RoboCup Soccer Simulation 3D League

Rules for the 2022 Competition, Bangkok, Thailand (July 11th – July 16th)

These rules are based on RoboCup 2015 to 2021 Competition Rules

1 Organizing Committee

This document describes the rules of the RoboCup 2022 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 2022 simulation 3D league is composed by Luis Paulo Reis (lpreis@fe.up.pt), Marco Simões (msimoes@uneb.br) and Miguel Abreu (m.abreu@fe.up.pt).

New Rules and Server Changes for RoboCup 2022

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

Rule changes

- The time for a team to kick the ball in a Dead Ball situation (e.g. corner kick, goal kick, kickOff, kickIn , free kick) is increased to 20 seconds. The *naosoccer sim.rb* file is setup with the following parameter:

  RuleDropBallTime=20
- Start time modification. All players of a team must be running in **less than 3 seconds** when start.sh is executed. This change is a result form the bug fix described below. To achieve this deadline, teams must remove any *sleep* commands from their *start.sh* scripts.

Technical changes

- Fixed a bug in the server during agents' connection time. The server usually has a warning message about "*Allow clients to connect/disconnect in the same frame*" and after this, it freezes, and the remaining agents cannot connect to the server. In the last years, teams were asked to add a *sleep* command in their start scripts to avoid agents connecting in the same time frame. This bug fix makes agents spawn in different places on the field. So, the server will not freeze anymore during agents' connections and the *sleep* between agents' connections is not needed anymore. It is important that all teams can test your binaries using this new release and remove the *sleep* commands from your *start.sh* scripts

2 Competitions and Schedule

2.1 Server and Server Changes

The competition will use the latest version of Simspark (0.3.3 released soon) and rcssserver3D (0.7.4, released soon) and Linux Ubuntu 18.04 64 Bits. See the latest versions at https://gitlab.com/robocup-sim/SimSpark/-/releases. A Java 8 Runtime will be available for clients, as well as Python 3.6.9. Clients have to work with 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. There are two client computers – one for each team – named *client1* and *client2*; one *server* computer; and one *monitor* computer (roboviz). 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.**

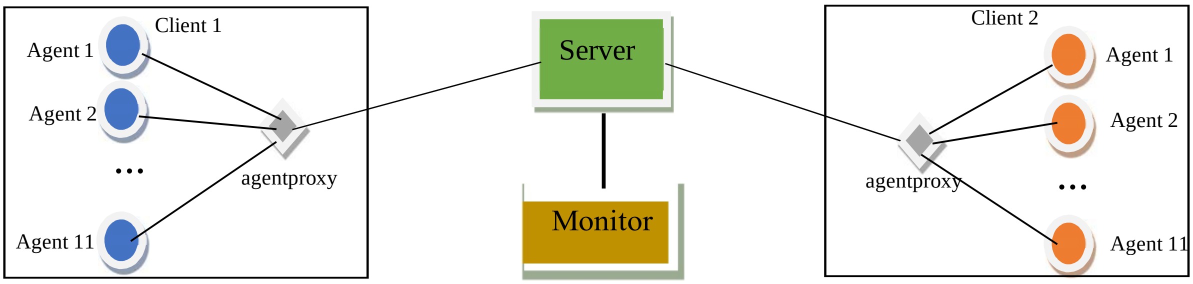

Figure 1: System Setup

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 try to run the complete setup in a single remote high-performance computer if the internet in the venue allows using this setup, and all teams agree with this.

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 two challenges: Free/Scientific Challenge and a FatProxy Challenge. The times will be fixed at a later point, but the tentative format and schedule are the following:

July 11th – Monday:	Team warm-up and Competition setup
July 12th – Tuesday:	Team warm-up and Competition setup Team Leaders Meeting (TLM) & Group Drawing Seeding Round (code upload deadline [Cud] 30 minutes ahead of start)
July 13th – Wednesday:	10:00 – 12:00 – Opening Cerimony

	First Round (Cud: 30 minutes ahead of start)
	TLM, TC Election and League Discussion
July 14th – Thursday:	Second Round (Cud: 30 minutes ahead of start)
	Free/Scientific Challenge
	TLM
July 15th – Friday:	Third Round (Cud: 30 minutes ahead of start)
	FatProxy Challenge
	Full-Ranking Rounds (5th place onwards)
July 16th – Saturday:	Semi Finals (Cud: 30 minutes ahead of start)
	3rd place match (Cud: 15 minutes ahead of start)
	Finals (Cud: 15 minutes ahead of start)
	15:00 – 17:00 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 schedule of matches for defining the full ranking will be defined by the OC according to the number of teams not-qualified on each round.

