Assignment 5 ROS

Rafal Grasman

Menno Broeders

**Manual**

The goal in this assignment is to learn using the actionlib from ROS.

To use the package:

1. Extract the package into your catkin workspace
2. Follow <http://wiki.ros.org/catkin/Tutorials/using_a_workspace> to build the package
3. Roslaunch assignment5 assignment5.launch

Then use in another terminal that also has access to the catkin workspace:

**Rostopic pub /cmd assignment5/Triangle 5.0 false**

To issue a triangle command with length 5 and clockwise false.

**Solution**

The turtlebot\_actions package has a server\_turtlebot\_move node whch is used as the actionlib server. Our draw triangle node has a client made with:

typedef actionlib::SimpleActionClient<turtlebot\_actions::TurtlebotMoveAction> Client;

Apart from that, the solution is exactly the same as assignment 2 except that the move commands have been replaced with actionlib:

void MoveRotate(double speed, double move\_distance, double rotate\_degrees)

{

std::cout << "a5n-m1" << std::endl;

Client client("/turtlebot\_move", true);

std::cout << "a5n-m2" << std::endl;

client.waitForServer();

std::cout << "a5n-m3" << std::endl;

// turn first

turtlebot\_actions::TurtlebotMoveGoal goal;

goal.forward\_distance = 0.0f;

goal.turn\_distance = rotate\_degrees;

client.sendGoal(goal, &SimpleDoneCallback, &SimpleActiveCallback, &SimpleFeedbackCallback);

client.waitForResult();

// then go forward

goal.forward\_distance = move\_distance;

goal.turn\_distance = 0.0f;

client.sendGoal(goal, &SimpleDoneCallback, &SimpleActiveCallback, &SimpleFeedbackCallback);

client.waitForResult();

std::cout << "a5n-m4" << std::endl;

}