



# Rescue Simulation Judge Training

Robocup 2026

# Important Competition Information



## Score(sheet)

- The scores are calculated automatically by the simulation supervisor. The scores will then be entered manually into the competition system.
- The **supervisor's** scoring of a run is **final**.
- If a team **disagrees** with the score, the team captain has to **write a comment in the designated part of the form** and **sign** it.
- The score will be reviewed **later**, and the team will be approached to present **video evidence** for their claim.

### Note:

The teams **are allowed** to film their runs, but video evidence will **only be accepted during the review**.

**No videos are to be discussed at the field.**

Videos will **only** be inspected if a **comment was written** on the **signed form**.

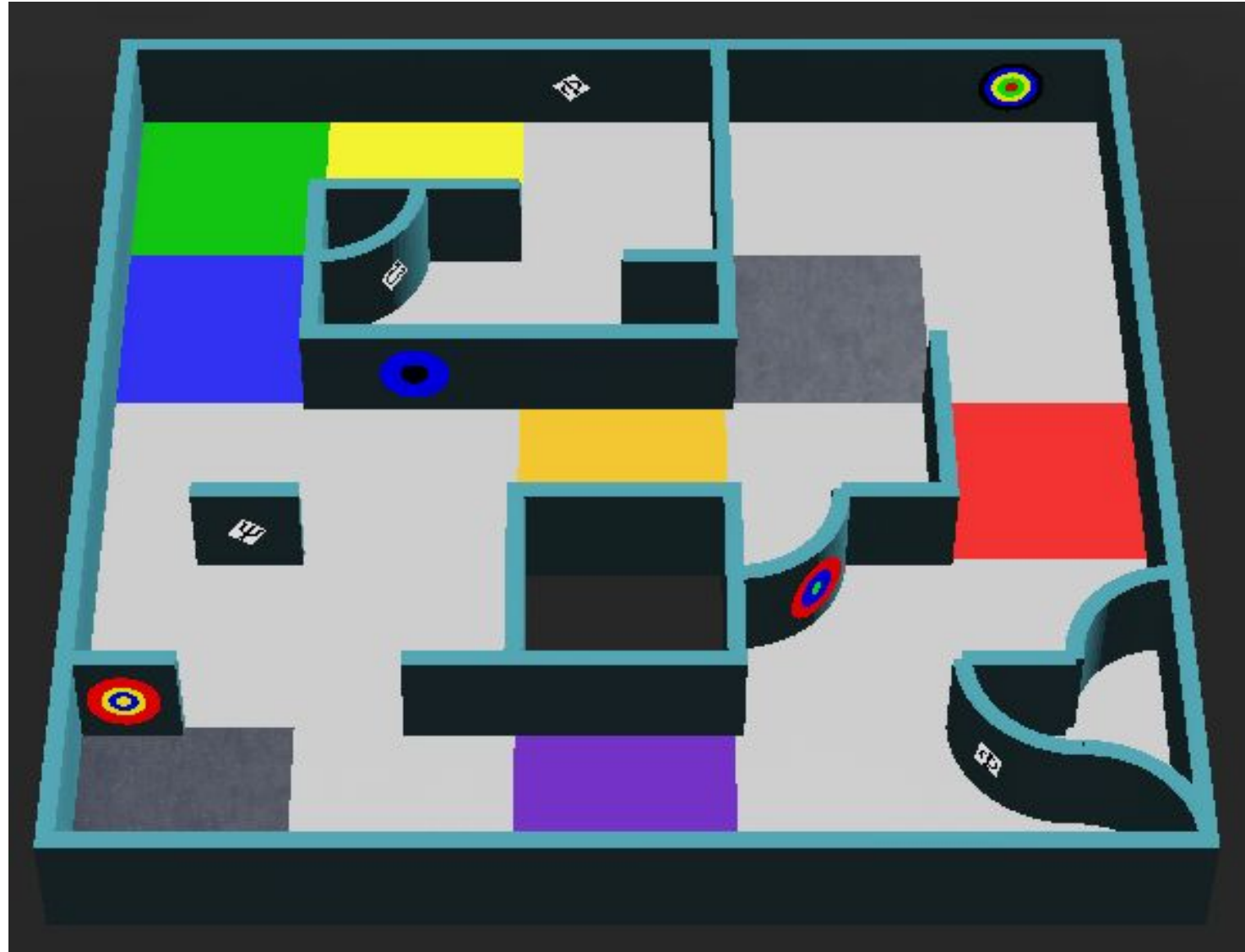
## Important Competition Information



# Competition time

- Teams are responsible for managing their own competition schedules, including all assigned activities such as rounds, interviews, poster sessions, and announcements.
- A team's time slot begins exactly at the scheduled time, regardless of whether the team is present. If a team arrives late, the lost time is not compensated, and the remaining time should be reduced accordingly.
- Judges should enforce this consistently to ensure fairness across all teams.

