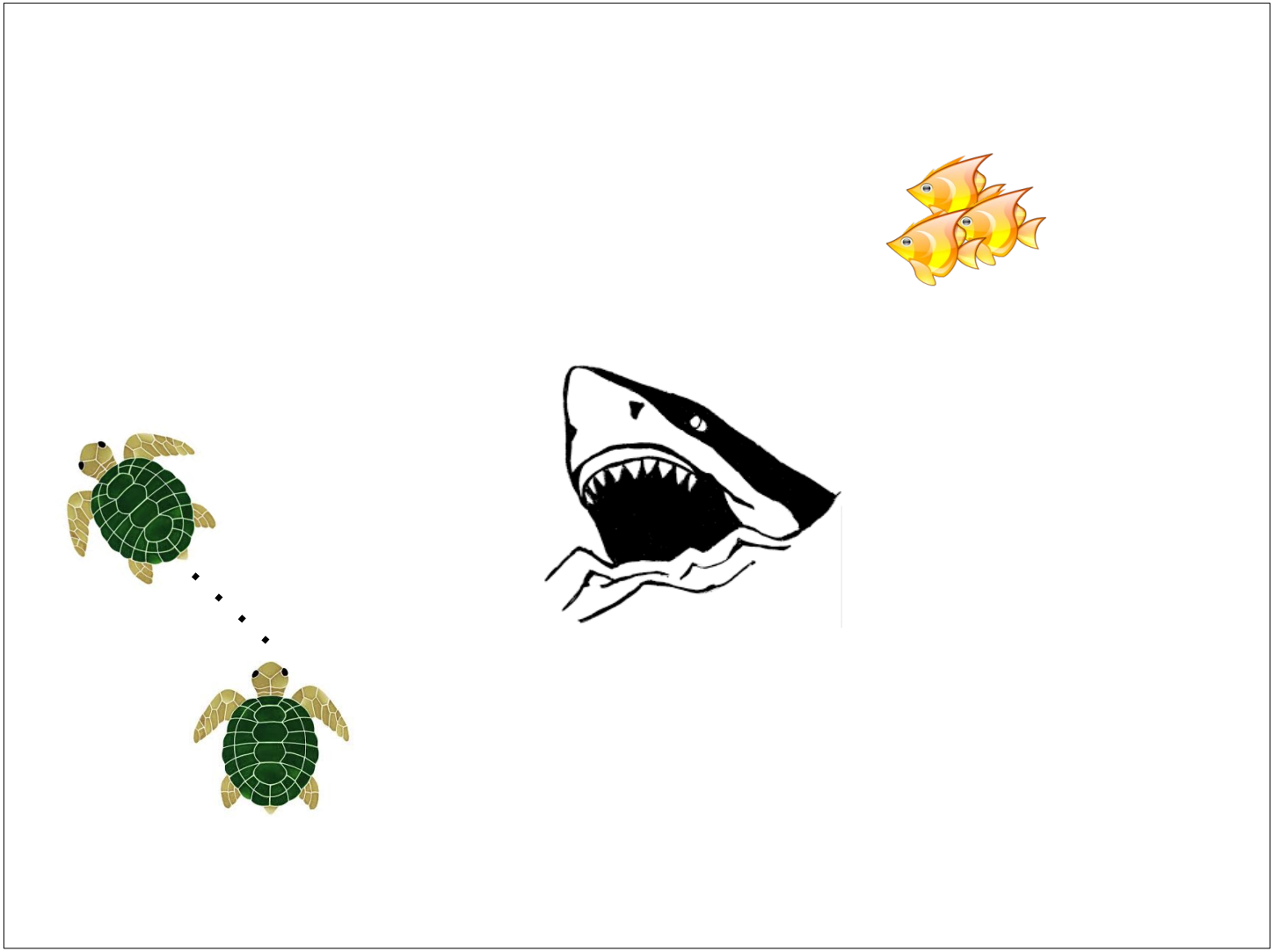
Reactive Paradigm Example

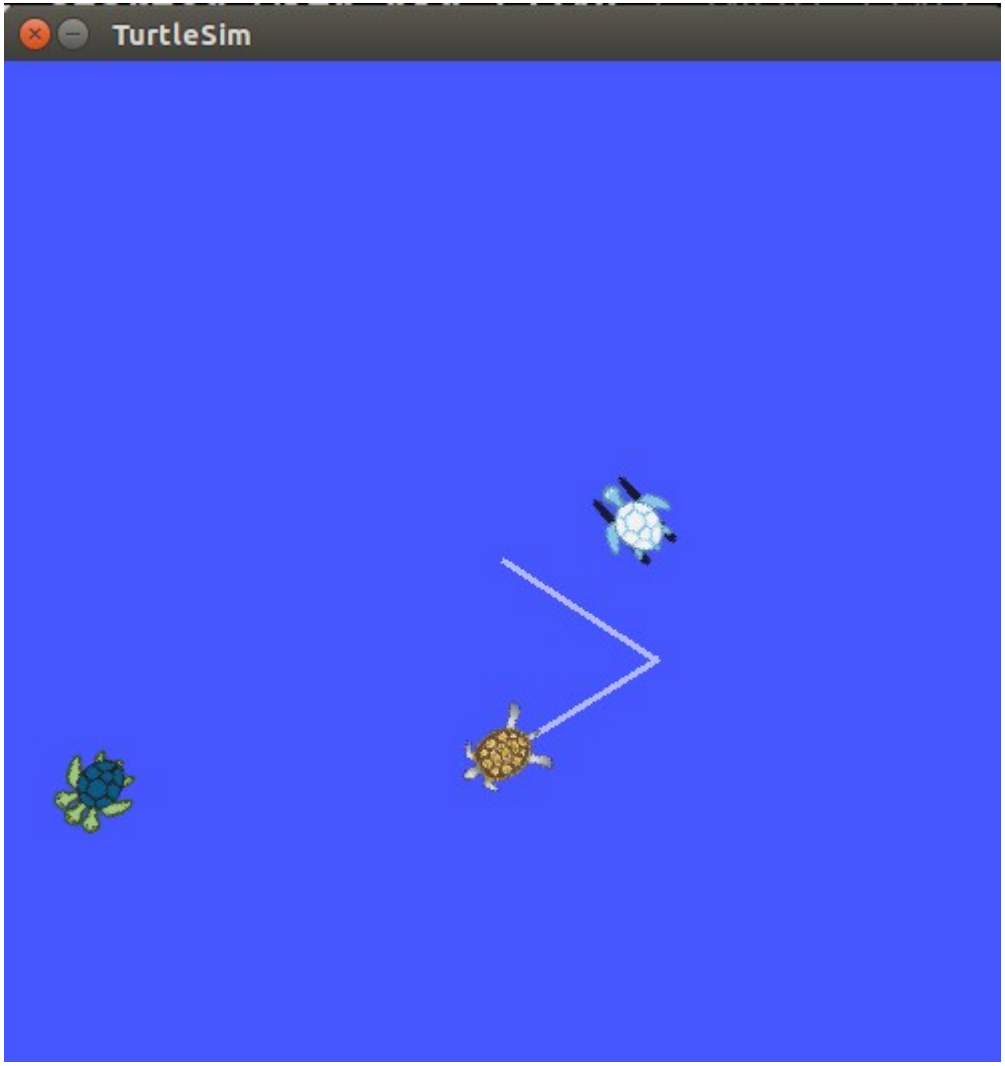


Reactive Paradigm Example



Reactive Paradigm Example




TurtleSim

Homework 4: Prerequisites

- ROS tutorial on launch files (#8):

  http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch
- ROS tutorial on services (#14)
- Turtlesim video tutorial:

  http://wiki.ros.org/turtlesim/Tutorials#Video_Tutorials

Homework 4: Part 1

- Create a new package called “cs378_<eid>_hw4”
- The package's dependencies should include the *turtlesim* package

Homework 4: Part 1

- For part 1, the task is to write a ROS node which adds a new turtle to the simulator
- After adding the new turtle, it should follow turtle1
- Include a launch file called “hw4_part1.launch” which should launch the simulator, your node and the keyboard teleop node to control turtle1

Homework 4: Part 2

Homework 4: Part 2

- For Part 2, you should implement three different ROS nodes, with each corresponding to the “turtle”, the “shark”, and the “fish”.
- Behavior:
  - “fish” should move randomly with low velocity
  - “shark” should follow the turtle
  - “turtle” should avoid the shark but try to get to the fish

Homework 4: Part 2

- A single launch titled “hw4_part2.launch” should launch all 3 nodes along with the turtlesim simulator
- 2 of the 3 nodes, the “fish”, and the “shark” should make a client call to the simulator to add a turtle that will represent them

Homework 4: Part 2

- At start time, the “fish” and the “shark” nodes should:
