

Rules 2D Simulator League RoboCup-2005 Osaka

May 20, 2005

Abstract

This document contains the rules for the RoboCup-2005 Osaka 2D Simulation League competition. More information about this competition can be found at

<http://www.science.uva.nl/~jellekok/robocup/rc05/>.

1 Schedule

Every day the competition will start at 9:30 AM before which the team binaries should have been uploaded to the network. Competition lasts till 19:00 after which there is a teamleader meeting. The building will close at 21:00.

2 Soccer Server

Soccer server version 10.x will be used. In terms of the simulation there is no difference between 10.x and 9.4.5, but the 10.x version has better usability and platform support. **The used server configuration files (server.conf and player.conf) are provided on the aforementioned website..**

3 Tournament Modus

3.1 Automatic matches

All matches will be started automatically by script (or by referee). You need to provide 7 small scripts starting and killing your team. The scripts you need have to be called start, start1, start2, start3, kill1, kill2, and kill3; place them in top of your home directory (e.g. /home/robolog/start). Please note these important points:

- **Teams are only allowed to update their binaries before 9:30 AM to make sure games can run smoothly during the remainder of the day.**
- **Every team is allowed to use three machines for their team. Each machine has Xeon 3.4GHz processor and 1MB of memory and is running Gentoo (Linux)**

- Scripts will be executed by a different user in your user group. Your scripts and your team have to be at least group read- and executable.
- The scripts should use absolute paths or change to the respective directories.
- Double check that your kill scripts kill all of your programs (goalie, players, coach) – even if your programs terminate automatically.
- Teams can only use Linux operating system.
- The scripts will also be called for penalty kick out. Make sure your team can handle this.
- If it doesn't work, we don't fix it.

Please have a look at the sample start scripts located at the website. It should be fairly easy to adopt them for your team.

Teams will be published automatically after the competition. To help us with publishing your team, create a tar.gz file before the final round and rename it to /home/[teamdir]/teamdir.tar.gz.

In case a team fails to provide this file, we will publish the complete home directory of that team.

4 General Tournament Rules

The 2D competition will consist of 16 teams. The setup of the tournament is as follows:

- There will be three points allocated for a win and one point for a draw. A forfeit will record a score of either 3:0 or the score of the forfeiting team's other game in that round with the largest goal differential, if it is larger than 3.
- The Simulation League Team Competition will consist of three rounds.
- The first round will consist of four groups of four. Four teams are seeded and directly assigned to the different groups. The decision which teams are seeded, is made by the committee according to rankings of last years tournaments, i.e. Robocup 2004, and the local competitions like Japan, SSIL or German Open. Probably, this year this will be the first three teams of RC-2004 and the winner of SSIL. The seedings are necessarily somewhat subjective and not open to discussion. The other teams are randomly assigned to the groups.
- In the second round teams will be distributed to 2 groups of 8, where each team plays against all other teams of the same group. The assignments of a team to a group is based on the ranking of the first round as follows:
 - Group E: A1,B1,C2,D2,A3,B3,C4,D4

– **Group F: A2,B2,C1,D1,A4,B4,C3,D3**

The first two teams of each group will advance to the semi-final in the final round. The other teams will play for their final position in the final round.

- In the final round, the semi-finals (F1-E2, E1-F2), placing position (Ex-Fx) and final (winners semi-final) will be played. Each game must have a winner. 3000+3000 cycles of extra time will be played with the golden goal rule. If there is still no result, the game is decided by penalty shootout (see below).

5 Tiebreakers

Tiebreakers between $n \geq 2$ teams in the first two rounds will apply in the following priorities:

1. points
2. head-to-head results
for $n = 2$, this breaks the tie if and only if the head to head match was not a draw
for $n > 2$, this breaks the tie if and only if one team won against all other teams who are tied.
3. Overall goal difference for this round
4. if ($n > 2$), overall goal difference including only games with the tied teams
5. Overall number of goals scored
6. if ($n > 2$), overall number of goals scored including only games with the tied teams
7. Penalty shootout between the tied teams. Details of the structure will be explained further below.

If $n > 2$ teams are tied, then the above list of tiebreakers is used until 1 team can be put first. That one team is ranked first and the remaining teams are grouped into another tiebreaker, and the criteria are applied from the beginning of the list.

The exception to the above list: If teams are tied and it is the case that some team(s) in the tie will advance to the next round and some team(s) will not, then game statistic values will NOT be used to break the tie. Instead penalty shootout will be performed between the tied teams. If more than two teams are involved in the tie, the teams are first ranked using the game statistic values (numbers 3 through 6) above. Any remaining ties are broken uniformly randomly. This ranking is used to group the teams in a standard single elimination bracket (highest ranked team plays against lowest ranked team in the first round, etc.). With an uneven number of team, the highest ranked team is free in the first round. Each match follows the 2 team penalty shootout described below.

