Aria Homework 1

Noah Harvey

November 10, 2014

Code

asn3.cpp

```
1  #include <Aria.h>
2  #include <ArAction.h>
3
4  class ArActionFollowFar : public ArAction
5  {
6          public:
7                  ArActionFollowFar(double startAngle, double endAngle):
                       ArAction("farFollow","makes the robot go to the farthest
                       located object"), startAngle(-5), endAngle(5) {};
8                  virtual ~ArActionFollowFar() {};
9                  virtual ArActionDesired* fire(ArActionDesired currentDesired);
10
11         private:
12                 /** the start angle for the sonar sweep
13                  * @see checkRangeDevicesCurrentPolar()
14                  */
15                 double startAngle;
16                 /** the end angle for the sonar sweep
17                  * @see checkRangeDevicesCurrentPolar()
18                  */
19                 double endAngle;
20                 /** the range of the currently farthest detected object
                       relative to the
21                  * robot */
22                 double maxRange;
23                 /** the angle of the currently farthest detected object
                       relative to the
24                  * robot */
25                 double maxAngle;
26
27                 ArActionDesired myDesired;
28 };
29
30 ArActionDesired* ArActionFollowFar::fire(ArActionDesired currentDesired)
31 {
32         double range = 0;
33         double angle = 0;
34
35         //get farthest range
```

```
36          range = myRobot->checkRangeDevicesCurrentPolar(startAngle,endAngle,&
                angle);
37          if(range > maxRange)
38          {
39                  maxRange = range;
40                  maxAngle = angle;
41          }
42
43          return &myDesired;
44  }
45
46  void asn3(ArRobot* robot)
47  {
48          if(!robot)
49                  return;
50
51          /** add robot actions */
52          ArActionStallRecover recover;
53          ArActionBumpers bumpers;
54          ArActionAvoidFront avoidFrontNear("fAvoid", 1000, 400);
55          ArActionConstantVelocity constantVelocity("Constant Velocity", 400);
56          robot->addAction(&recover, 100);
57          robot->addAction(&bumpers, 75);
58          robot->addAction(&avoidFrontNear, 50);
59          robot->addAction(&constantVelocity, 25);
60
61          robot->enableMotors();
62          robot->waitForRunExit();
63  }
```