RoboCup Junior Simulation Competition Guide

1. Introduction

In 2020 we introduced a new Demo competition named New Simulation Demonstration, which was a new exciting way for students to be part of the RoboCup, without having to spend money on expensive hardware. Due to the success and positive feedback that we received, we decided to implement the League as the official RoboCup Junior Simulation League.
In 2021 we held the competition as well as the selection event all by ourselves. As the League will grow over time, the RoboCup Junior Rescue Committee won't be able to handle this task alone. For this reason, we would like the regional leagues to help us with the selection for the world championship.
This guide is supposed to give you a rough idea of how a competition for RoboCup Junior Simulation can be held.
For information about how the Erebus simulation platform works and how to install the software, please refer to the official documentation and GitHub:
https://erebus.rcj.cloud/docs/introduction/
https://gitlab.com/rcj-rescue-tc/erebus/erebus

The official rules can be found here:
https://cdn.robocup.org/junior/wp/2021/03/2021_RescueNewSimulationDemo_Rules_draft01.pdf

If there are any questions regarding this guide, please ask them in this thread in the RoboCup Junior Forum:
https://junior.forum.robocup.org/t/questions-about-the-robocup-junior-simulation-competition-guide/2443

2. Competition format

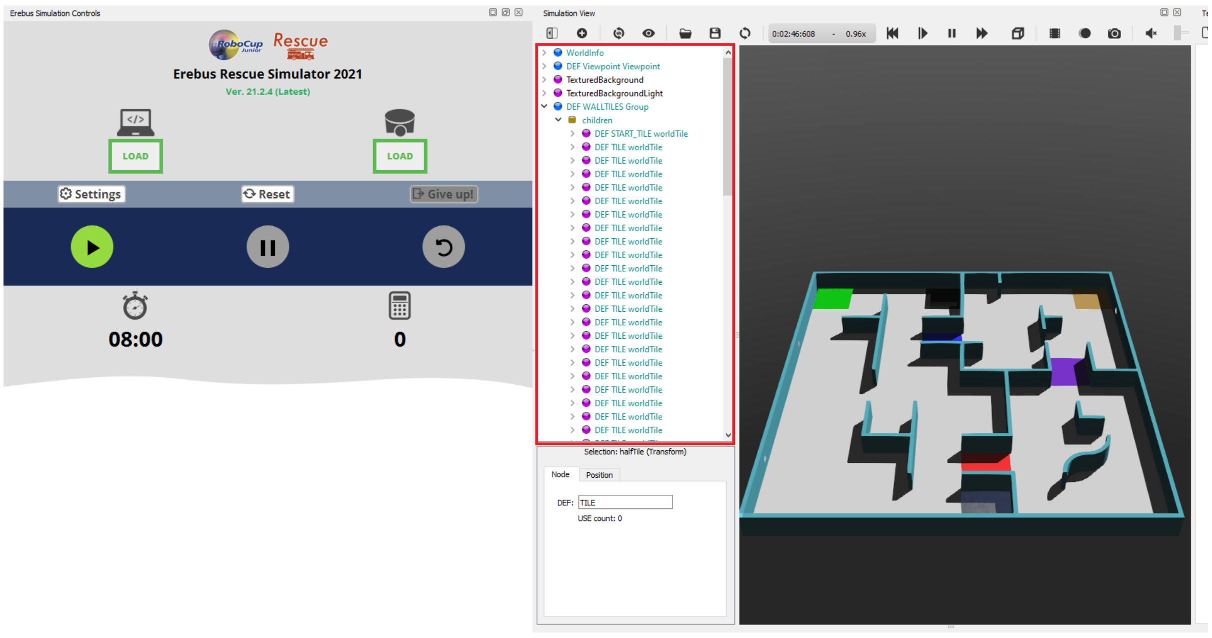
The competition format of the Simulation League can be chosen pretty freely. As we are not bound to physical hardware, the competition itself doesn't have to be held onsite. Meaning that a fully virtual or hybrid event is also manageable. In terms of learning value and experience for the students, an onsite competition is advised nonetheless.
The format of the runs itself can be organized similarly to the way that Line and Maze are handled currently. We suggest a set of multiple runs, spread out over one or more days, so teams have time to improve on prior experiences and new challenges.

