



# Task Explanation & Setup

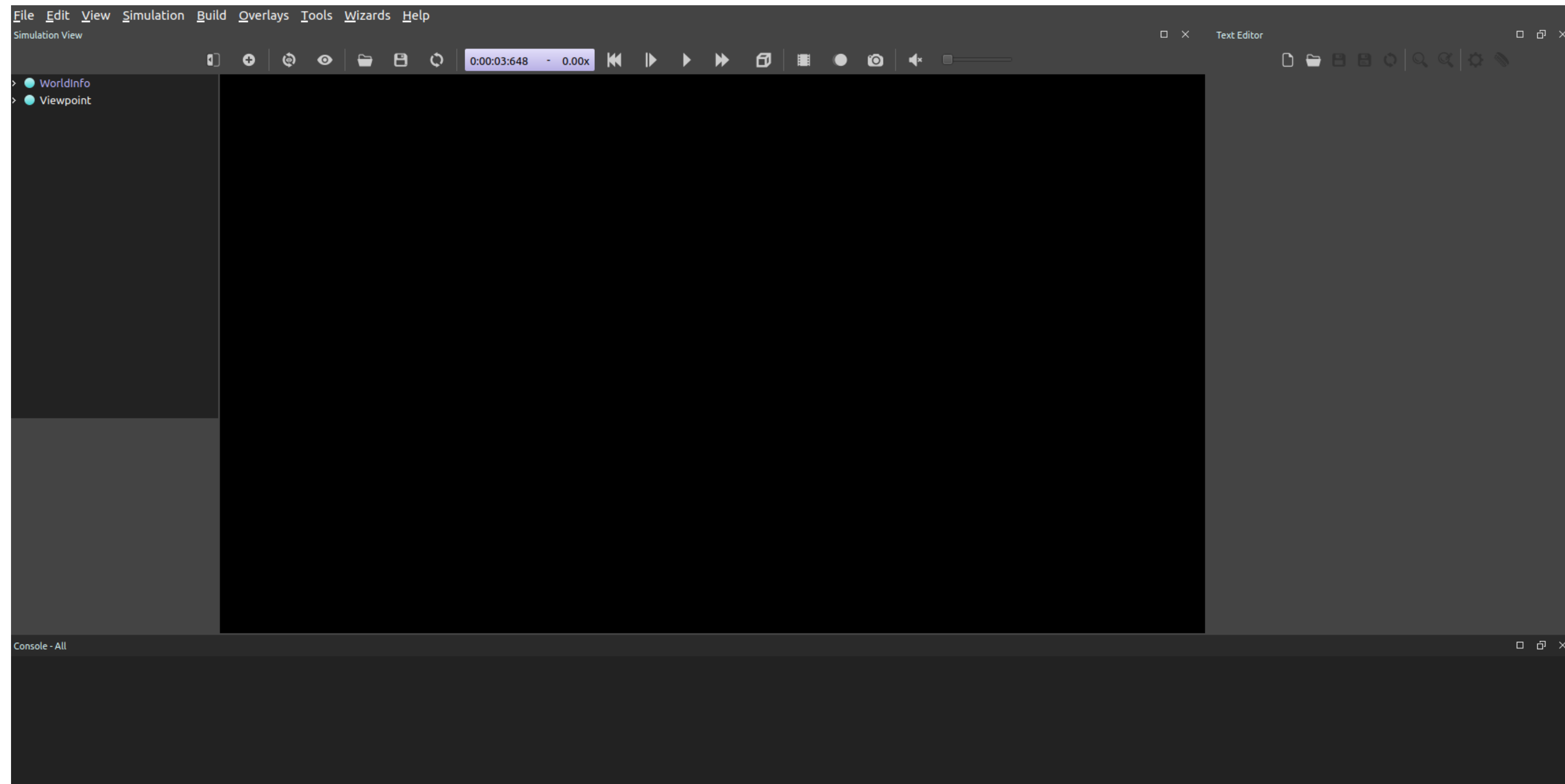
**Together, Stronger!**

# Contents

---

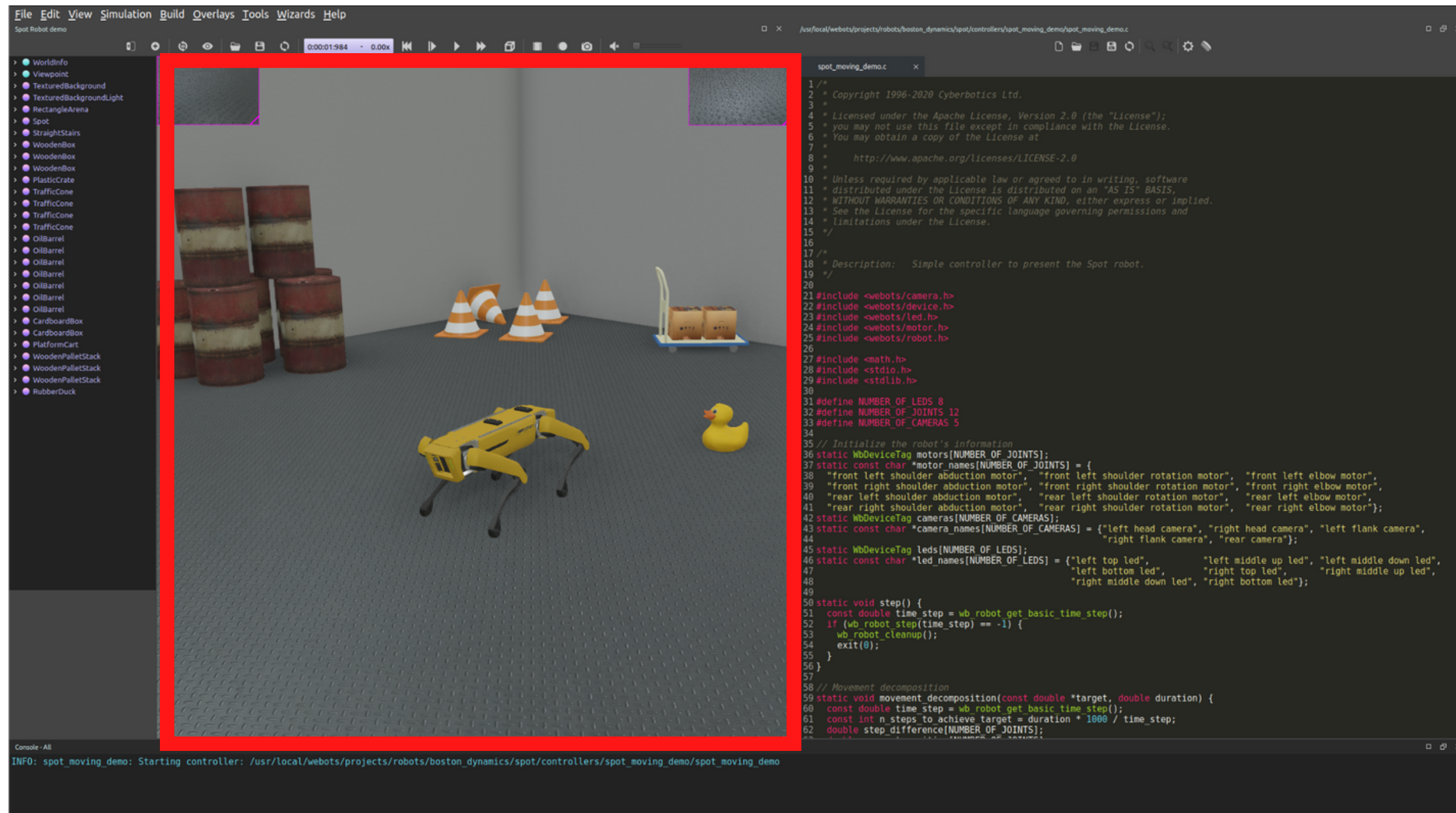
1. Introduction to Webots
2. Task explanation
3. Webots live demo
4. Guidelines for submitting the results

# Webots



Together, Stronger!

