

# RoboCup Soccer Simulation 3D League

Rules for the 2023 Competition, Bordeaux, France (July 4<sup>th</sup> – July 9<sup>th</sup>)

These rules are based on RoboCup 2015 to 2022 Competition Rules

## 1 Organizing Committee

This document describes the rules of the RoboCup 2023 3D Soccer Simulation League. The goal is to provide for fair games and smooth execution of the competitions. The Organizing Committee (OC) has the right to append or change these rules at any time within reasonable bounds. All teams will be notified of these changes as soon as possible. Teams can appeal to these changes or the rules in general. However, the OC will make the final decision. The organizing committee for RoboCup 2023 simulation 3D league is composed by Klaus Dorer ([klaus.dorer@hs-offenburg.de](mailto:klaus.dorer@hs-offenburg.de)), Marco Simões ([msimoes@uneb.br](mailto:msimoes@uneb.br)) and Hannes Braun ([hbraun@stud.hs-offenburg.de](mailto:hbraun@stud.hs-offenburg.de)).

## New Rules and Server Changes for RoboCup 2023

The Technical Committee (TC) has approved the following changes and new rules:

Rule changes (for details, see the rule descriptions). The constants used for the new rules are still subject of fine tuning and might be changed before the competition by the tc or during the competition, if all teams agree.

- **Ball holding:** A player is considered to be holding the ball if, during PlayOn mode, the ball is within *BallHoldRadius* meters of the player longer than *BallHoldMaxTime* and no opponents are closer to the ball.
- **Goalie catch:** A new catch command allows the goalie to stick the ball to its hands for at most *CatchTime* seconds. It succeeds if the ball is close enough to one of the hands and is not too fast.
- **New penalty format:** all four players involved in penalty shootouts (two strikers and two goalies) are on the field simultaneously. One team scores to the left, the other to the right goal. A penalty is a single shot from *PenaltyShootoutBallDistance* meters away from the goal to score. The player has to walk to the ball and kick a single time. Teams have to provide a single start script for a penalty shootout named *start\_penalty.sh*.

Technical changes

- Increased camera frequency: The camera frequency has been increased from 16  $\frac{2}{3}$  Hz (every third cycle) to 25 Hz (every second cycle)
- No agentproxy will be used in case of all processes running on a single machine
- We try to use Ubuntu 22.04, but might end up using an older version
- A New RoboViz version required because of ball-holding fouls

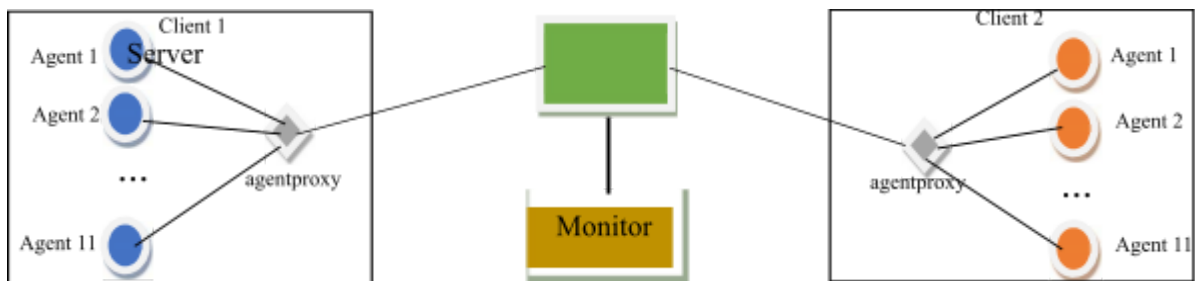
## 2 Competitions and Schedule

### 2.1 Server and Server Changes

The competition will use the latest version of Simspark (0.3.5) and rcserver3D (0.7.6) and Linux Ubuntu 22.04 64 Bits (the last is still pending, but be prepared to run on it). See the latest versions at <https://gitlab.com/robocup-sim/SimSpark/-/releases>. A Java 8 Runtime will be available for clients, as well as a recent Python. Clients have to work with or without the latest version of agentproxy (now 2.1.4) available at <https://github.com/magmaOffenburg/magmaProxy> to connect to the server. The following figure shows the software and hardware setup for the competition's environment. We distinguish two kinds of setup:

1. There are two client computers – one for each team – named *client1* and *client2*; one *server* computer; and one *monitor* computer (roboviz). The clients connect through an agentproxy running on the local computer to the server (see figure below).
2. All client processes and the server run on a single computer. In that case, no agentproxy will be used.