3. Running the competition

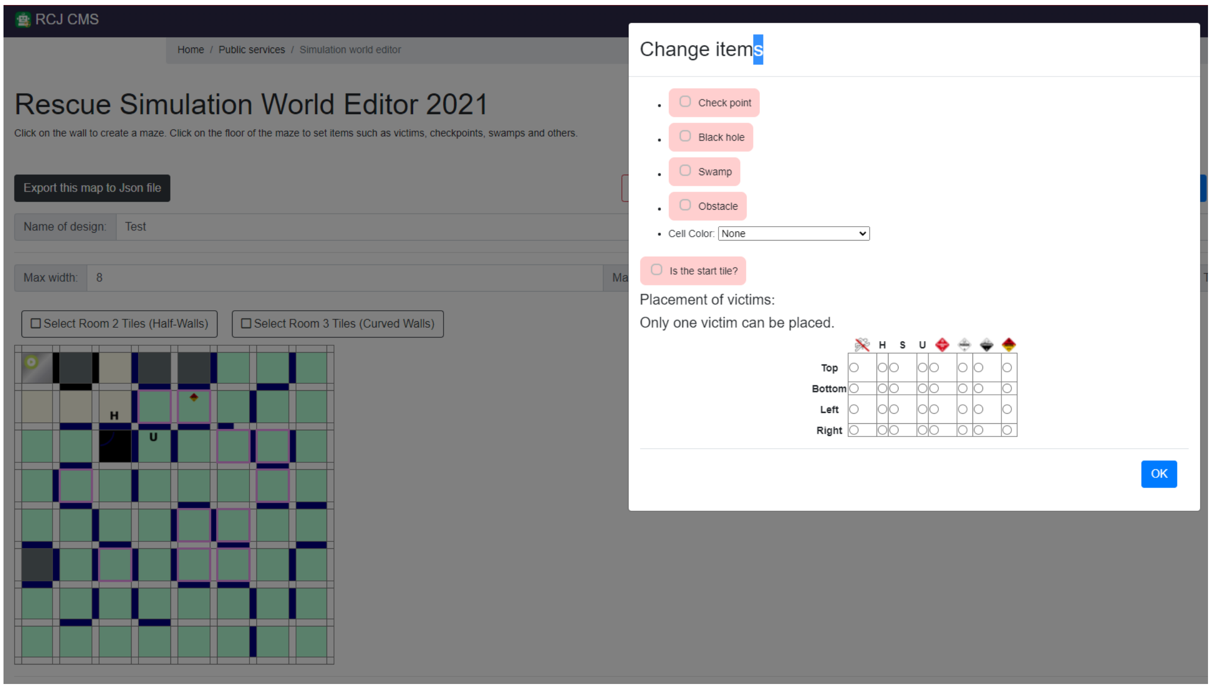
The Erebus simulation works with client-side execution of the code. As described in the rules, the simulation has to be run by the referee. Resulting from this, the organizer/ referee needs a working and stable Erebus platform to run the competitors code. Depending on the code, the simulation can be quite resource heavy as well, so a machine with reasonably high specs is needed.
Recommended Specs:
OS: Windows 10, 8.1 64bit / macOS 10.14+

CPU: 4 @2GHz
RAM: 8 GB
GPU: Nvidia GTX1050 or higher

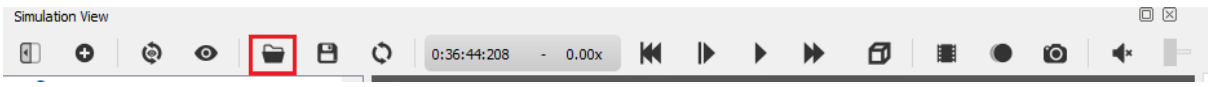
Before you can run the code, you need a map to run on. For initial tests, there are a few example maps in the /game/worlds directory of the Erebus folder. You can edit the walls and change the properties of the tiles inside the Webots attributes editor:



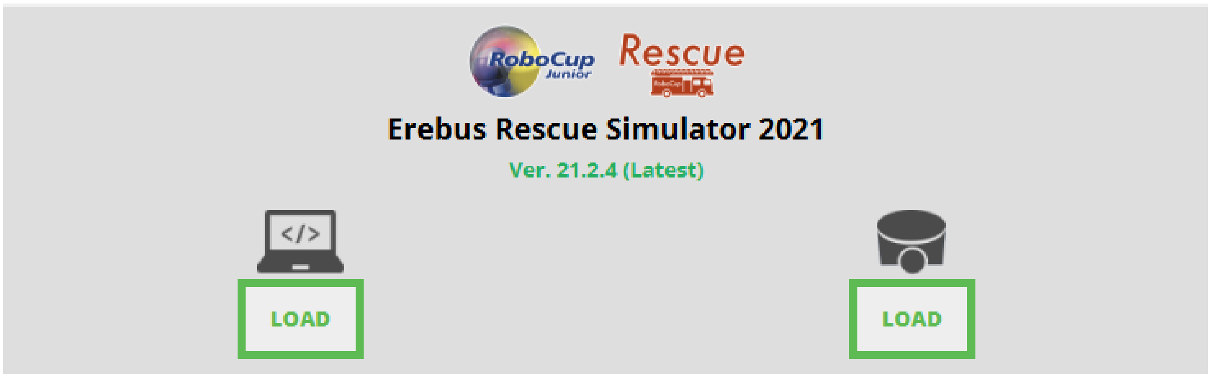
If you want to create a completely new map, then you can use the online map generator: https://osaka.rcj.cloud/service/editor/simulation/2021
The generator provides a simple GUI to generate new maps with a set size, walls, victims etc.:


RCJ CMS
Home / Public services / Simulation world editor
Rescue Simulation World Editor 2021
Click on the wall to create a maze. Click on the floor of the maze to set items such as victims, checkpoints, swamps and others.
Export this map to Json file
Name of design: Test
Max width: 8
Select Room 2 Tiles (Half-Walls)
Select Room 3 Tiles (Curved Walls)
Change items
Check point
Black hole
Swamp
Obstacle
Cell Color: None
Is the start tile?
Placement of victims:
Only one victim can be placed.
H
S
U
Top
Bottom
Left
Right
OK

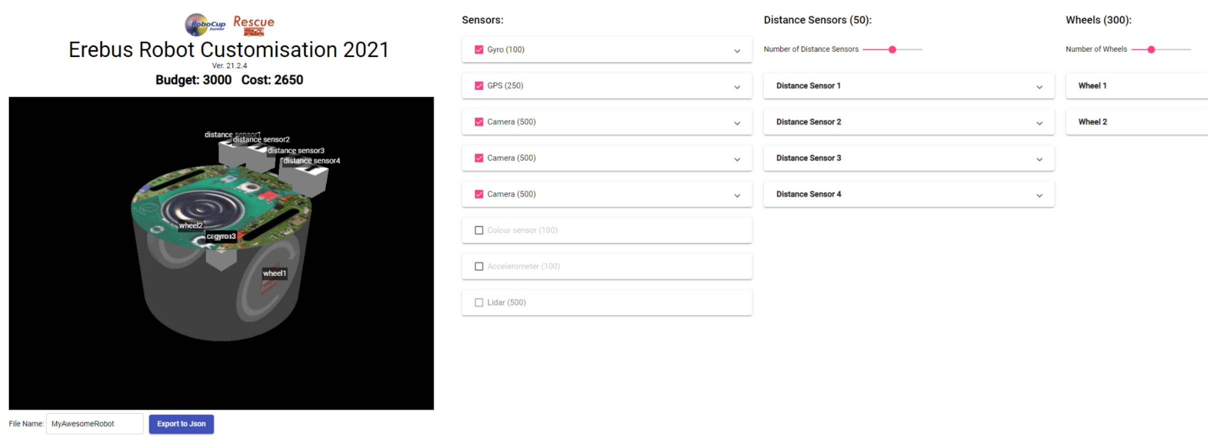
Afterwards, the map can be exported and moved into the /game/worlds folder. To open the map in a new Webots instance, just double-click the file. You can also load a new map with the small Folder-Icon in the menu bar.