# Simulation Field and Controller



# Simulation Field and Controller



The image displays the interface of the Erebus Rescue Simulator 2026. On the left is a control panel with the following elements:

- Logo: RoboCup Junior Rescue
- Title: Erebus Rescue Simulator 2026, Ver. 26.0.1
- Buttons: LOAD (with code icon), TOGGLE REMOTE (with question mark), and another LOAD (with server icon).
- Input: Docker path: [text box]
- Navigation: Settings, Worlds, Reset, Give up!
- Control: Play, Pause, and Refresh buttons.
- Timer: 08:00 (10:00)
- Score: 0

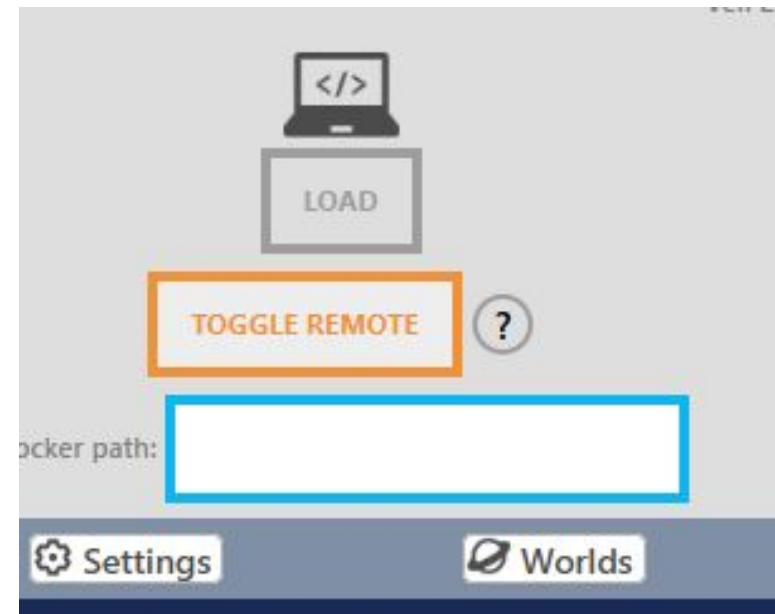
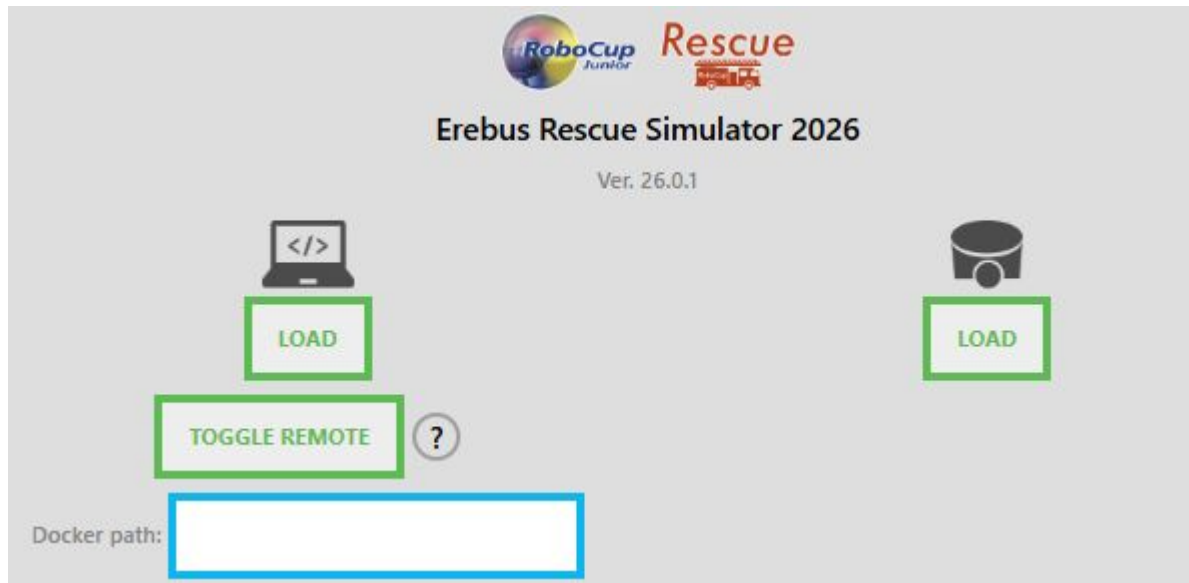
On the right is the Simulation View, which shows a 3D perspective of a maze-like environment. The environment features various colored platforms (green, yellow, blue, red, purple) and obstacles. A toolbar at the top of the simulation view includes icons for file operations and playback. Two red boxes highlight the 'Open World file' and 'Reload World file' icons, with red arrows pointing to them from labels. A console window at the bottom of the simulation view displays the following log message:

```
INFO: MainSupervisor: Starting controller: python.exe -u MainSupervisor.py
```