All teams are strongly encouraged to test their code using an environment similar to the competition's environment to avoid problems with faulty agents during competition. **The OC will not install any additional libraries. So, teams MUST manage to build their binaries using static links to all libraries they need.**



Teams should refer to the complete information available at <https://ssim.robocup.org/3d-tools/> for the latest version changes and other information.

The OC can run the complete setup in a single remote high-performance computer if the internet in the venue allows using this setup.

### 2.2 Competition Format and Challenges

#### General Format and Schedule

The competition includes four rounds (seeding round, first, second and third round) followed by semifinals and finals. Teams not qualified in any round play additional matches for the full ranking. The competition also includes three challenges: Free/Scientific Challenge, a Kick a Rolling Ball Challenge and a FatProxy Challenge. The times will be fixed at a later point, but the tentative format and schedule are the following:

July 4<sup>th</sup> – Tuesday: Team warm-up and Competition setup

July 5 <sup>th</sup> – Wednesday:	Team warm-up and Competition setup Team Leaders Meeting (TLM) & Group Drawing Seeding Round (code upload deadline [Cud] 30 minutes ahead of start)
July 6 <sup>th</sup> – Thursday:	Opening Ceremony First Round (Cud: 30 minutes ahead of start) TLM, TC Election and League Discussion
July 7 <sup>th</sup> – Friday:	Second Round (Cud: 30 minutes ahead of start) Free/Scientific Challenge TLM
July 8 <sup>th</sup> – Saturday:	Third Round (Cud: 30 minutes ahead of start) FatProxy Challenge Kick Rolling Ball Challenge Full-Ranking Rounds (5 <sup>th</sup> place onwards)
July 9 <sup>th</sup> – Sunday:	Semi Finals (Cud: 30 minutes ahead of start) 3 <sup>rd</sup> place match (Cud: 15 minutes ahead of start) Finals (Cud: 15 minutes ahead of start) Award Ceremony

This schedule is subject to the general schedule definitions and the number of teams effectively participating in the competition. Both format and schedule may be adapted by the OC as needed. The OC will define the schedule of matches for defining the full ranking according to the number of teams not-qualified in each round.

## Qualifying Rounds

For the seeding and first rounds, all teams may be included in a single group or be split into two or more groups, according to the total number of teams. All teams play a round-robin within their groups. For the second round, one group of 6 teams will play a round-robin. If more than 6 teams are qualified for the second round, they may be split into two or more groups. In the third round, one group of 5 teams will play a round-robin. If more than 5 teams are qualified for the third round, they may be split into two or more groups. Each team plays with every other team in the group once. The following criteria define the ranking in each round:

- Points gathered. A win gives 3 points, a tie 1 point, and a loss 0 points.
- Goal difference (goals scored minus goals conceded) in this round.
- Goals scored in this round.
- Head-to-head games of this round.
- An extra match (only in cases where it matters for advancing to the next round)
- Two extra halves of 3 minutes each
- Penalty shoot-outs (for 2-team ties only)
- Coin toss

## Semifinals and Finals

After the third round, a knockout system is used. The matches are defined according to the ranking in the third round. The first place plays against the fourth place for the first semifinal, and the second place plays against the third place for the second semifinal. The winners of these matches will play against each other in the final. The losers will play a match to decide who

wins the third prize. If a knockout game ends in a tie, two extra halves of 3 minutes are played. A penalty shootout determines the winner if these also end in a tie. In the knockout matches, the winner will be determined in the following order:

- Number of goals scored in the game
- Two extra halves of 3 minutes each
- Penalty shoot-out
- Coin toss

### **Scientific/Free Challenge**

The Simulation 3D scientific/free challenge is a presentation/demonstration competition that intends to encourage scientific work development and sharing in the context of the Simulation 3D league, allowing the teams to present interesting research work developed using Simspark and RCSSServer3D simulation platform. Each team will be given five minutes to present and demonstrate an interesting research topic developed by their team. Teams should deliver to the technical committee a short, one-page description of their intended presentation before the competitions (deadline 24 hours before the challenge start) to be distributed to all other teams and the jury during the competition. Presentations must be focused on only one significant research topic of the team and should not describe the whole team like in a TDP. Besides describing the scientific issue, presentations must also show a live demonstration using the RCSSServer3D simulator. Demos should show the real-time execution of the simulator. However, they may also include pre-recorded videos and demonstrations of other related software tools. A voting process finds the winner.

