

## CSE 398/498 Robocup

**Team Challenge 1:** Reactive Obstacle Avoidance  
**Challenge Date:** Wednesday, 16 Feb 05

### A. Objective:

1. Your objective is to code up a reactive obstacle avoidance (OA) behavior for the Aibo using only feedback from the 2 on-board IR sensors.
2. You will implement this as a behavior in `ControllerGui`

### B. Requirements:

1. On the challenge date, representatives from each team will meet at PL450.  
NOTE: I realize that Wednesday lectures have been eliminated at this time, but because of cancelled lab sessions I am certain that you can indulge me.
2. An Aibo will be placed in an unknown environment with obstacles and an exit.
3. The same robot will run the OA behavior implemented by each team.
4. Each team will have 3 attempts to exit the “maze”. No code updates will be permitted between attempts.
5. The time to reach the exit for each team will be recorded, or the time to collision with an obstacle - whichever comes first.
6. In each trial, teams will be ranked based upon the fastest exit times, and then by longest collision times. Thus, if no team successfully exits without colliding, the team that survived the longest without a collision will be declared the winner.
7. All code must compile cleanly *without deprecations*.

### C. Some Potentially Helpful Tips:

1. I strongly suggest you re-review the beginner’s tutorial on Events and Motion Commands.
2. Sensor information can be obtained from the global state object `state` (see the documentation for `WorldState`).
3. Walk commands can be constructed similarly to the head motion commands in the motion example.

```
MMAccessor<WalkMC> walk_acc(walk_id);  
walk_acc.mc()->setTargetVelocity(50, 50, 0);
```

will enable `Main` to set the velocity of the Aibo to ( $vX=50$ ,  $vY=50$ ) by locking out the *MotionManager* and executing the `setTargetVelocity` motion command. There are velocity limits. You probably want to look under `WalkMC` for more details.

4. It will probably be necessary to pan the head around in order to locate obstacles well. Again, the motion example will be very helpful here.

5. For those of you who feel that you learn better through example, there is a rudimentary OA example by Salkin & Turner from the University at Albany at <http://www.ils.albany.edu/robotics/WalkForwardAndStop.h>

NOTE: Some this code is from a previous release, and may not compile correctly with our release.

**D. Turn in:**

1. A hard copy of your code, and a writeup  $\leq 1$  page from the team leader describing *in clean detail* the approach used.