# Webots World

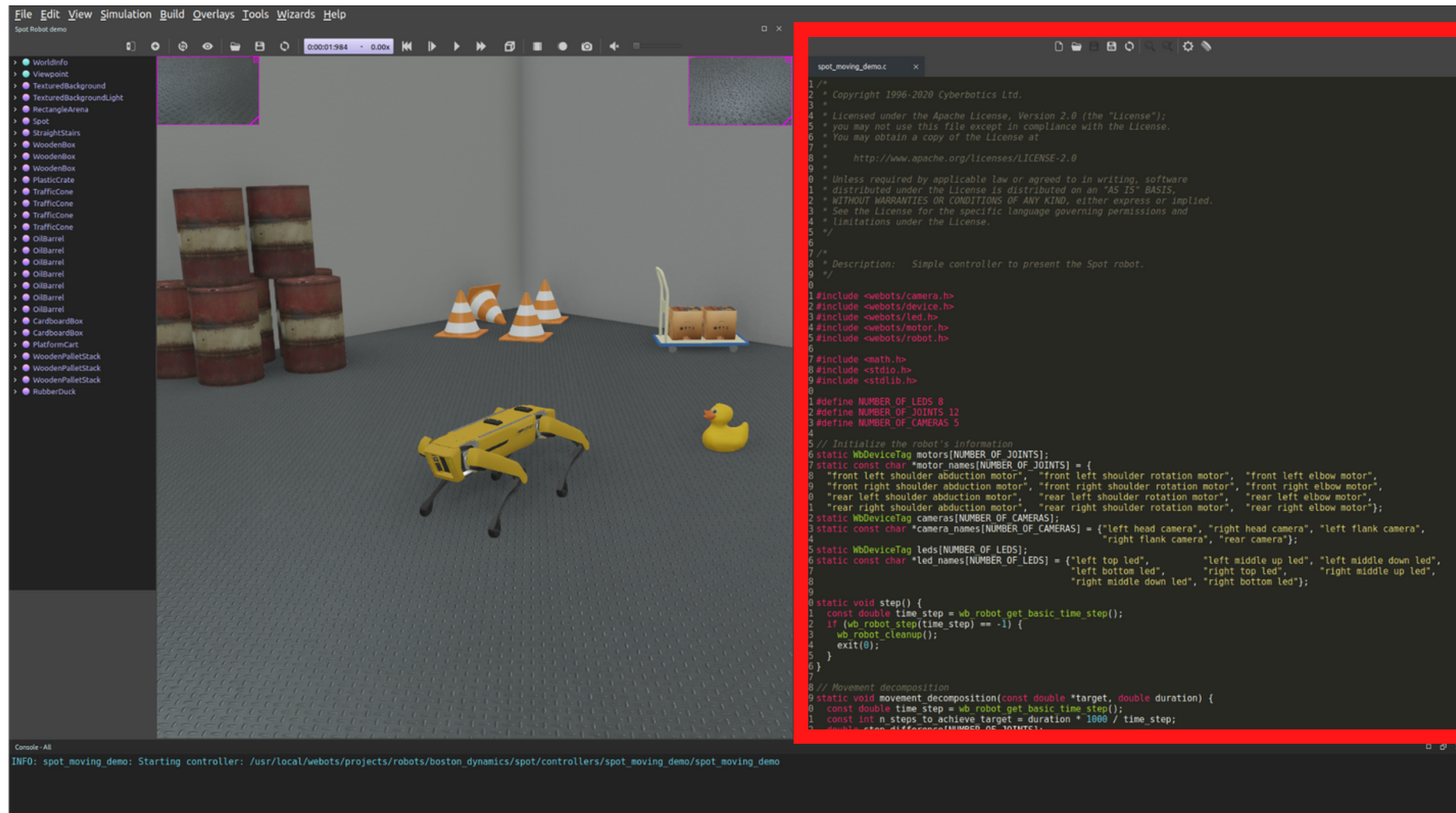


- To load the world file, go to **File > Open World...**
- The world files will have **.wbt** extension
- Upon selecting the desired **.wbt** file and clicking **Open**, a world will be displayed in this area

Together, Stronger!