Qualifying Rounds

For the seeding round and first round, 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 there are more than 6 teams qualified for 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 there are more than 5 teams qualified for third round they may be split into two or more groups. Each team in each group plays with every other team of the group once. In each round, the following criteria defines the ranking:

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

Semifinals and Finals

After the third round, a knock-out 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 knock-out game ends in a tie, two extra halves of 3 minutes are played. If these end in a tie as well, a penalty shootout determines the winner. In the knock-out 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 simulation technical committee a short, one-page description of their intended presentation prior to the competitions (deadline 24 hours before challenge start) in order 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. Presentations, besides describing the scientific issue, must also show a live demonstration using the RCSSServer3D simulator. Demos should show real-time execution of the simulator. However, they may also include pre-recorded videos and demonstrations of other related software tools. The winner is found by a voting process.

FatProxy Challenge

The magmaOffenburg team released a fatProxy software, that can replace the agentproxy (see Figure 1). It allows to control the robots using high level dash and kick commands. It is thought of 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 of all teams that want to participate in it. The rank of the team is the criteria for getting points (see below).

Challenge Trophy Rules

A trophy will be awarded for the team with the highest combined score on the two challenges. The combined score will be calculated by the following rules:

- 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 place 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. score = (number_of_participants - rank) / (number_of_participants - 1) * 20 + 5. Example: If there are five teams participating 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 challenges 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 be able to run their agents. Teams will be given the opportunity 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: The matches will be started automatically by a competition managing system. Teams are required to provide scripts named start.sh and kill.sh for running and terminating their team. All players of a team must be running in less than 3 seconds when start.sh is executed. Additionally, each team must provide scripts named start_penalty_kicker.sh and start_penalty_goalie.sh to run their penalty kicker and goalie respectively. The IP address that the agents should connect to is given as the first parameter to the start scripts. Examples of start and kill scripts can be found on the SSL wiki.

3. Team format: A match is played by two teams of 11 players. 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 has the chance to appoint a single team member as the team's representative. This representative should announce herself 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 forfeits will be replaced by the next team in the rank from 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 of the match, 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 make the binding decision whether to restart the game on his own discretion.
- If the problem persists after restarts, the team's representative can ask the referee for a 2-minute time-out to fix the problem. A fix may include the use of a binary from a previous round.
- If the problem still 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 (that is 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 that were 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 it was touched by another player. 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 kickoff touches the ball outside the main circle or if an opponent touches theball.

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

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 on his own 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 own penalty area.

5. **Obstructing the ball**: Obstructing the ball involves 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 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 of their own penalty area. If a 4th 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 touch each other at the same time, one player of the team that has the majority of 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 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 which are involved in the self-collision will be frozen during *SelfCollisionJointFrozenTime* seconds. To freeze a joint means set its velocity to zero and ignore all commands sent by the agent to move this joint. After the frozen period, the joints cannot be frozen again within *SelfCollisionJointThawTime* seconds. Current self-collisions parameters used by the simulator are:

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

11. A **Pass command:** A Pass command is available to be used by agents during games to initiate a Pass mode for their team. To use this command, an agent should send a "(pass)" message to the simulator. 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 member of the team 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 member of the team 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 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*

12. 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.

13. **Penalty shoot-outs**: During a penalty shoot-out, each team only has a single player in the field: the goalkeeper for the defending team and a striker for the attacking team. Each penalty shot starts with a drop ball in the center of the field. The attacking team gets 40 seconds to attempt to score. The goalkeeper of the defending team must stay within his penalty area during the whole time. If neglected, it will be considered as a goal for the attacking team. When penalty shots are used to decide a tied game, each team gets five penalty shots. If after five 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. If after ten penalty shots no winner is determined, a coin-toss is used to decide.

14. **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.

15. **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 violating 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

Any of these is strictly forbidden. Other strategies might be found violating 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 not fair play. If you are in doubt of 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 laid out in this document. In situations that are 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. Decisions of the referee 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 on the internet 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 responsibility of the team 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 are required to 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 disqualification of the team from the current and/or future competitions.