### **FatProxy Challenge**

The magmaOffenburg team released a fatProxy software that can replace the agentproxy (see Figure 1). It allows controlling the robots using high-level dash and kick commands. It is thought of as a means to have a common interface for 2D and 3D teams that want to use the 3D simulator. More details on how to use the magmaFatProxy and its perfect kick abilities can be found at <https://github.com/magmaOffenburg/magmaFatProxy>.

The FatProxy Challenge will be a tournament for all teams wanting to participate. The team's rank is the criteria for getting points (see below).

### **Kick Rolling Ball Challenge**

The magmaOffenburg team released a magmaChallenge software that will be used to run this challenge (<https://github.com/magmaOffenburg/magmaChallenge>).

In this challenge, a single player has to try to kick the ball rolling toward it. The ball's initial position is at a random distance of 1 to 4 m away from the player on a random angle between -90 to 90 degrees. The player has 5 seconds to find the ball. After 5s, the ball is kicked by the tool to a point 0.15m in front of the player with a noise of  $\pm 0.05$ m in the x and y direction and a speed that depends on the distance and is chosen so that the ball will reach the player.

The start player script (startChallengePlayer.sh) has to start a player that will act as a kicker when it receives "RollingBallChallenge" as the challenge name argument. It waits or steps in place at its spot until the ball arrives.

The final score is calculated as  $dx - \text{abs}(dy) - \text{fallen} * 3$ , where

- dx is the distance of the ball when stopped to the initial beaming position of the player or 0 if closer to the own goal
- abs(dy) is the absolute value of the ball's y coordinate when stopped
- fallen is 1 if the player did fall before the episode stops, otherwise 0

An episode ends if the ball was kicked by the tool and reaches a speed below 0.01 m/cycle. This means that it is not allowed for the player to stop the ball and kick it in sequence.

### **Challenge Trophy Rules**

A trophy will be awarded to the team with the highest combined score on the three challenges. The following rules will calculate the combined score:

- Each participating team in a challenge is given a score between 5 and 25, with the winner of a challenge receiving 25 points and the last placed team in a challenge receiving 5 points.
- Teams not finishing first or last in a challenge receive a score based on a linear equation of their finishing spot in the challenge.  $\text{score} = (\text{number\_of\_participants} - \text{rank}) / (\text{number\_of\_participants} - 1) * 20 + 5$ . Example: If five teams participate in a challenge, then 1st place = 25, 2nd = 20, 3rd = 15, 4th = 10, 5th = 5.
- Teams not participating in a challenge receive 0 points for that challenge.

The overall winner of the technical challenge is the team with the highest total sum of points earned across all the challenges. In case of a tie, the best team in the Free/Scientific Challenge will be declared the Winner of this Trophy.

## 4. General Rules

### Match Preparation and Format

1. Set-up: The OC will supply upload instructions on the first day, which teams must use and follow to run their agents. Teams will be allowed to upload and test their agents on the competition system before the start of each round. When a round is started, it will no longer be possible to upload or make changes to the agents until all matches of that round have been played.
2. Start/end: A competition managing system will automatically start the matches. Teams are required to provide scripts named *start.sh* and *kill.sh* for running and terminating their team. All team players must run in less than 3 seconds when *start.sh* is executed. Additionally, each team must provide a script named *start\_penalty.sh* to run their penalty kicker and goalie in a single script. The IP address the agents should connect to is the first parameter to the start scripts. Examples of start and kill scripts can be found on the SSIM3D wiki.
3. Team format: Two teams of 11 players play a match. A match may not start if either team consists of fewer players. Each match consists of two halves of five minutes each. After the first half, the teams switch sides. Each player can select any available agent models, possibly forming a heterogeneous team. The details of the available models were previously announced. The possible team composition is described in the heterogeneous player section of this document.
4. Representation: For a match, each team can appoint a single team member as the team's representative. This representative should announce himself to the referee before the start of the match; otherwise, it is assumed that the team has none for this match. During the match, only the team's representative can approach the referee.
5. Forfeiting rule: We define forfeiting as refusing to make a good-faith effort to participate in a scheduled game. A team that forfeits is disqualified from the competition. In round robins, when a team forfeits, all the scores will be 3x0 against this team. In a knock-out round, the teams who forfeit will be replaced by the next team in the rank from the previous round.
6. Faulty agents: When a match has started, and an agent shows clearly abnormal behavior or gets disconnected, the following rules apply:
  - If this occurs in the first 30 seconds, the match will be restarted up to 2 times. In the case of abnormal behavior, the team's representative is responsible for notifying the referee, who will decide whether to restart the game at his own discretion.
  - If the problem persists after restarts, the team's representative can ask the referee for a 4-minute time-out to fix the problem. A fix may include the use of a binary from a previous round.
  - If the problem persists after the time-out, or if it occurs after the first 30 seconds of the match, the game will continue. If, due to the problem, the team ends up with less than the minimum number of players, the match will result in a win by the average score against this team but at least 1:0.