The next step is to load the code of the team. The code has to be moved into the player_controllers folder inside the Erebus directory. You can find some example code in the folder as well. To load the code, just click on the "Load" button underneath the Laptop-Icon. To load a custom robot file, use the "Load" button on the right:



The custom robot generator can be accessed here:
https://robot.erebus.rcj.cloud/
The generator is based on a budget system, which means that the robot can only use a limited number of parts, where each part is valued differently:



Erebus Robot Customisation 2021
Ver. 21.2.4
Budget: 3000 Cost: 2650

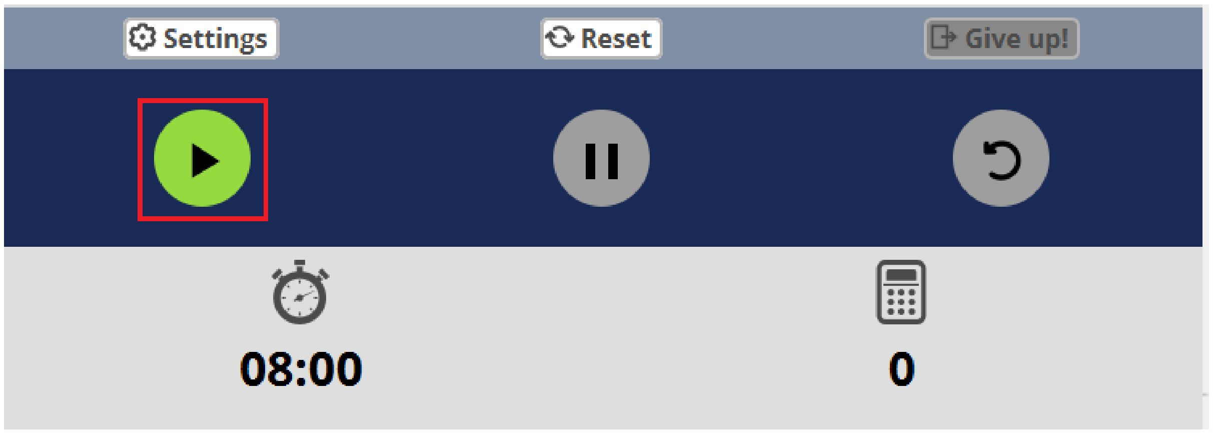
File Name: MyAwesomeRobot
Export to Json

Sensors:
- [✓] Gyro (100)
- [✓] GPS (250)
- [✓] Camera (500)
- [✓] Camera (500)
- [✓] Camera (500)
- [ ] Colour sensor (100)
- [ ] Accelerometer (100)
- [ ] Lidar (500)

Distance Sensors (50):
Number of Distance Sensors
- Distance Sensor 1
- Distance Sensor 2
- Distance Sensor 3
- Distance Sensor 4

Wheels (300):
Number of Wheels
- Wheel 1
- Wheel 2

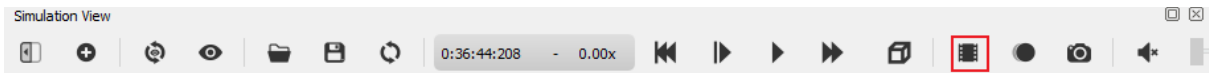
After everything is set up, the simulation can be started:



The program will automatically run the simulation and score the run as well. No further input from the team or referee is needed.

4. Showing the Runs

Just running the code and telling everyone the score afterwards wouldn't be very exciting. To make the simulation accessible to watch for teams and other competitors, we have two possibilities. Erebus has a built-in recording function that we can use to record the simulation with.



The recording will be saved as a standard .mp4 file and can be played back later, maybe with some commentary.

The second possibility is to use a separate recording program and live-stream the simulations while they're run. This option can be a lot more exciting if it's an onsite event. Be aware that running the simulation and streaming it at the same time will demand a lot more resources and could be hard to run on a laptop.