  - Use the /spawn service to add the agents to random positions on the board
  - Use the /<turtle_name>/set_pen service to set the fish's pen to yellow and the shark's pen to red
- At start time, the “turtle” node should use the /turtle1/teleport_absolute service to teleport to a random location
- As a result, all three agents should start at random locations and have different color pens so you can tell them apart

Homework 4: Part 2

Extra Credit:

- Write an additional node which monitors the positions of all three agents. If the turtle gets within a set threshold of the “fish”, then the node should call the /kill service on the fish. Similarly, it the “shark” gets within a threshold of the turtle, bye bye turtle :(
- Extra credit can also be gained if the turtle reasons explicitly about the boundaries of the stage (i.e., it should be turning away form the edge if it is running into the wall

Homework 4: Part 2

The README.txt file should contain:

- A detailed explanation in plain English describing how you implemented the behavior of each agent
- Comments on any completed extra credit
- Statistics on how often the turtle gets to the fish and how often the shark gets to the turtle. A successful completion of this assignment should result in both cases. To estimate the statistics, run your solution 10 different times (at least).

Homework 4: Part 2

- Due Friday March 4th
- What to turn in:
  - A zip of your package as it is in the catkin_ws/src folder
  - A README.txt file inside the package describing how you solved the problem and whether any extra credit was completed