# Webots Controller

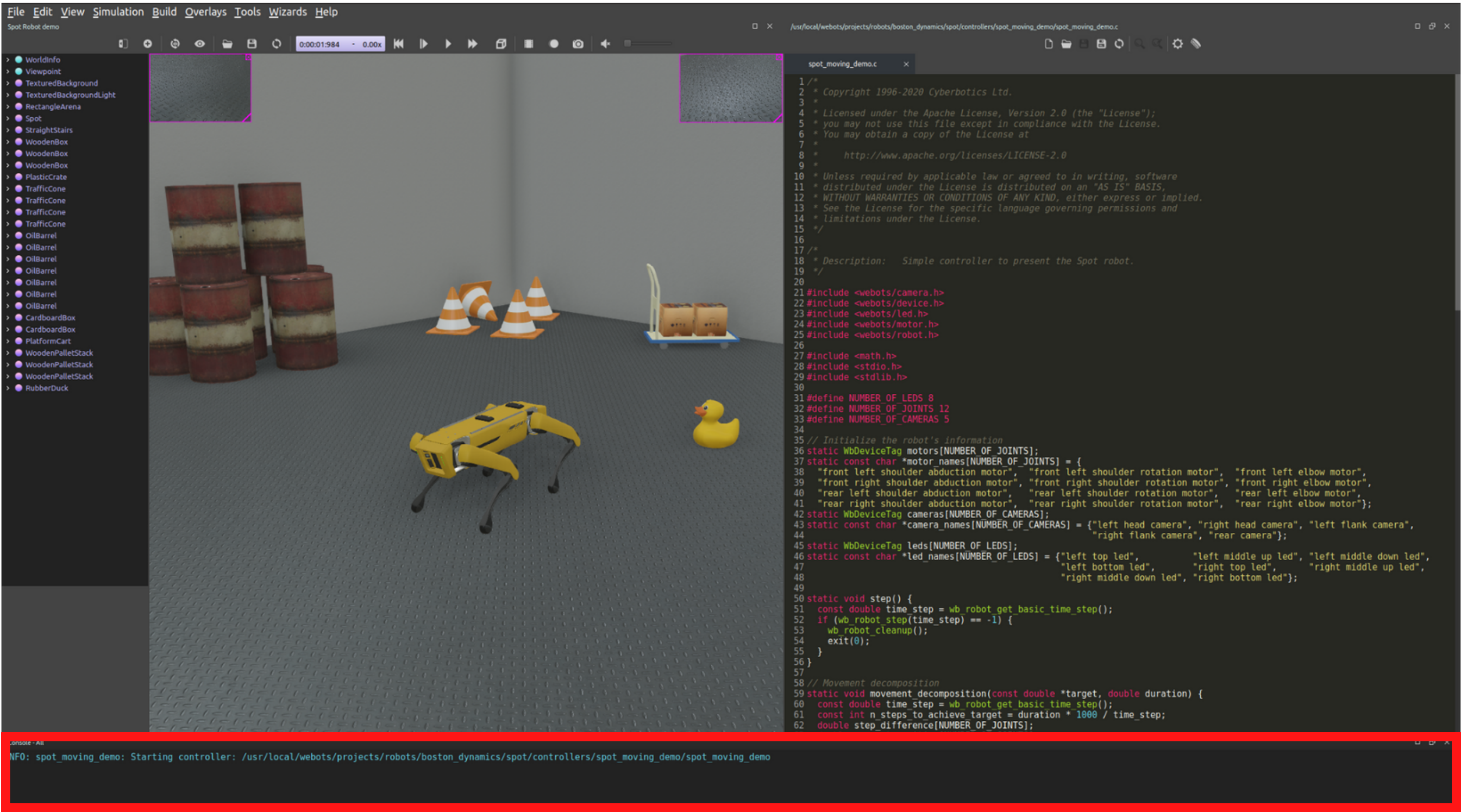


- The controller is responsible for producing desired behavior in the robot.
- Webots controllers can be written in C/C++, Java, Python or MATLAB.
- For the Hackathon we use controllers written in C++.