## Heterogeneous Players

Each team has to use a minimum of 3 different player types and no more than 7 players from one player type. If only 3 different player types are used then, a minimum of 2 players for each type must be used. The standard player type is counted identically to heterogeneous player types. Possible configurations for the different player types are: (7-2-2), (7-1-1-1-1), (4-4-3), (3-3-3-2), (3-2-2-2-2), etc. If teams do not comply with this rule, the server will not allow the faulty player to connect (either the 8th player of a given player type or the first player that does not allow the '3 player types' rule to be fulfilled). The heterogeneous types used last year will also be used during this year's competition.

## Game Play

1. **Kickoff: Goals** scored directly from the kickoff are not accepted. Also, the agent performing the kick-off may not touch the ball again until another player touches it. In case of infringement, a kick-off is awarded to the opposing team. A goal can be scored after a kickoff only if a second player from the team performing the kickoff touches the ball outside the main circle or if an opponent touches the ball.

2. **Dead Ball situations:** Dead Ball situations are Kickoff, Free Kick, Kickin, Corner Kick, and Goal Kick. Whenever the automatic referee assigns a Dead Ball situation, the team supposed to kick the ball must do it in *RuleDropBallTime* seconds. After that time, a different decision is made by an automatic referee for each situation:

- Goal Kick: Drop Ball in the upper corner of the goal area
- KickIn: KickIn assigned to the opponent
- KickOff: Indirect Free Kick assigned to the opponent
- Corner Kick: Drop Ball in the same position where the ball is
- Indirect Free Kick: Indirect Free Kick assigned to the opponent

*RuleDropBallTime=20*

3. **Goalkeeper:** The player with uniform number 1, and only this player, will be designated as the goalkeeper. The goalkeeper can catch a ball with his hands using the catch command. It will only succeed if the ball is near one of the hands and not too fast. The goalkeeper has to call the catch command each cycle until he wants to release the ball. The ball is released after 6 seconds and can not be caught again for 3 seconds.

4. **Handling the ball:** Handling the ball involves a deliberate act of a player contacting the ball with his hand or arm, as decided by the referee at his discretion. Handling the ball is penalized with a free kick to the opposing team. The goalkeeper cannot be guilty of a handling offense when his hands are within his penalty area.

5. **Obstructing the ball:** Obstructing the ball involves the use of a player's body, arms or legs to prevent the progress of the game, which includes covering the ball, carrying the ball, holding the ball between arms or legs, and lying in front of the goal in an attacking

situation. Deliberately obstructing the ball, as judged by the referee, for longer than 10 seconds is penalized with a free kick to the opposing team. A drop ball ends with unintentional obstruction of the ball for longer than 10 seconds, as decided by the referee.

6. **Illegal defense:** At any time during the match, no more than 3 players of a team may be inside their penalty area. If a 4<sup>th</sup> player enters the penalty area, it will be placed outside of the penalty area. When the last to enter is the goalkeeper, the player furthest away from the own goal of the three that were already in the penalty area will be placed outside of the penalty area instead.

7. **Incapable robot:** Players that have been immobile or fail to stand up for too long will be charged for an *incapable foul* and be repositioned. The goalkeeper will be repositioned after 30 seconds of immobility or after failing to stand up for 60 seconds; other players will be repositioned after 15 seconds of immobility or after failing to stand up for 30 seconds.

8. **Charging fouls:** Players must avoid touching other players. The situations where an illegal charge of one player to another one is considered as a foul are described at <http://www.acso.uneb.br/bahiart/uploads/Main/FoulModel2016.pdf>. The automated referee will call such fouls. Fouling players will be beamed outside the field as with other fouls.

9. **Touching:** If a group of three or more players touches each other simultaneously, one team player with the most touching robots is moved away from the point of collision. This might be the case if three fallen robots lay on each other, in which case no charging foul is called. The rule has been introduced to reduce the number of robot-to-robot collisions making the server more stable.

