UNSW COMP3431: Robot Architectures S2 2004

Assignment #3

Penalty Shooting

Due: Start of Lab, Week 12 (11am, 21 October 2004)

This goal of this assignment is build upon previous assignments to make a robot achieve a more complex task. In particular, in this assignment you will write a behaviour for a single role in a RoboCup soccer team - the penalty shooter.

Students will perform this assignment in small groups of 2 or 3. You are requested to work closely together on the project, developing the code together (rather than using the "divide and conquer" technique of splitting the project into parts, developing the parts separately, and then recombining).

0.1 Use of others' software

You may use the same code as in the previous two assignments. You may also build on the solution to assignment 2 from any group.

You already have access to the source code of a complete soccer playing robot, including behaviours to complete most of this assignment. For this assignment you are free to use any base-station code, any C++ code on the robot (accessed through the VisionLink module) and the basic python behaviour files on the robot listed here:

```
PyLib/*
PyCode/Behaviou.py
PyCode/Constant.py
PyCode/Debug.py
PyCode/Global.py
PyCode/HelpLong.py
PyCode/HelpShort.py
PyCode/HelpWhere.py
```

Note that parts of other files may be called by those files, and in particular Global.frameReset() calls HelpTrack.determineBallSource(). You should not add further references to other files, and you should look at the above code path to check that it does what you want if you use the global ball location variables.

You are *encouraged* to read the other python files, but please, no direct copying. You are also encouraged to build upon your code from assignments 1 and 2. You are encouraged to look at other RoboCup teams' approaches (please acknowledge any borrowed ideas in your report), although again, no direct copying of code. Video of the 2004 final is available on the web here: `http://www.openr.org/robocup/movies/2004movies.HTML`, and mirrored on the course web page (`http://www.cse.unsw.edu.au/~robocup/2004-final.mpg`).

0.2 Deliverables

Part 1 of this assignment requires you to develop behaviours on the robot. These behaviours will be handed in as follows: Before the start of the lab when the assignment is due, each group should have checked into the Subversion repository (`https://roborouter.cse.unsw.edu.au/svn/robocup/`) a branch containing their code. This branch should use Subversion correctly so that the `svn diff` command returns a reasonably sized output. Each group should also have a checked out working copy of their code with no local modifications on one of the lab machines. The code in that working copy should be compiled and ready for installation on a memory stick. The compiled code should have been tested on the robot before the class starts.

During the lab we will make memory sticks from those working copies. The solutions will be compared and graded. Your behaviour should have its own python module usable by the `spip` script. The name of the python modules should be included in the report.

The report in part 2 should be an electronic file checked in to the same branch as your code. It should be in an easily readable format, *e.g.* plain text, html, PDF or postscript.

1 Penalty Shooter (7 marks - graded in lab)

In this part of the assignment your team will develop code to make a RoboCup legged league penalty shooter. Your robot may play as either colour, shooting into the opposite coloured goal. There will be a goal keeper robot defending that goal using one of the python modules rUNSWift developed for the 2004 robocup competition.

The ball will initially be placed in the centre of the field. The penalty shooter will be placed by the team outside the centre circle. The referee will start timing and touch the back of the robot. When the robot scores, the ball will be moved back to the centre of the field, and the back button on the robot will be touched. If the attacker and goalie are entangled, they will be disentangled, but not moved significantly away from the goal.

The assignment will be graded based upon how many goals can be scored in 3 minutes[1].

Your robot will have to follow robocup rules (`http://www.tzi.de/~roefer/Rules2004/Rules2004.pdf` or `http://www.cse.unsw.edu.au/~cs3431/Rules2004.pdf`). In particular, you are not allowed to hold the ball for more than three seconds, and you are not allowed to push the goalie for more than three seconds. Pushing the goalie through the ball will be counted as pushing the goalie unless it is clear that there is no contact between the robots. 30 second penalties apply if these rules are broken. A goal is defined as having the *entire* ball behind the goal line. (Note that it is the rules for a forward that will apply to your robot, not the rules for a penalty

[1] It is the average time to score that is important. If goals are coming slowly then I may increase the time alotted so as to gather more samples for that average.

shooter. In the 2004 competition, the penalty shooter did not have to face an opposing goalie.)

2 Report (8 marks)

Describe the approach you used in the penalty shooter. Describe the strengths and weaknesses of that approach. Describe any testing or empirical comparison of different approaches you performed.