# Remote Setup

- Click on “Load” beneath the robot symbol, if the team is using a custom robot controller
- Custom robot controllers will be uploaded to the CMS before the rounds

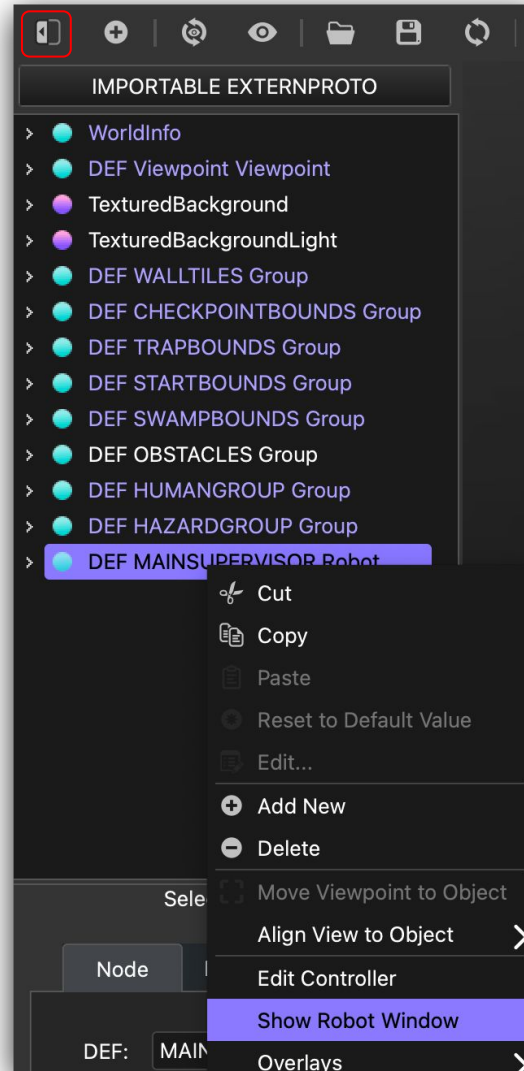
- Click on “Toggle Remote” to enable the remote controller



# Show missing Robot Window



In case the robot window is not showing





## Handling the run in general

- There is only one team member (captain) allowed at the competition field
  - All other members have to watch from the grandstand
- All actions on the game server will be performed by the judge
  - Loading the Maps
  - Starting the run
  - Pausing the simulation
  - Executing a LoP
  - etc.
- In case of any unexpected events, write a comment when putting in the scores into the CMS
  - This goes for judges as well as team captains

# Handling during a run



The screenshot shows the simulation control panel with the following elements and annotations:

- Settings**, **Worlds**, **Reset**, and **Give up!** buttons in the top bar.
- Pause** (green circle with 'II') and **LoP** (green circle with a refresh icon) buttons in the middle bar.
- Simulation runtime**: 06:55 (highlighted with a red box).
- Real life runtime**: (08:53) (highlighted with a red box).
- Score**: 20 (highlighted with a red box).
- Event Log**:
  - 00:56 Found checkpoint +20
  - 00:44 Lack of Progress (via robot) -5

Red arrows point from text labels to the corresponding UI elements: 'Pause' to the pause button, 'Simulation runtime' to the 06:55 timer, 'Real life runtime' to the (08:53) timer, 'End the round' to the Give up! button, and 'LoP' to the LoP button.

## After the run

- The round ends when the team captain calls the game end
- The robot sends the exit message on its own
- **Write down the duration and score**

Settings Worlds Reset Give up!

06:30 (08:27)

32.5

01:30	Manual give up!
01:27	Found checkpoint +12.5
00:56	Exited swamp
00:56	Entered swamp
00:23	Found checkpoint +20



## Execute the remote controller

- (Client) Controller will be executed on the command line of the participant's computer.

### Windows example

- (Client) `<Webots Controller Path>/webots-controller.exe --protocol=tcp --ip-address=<Server IP Address> <Controller Path>`
  - The standard installation path is ***C:\Program Files\Webots***
    - ***C:\Program Files\Webots\msys64\mingw64\bin***
  - The IP address will be individual for each competition computer - is written on the field sign
  - The controller path is the path to the participants' controller file
- After executing the command, the client should output an info message every 5 seconds:  
***No robot name provided, exactly one robot should be set with an <extern> controller in the Webots simulation, retrying for another 50 seconds...***
- (Server) Press the start button to start the simulation

### Linux and macOS

- On Linux and macOS you also have to set the WEBOTS\_HOME environment variable which should point to the installation folder:  
Client) `$WEBOTS_HOME/webots-controller --protocol=tcp --ip-address=<Server IP Address> <Controller Path>`