10. **Self-Collisions Fouls:** A self-collision foul will be charged to a player that violates the normal physics rules by self-colliding their body parts. There is a tolerance of *SelfCollisionsTolerance* m to a robot's body part trespassing on another body part of the same robot. This tolerance is defined to avoid false positives from the self-collisions detector. When a robot is charged for a self-collision foul, its joints involved in the self-collision will be frozen during *SelfCollisionJointFrozenTime* seconds. To freeze a joint means setting its velocity to zero and ignoring all commands the agent sends to move this joint. After the frozen period, the joints cannot be frozen again within *SelfCollisionJointThawTime* seconds. The simulator's current self-collision parameters are:

- *SelfCollisionsTolerance* = 0.04
- *SelfCollisionJointFrozenTime* = 1.0
- *SelfCollisionJointThawTime* = 2.0

11. **Ball Holding:** A player is considered to be holding the ball if, during PlayOn mode, the ball is within *BallHoldRadius* meters of the player and no opponents are closer to the ball. A player is penalized for holding the ball if they continuously hold the ball for *BallHoldMaxTime* seconds (*BallHoldGoalieMaxTime* seconds if the player is a goalie with the ball in their own penalty area) or the ball travels at least *BallHoldMaxDistance* from the ball's position when it was first held, and an opponent is within *BallHoldOppDistance* of the ball. The following are tune-able parameters for the ball holding penalty:

- *BallHoldRadius* (0.12): Radius in meters from a player's center in which the ball is considered being held
- *BallHoldMaxTime* (5.0): Max time in seconds a player can hold the ball



- *BallHoldGoalieMaxTime (10.0)*: Max time in seconds a goalie can hold the ball while the ball is in their own penalty area
- *BallHoldResetTime (0.5)*: Time in seconds since a player stopped holding the ball before ball holding time resets
- *BallHoldMaxDistance (1.0)*: Max distance in meters the ball can travel while a player is holding the ball
- *BallHoldOppDistance (0.75)*: How close an opponent must be from the ball in meters for a holding penalty foul to be called (a negative value causes a foul to be called independent of where opponents are)
- *BallHoldBeamPenalty (false)*: Use beaming to penalize ball holding instead of repositioning a player off the ball

You have to use the latest version of RoboViz that knows about BallHolding fouls.

12. **A Pass command:** A Pass command can be used by agents during games to initiate a Pass mode for their team. An agent should send the simulator a "(pass)" message to use this command. After an agent successfully initiates Pass mode with the Pass command for that agent's team, all opponents will be prevented from getting closer than 1.0 m to the ball during Pass mode duration. A Pass mode lasts until a team member touches the ball or *PassModeDuration* seconds have passed. Once Pass mode for a team ends, team members must wait at least *PassModeRetryWaitTime* seconds before a team member can initiate Pass mode again for their team. When any robot of a team initiates Pass mode, the team can only score if at least one of the following conditions are met: i) *PassModeScoreWaitTime* seconds have gone by from the time Pass mode ends; ii) the PlayOn playmode after Pass mode ends transitions to something else (e.g., the ball goes out of bounds); iii) two players from the team who started the Pass mode touch the ball with at least one of the touches occurring after the ball had left the pass mode area (a circle with radius 1.0 m around the position of the ball when Pass mode was started). If a goal is scored that violates these requirements, it will not be counted, and a goal kick will be awarded to the opponent's team. There are some requirements for an agent to successfully use the Pass command to initiate Pass mode for their team. First, the ball should be at no more than a maximum speed of *PassModeMaxBallSpeed* m/s. Second, the ball should be within a maximum distance from the agent calling the Pass command of *PassModeMaxBallDist*. Third, no opponent agent can be within a circular region with a radius of *PassModeMinOppBallDist* m around the ball. The Pass command will be accepted by the server only if all these three conditions are met. Otherwise, the server ignores the command, and Pass mode is not initiated. The parameters defined by the TC are:

- *PassModeMaxBallSpeed = 0.05*
- *PassModeMaxBallDist = 0.5*
- *PassModeMinOppBallDist = 1.0*
- *PassModeDuration = 4.0*
- *PassModeScoreWaitTime = 10.0*
- *PassModeRetryWaitTime = 3.0*

13. Teams who use players purposely **crouching down** on the ground with legs splayed out that may trip opponents while not playing the ball will be penalized with a free kick assigned to the opponent team. The human referee will call this foul.