5.1 Automatic Penalty Shootout Procedure

To resolve tie-breaks, automatic penalty shootouts will be used. For penalty shootout, we are going to use these parameters:

```
pen_dist_x: 42.5 // 42.5 m from the goal is position 10.0
pen_allow_mult_kicks: true // allow multiple kicks so normal play
pen_taken_wait: 200 // nr of cycles waited after start pen.
```

(This means that the scorer starts 42.5 meters from the goal and can use dashes, turns and kicks. After at most 200 cycles, the penalty is stopped).

PLEASE SEE THE SPECIFICATION IN THE NEWS FILE FOR MORE INFORMATION ABOUT PENALTY SHOOTOUTS.

6 Free-Kicks and Kick-Ins

In certain situations, like free-kicks and kick-ins, the game is stopped.

If a team fails to put the ball back into play after a free kick, a drop-ball is given after 200 cycles automatically.

If repeatedly no player of the team that has to perform the free kick displays efforts to move towards the ball, the waiting time can be suitably shortened by the referee dropping the ball manually. The goal is always to keep the game running as smoothly as possible while giving the teams a fair chance to exert their rights.

If in a `play_on` situation no player goes to the ball, the referee can drop the ball after 200 cycles. Especially in this case, referees are encouraged to use some common sense. E.g. if a player needs to run a long way to get to the ball and is nearly there after 200 cycles they should be given the chance to take the kick before dropping the ball. The ball should not be dropped before the 200 cycles.

Ball drops should be as near as possible to the current position of the ball or on the corner of the penalty box.

7 Code of Honor

7.1 Coach Messages

The coach can issue arbitrary "freeform" messages during non-play-on mode.

The coach can send one advice, one info, and one define, every 30 seconds – the rest will be ignored by the server. Therefore, the Coach shall not send more than 4 of those standard-language directives per 30 seconds, so as to not flood the network.

7.2 Fouls

Free kicks and kick-ins are detected automatically by the soccer server in many relevant cases. Sometimes, however, fouls occur which can only be detected by the human referee who has to award a free kick to the disadvantaged team.

Reasons to call a foul are:

- if one team surrounds the ball so that the other team cannot kick
- if the goal is blocked by so many players so that the ball could not go in (rough guideline: a wall of players blocking the goal);
- if a team intentionally blocks the movement of opponent players;
- the number of goalie moves is limited to 2. It is possible to get around this by doing a small kick and catching again. This is allowed once then the referee is required to drop the ball on the closest corner of the penalty box (notice this practice is not encouraged we are just acknowledging the potential for mis-kicks - continual use may be considered violating the fair play commitment);
- anything else that appears to violate the fair play commitment may also be called as a foul after consultation with the rule committee.

8 Fair Play

The goal of the game is to play soccer according to fair and common sense understanding of soccer and to the restrictions imposed by the virtual simulated world of the soccer server. Circumvention of these restrictions is considered violating the fair play commitment and its use during the tournament games is strictly taboo.

Violation of the fair play commitment play includes for example:

- using another teams binaries in your team
- if a team is jamming the simulator by sending more than 3 or 4 commands per client per cycle;
- if a team communicates by other means than via the server using the 'say' command, for example by using direct inter-process communication;
- if a team attempts to disturb other teams communication by recording and sending strings of former communication or by attempting to fake communication of the opponent team.

Any of these is strictly forbidden.

Other strategies might be found violating the fair play commitment, after consultation with the rule committee. However, we expect it to be pretty clear what a fair team should look like. In particular, the destructive disruption of opponent agent operation or the gain of advantage by other means than explicitly offered by the soccer server count as not fair play. If you are in doubt of using a certain method, please ask the simulator rule committee before the tournament starts. If a team is found to use unfair programming methods during the tournament, it will be immediately disqualified.

If a team is under suspicion of violating the fair-play agreement, the committee has the right to ask for source code inspection.

9 Remote Participation

General Remark: *Remote Participation is only possible in extreme cases.*

We will not have the resources to search for problems in remote participants' startup procedure, so, in their own interest, remote participants are asked to make sure that a 3rd party (i.e. we) can start up your code easily and smoothly on a platform that might be different from their development platform. We unfortunately are not able to guarantee the participation of a remotely participating agent team if we do not succeed in getting the code quickly and smoothly to run.