Aria Homework 4

Noah Harvey

November 17, 2014

Code

asn4.cpp

```
1  #include <Aria.h>
2  #include <ArAction.h>
3
4  #include "ArActionWallFollow.h"
5  #include "ArActionStopAct.h"
6
7  void asn4(ArRobot* robot)
8  {
9          if(!robot)
10                 return;
11
12         /** add robot actions */
13         ArActionWallFollow wallFollow(ArActionWallFollow::leftSide,0.2);
14         ArActionConstantVelocity constantVelocity("Constant Velocity", 400);
15
16         robot->addAction(&wallFollow,50);
17         robot->addAction(&constantVelocity,49);
18
19         robot->enableMotors();
20
21         while(wallFollow.nowall == false);
22 }
```

../../include/ArActionWallFollow.h

```
1  /**
2   * @file ArActionWallFollow.h
3   *
4   * AriaRobot Action class to follow a specified wall (either on right or left
5   * side of robot)
6   *
7   * @author Noah Harvey (nharvey@spsu.edu)
8   * @copyright GNU Public License 2
9   */
10
11 #include <Aria.h>
12 #include <ArAction.h>
13
14 //TODO: add code to follow a wall given a distance from it
```

```
15 class ArActionWallFollow : public ArAction
16 {
17         public:
18                 typedef enum
19                 {
20                         leftSide,
21                         rightSide
22                 } FollowSide;
23
24                 ArActionWallFollow(FollowSide side = leftSide, double dta = 5,
                        bool frange = false, double drange = 700) :
25                         ArAction("folloWall","ArAction to orient towards a
                            wall"),
26                         nowall(false),
27                         fside(side),
28                         deprange(frange),
29                         dist(drange),
30                         delta(dta)
31                 {};
32
33                 virtual ~ArActionWallFollow() {};
34
35                 virtual ArActionDesired* fire(ArActionDesired);
36
37                 bool nowall;
38
39         private:
40                 FollowSide fside;
41                 ArActionDesired myDesired;
42                 bool deprange;
43                 double range, dist, angle, dangle, delta;
44 };
```

../ArActionWallFollow.cpp

```
1  /**
2   * @file ArActionWallFollow.h
3   *
4   * AriaRobot Action class to follow a specified wall (either on right or left
5   * side of robot)
6   *
7   * @author Noah Harvey (nharvey@spsu.edu)
8   * @copyright GNU Public License 2
9   */
10
11 #include <Aria.h>
12 #include <ArAction.h>
13
14 #include "ArActionWallFollow.h"
15
16 ArActionDesired* ArActionWallFollow::fire(ArActionDesired cDesired)
17 {
```

```
18          myDesired.reset();
19
20          /** get sonar data */
21          range = myRobot->checkRangeDevicesCurrentPolar(-179,179,&angle);
22
23
24          /* get which side we're on */
25          if(angle < 0)
26                  fside = rightSide;
27          else
28                  fside = leftSide;
29
30          /** set the new heading based on the distance from nearest object */
31          if(deprange)
32                  dangle = angle + (2*fside-1)*90*(dist/range);
33          else
34                  dangle = angle + (2*fside-1)*90;
35
36          //stop if no walls on side
37          if(myRobot->checkRangeDevicesCurrentPolar(-90,90) > 2*dist)
38          {
39                  nowall = true;
40          }
41          else
42          {
43                  nowall = false;
44                  if(fabs(dangle) > delta)
45                  {
46                          //set the desired rotational velocity
47                          myDesired.setRotVel(fabs(dangle*.125));
48                          myDesired.setDeltaHeading(dangle);
49                  }
50                  else
51                          myDesired.setRotVel(0);
52          }
53
54          //ArLog::log(ArLog::Normal,"%f %f %f",range,dangle*range/200,dangle);
55
56          return &myDesired;
57  }
```