14. **Penalty shoot-outs:** A kicker and a goalie from both teams join simultaneously. (4

players in total are on the field.) The kicker is expected to be behind the ball. The minimum distance is specified through *PenaltyShootoutBehindBallDistance*. If the player doesn't obey this, it will be automatically moved far enough away from the ball. The ball position is specified with *PenaltyShootoutBallDistance*.

When the game is started by switching the play mode to *KickOff\_Left*, the kicker from the left team has to kick toward the right goal. With *KickOff\_Right*, it's the other way around. The active kicker is only allowed to touch the ball once. When the ball is touched, the play mode is changed to *PlayOn* until a goal is scored (at which point the play mode changes to *Goal\_<Left or Right>*). If

- the striker wasn't able to score in time (see parameter *PenaltyShootoutMaxTime*),
- the striker touched the ball for a second time,
- the ball stops moving (speed < 0.002), the play mode will be set to *BeforeKickOff*.

After some seconds, the play mode will automatically switch to *KickOff\_<Left or Right>*, and it's the other striker's turn. Goalies are not allowed to move outside of the penalty area. If neglected, it will be considered a goal for the attacking team. The game ends after a specified minimum number of rounds (*PenaltyShootoutMinRounds*) if one team scores more goals than the other team. If after these penalty shots, the score is still equal, sudden death rounds of one kick each are used until one side scores, and the other does not. After a specified maximum number of rounds (*PenaltyShootoutMaxRounds*), the game will be over regardless of the score. A coin-toss then determines the winner.

- *PenaltyShootoutBehindBallDistance*: 4m
- *PenaltyShootoutBallDistance*: 6m
- *PenaltyShootoutMaxTime*: 25s
- *PenaltyShootoutMinRounds*: 5
- *PenaltyShootoutMaxRounds*: 10

15. **Output:** Each player is permitted to generate output in files named *stdout* and *stderr* with the player number appended (e.g. *stdout3* and *stderr3* for player number 3). These write-only files will be located in their home directory under a *log* subdirectory. Except for these files, the home directory is read-only during the games, so agents should not try to open any other file for writing.

16. **Fair play:** Games should be played according to a fair and common-sense understanding of soccer and restrictions imposed by the virtual simulated world of the 3D soccer simulator. Circumvention of these is considered to violate the fair play commitment and is strictly forbidden during the tournament. Violation of the fair-play commitment includes, but is not limited to, for example:

- using another team's binary in your team
- jamming the simulator by sending excessive commands per client
- direct communication between players by using communication means other than those provided by the simulator, such as inter-process communication
- manipulating competition machines or restarting them intentionally
- unnatural behaviors that exploit weaknesses of the simulation

Any of these is strictly forbidden. Other strategies might be found to violate the fair play commitment after consultation with the OC. However, we expect it to be clear how a fair team should behave. In particular, the destructive disruption of opponent agent operation or the gain of advantage by other means than explicitly offered by the simulator counts as unfair play. If you are in doubt about using a particular method, please ask the OC before the tournament starts. If a team is found to use unfair programming methods during the tournament, it will be immediately disqualified.

## **Referee**

During each game, a referee is appointed to enforce the rules in this document. In situations not handled by the rules, the referee should decide based on his/her common sense. The referee can be a member of the OC or a volunteer from a team other than the teams involved in a match who has full knowledge of the rules as deemed by the OC. During a match, only one representative of each playing team is allowed to approach the referee. The referee's decisions are binding and cannot be appealed during a match. If there is any complaint about a game or a decision, this should be brought to the OC's attention as soon as possible, who will decide how to handle it after all games of the current round are finished. If the referee cannot decide in a situation, he can consult the OC, who will give the final decision. Each team must supply a volunteer to be a referee, preferably the same as the team's representative.

## **Publication of Logs and Binaries**

The OC will make the logfiles of the matches available to the participating teams during the competition. After the competition, the OC will also publish these log files online at their earliest convenience. The OC will also publish the binary and additional files needed to run this binary of each team after the competition. To do this, at the start of each round, the home directory of the teams participating in that round will be archived. The team's last archive will be used for publication. If the home directory contains material that a team wishes not to be published and that is not needed to run the binary, it is the team's responsibility to notify the OC of this. Note that teams that have used third-party source code must follow that code's license regarding conveying the work. For instance, under GPL, you must accompany your binary with the corresponding source, a written offer to supply the corresponding source, or information about where the source is offered to the general public. Failing to comply with any licensing will result in the disqualification of the team from the current and/or future competitions.