Together, Stronger!



# Webots Console



Together, Stronger!



# Task Specifications

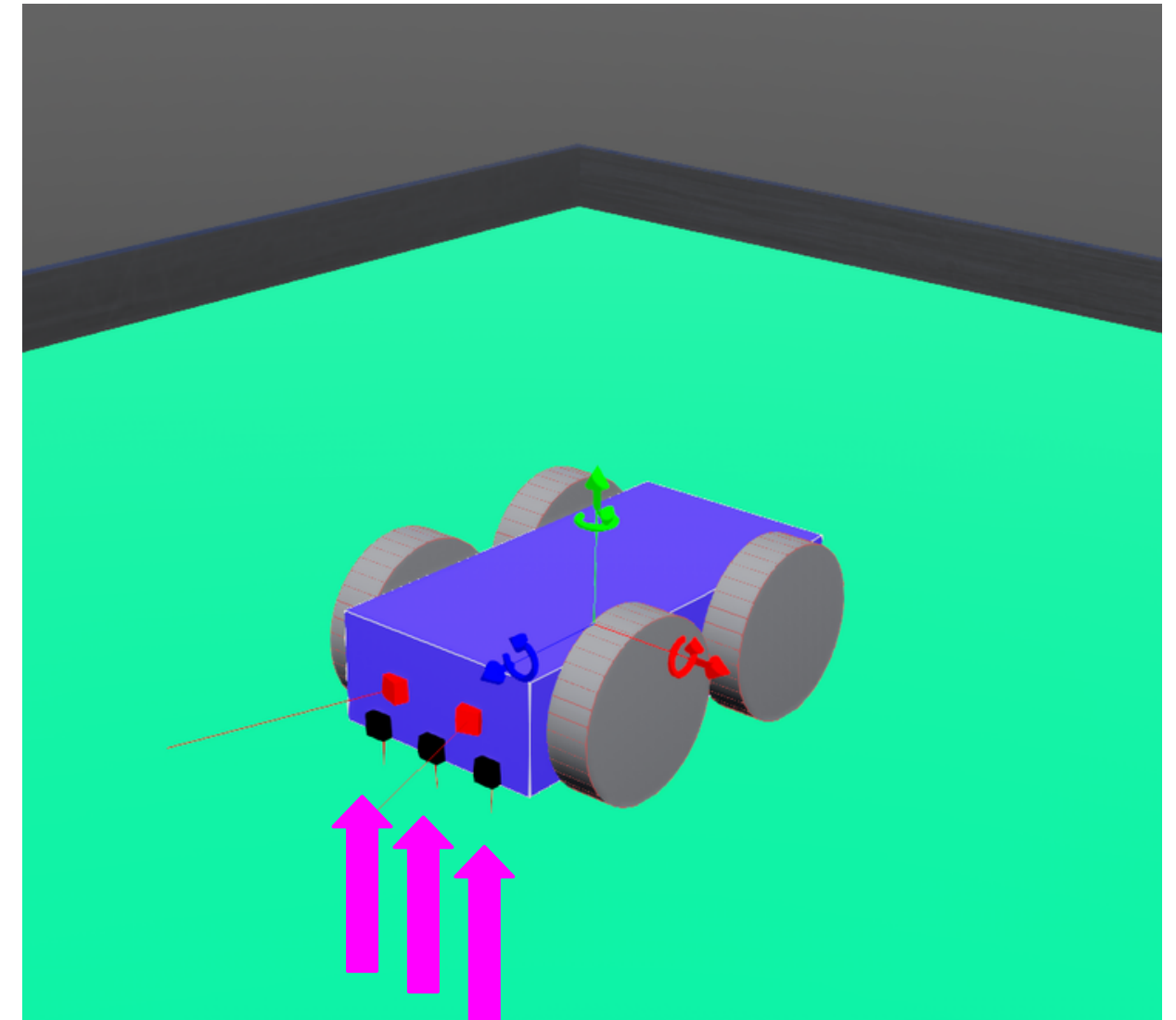
**Together, Stronger!**

# The Task

---

The robot for the task is equipped with:

1. Three Infrared Sensors





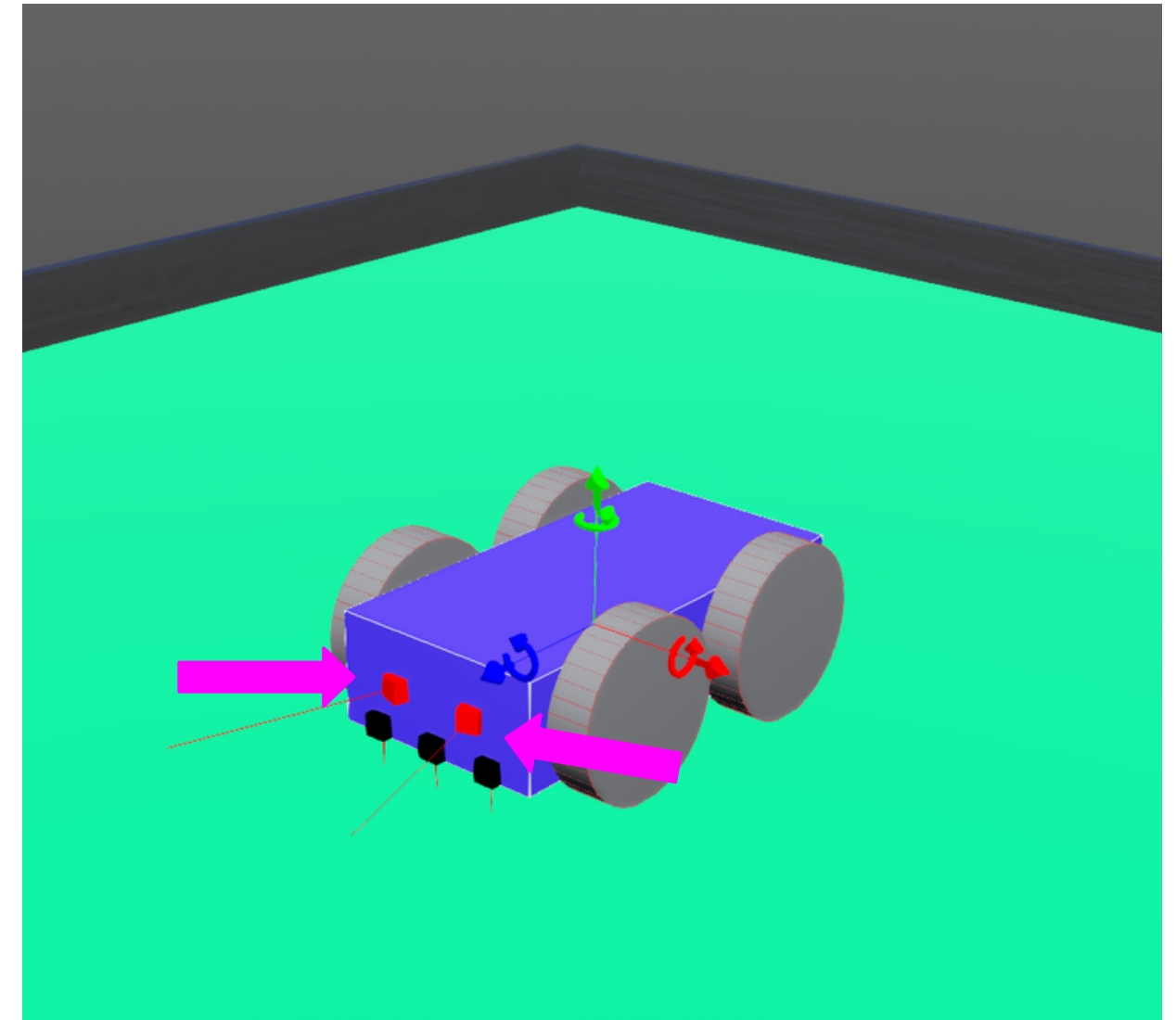
# The Task

---

The robot for the task is equipped with:

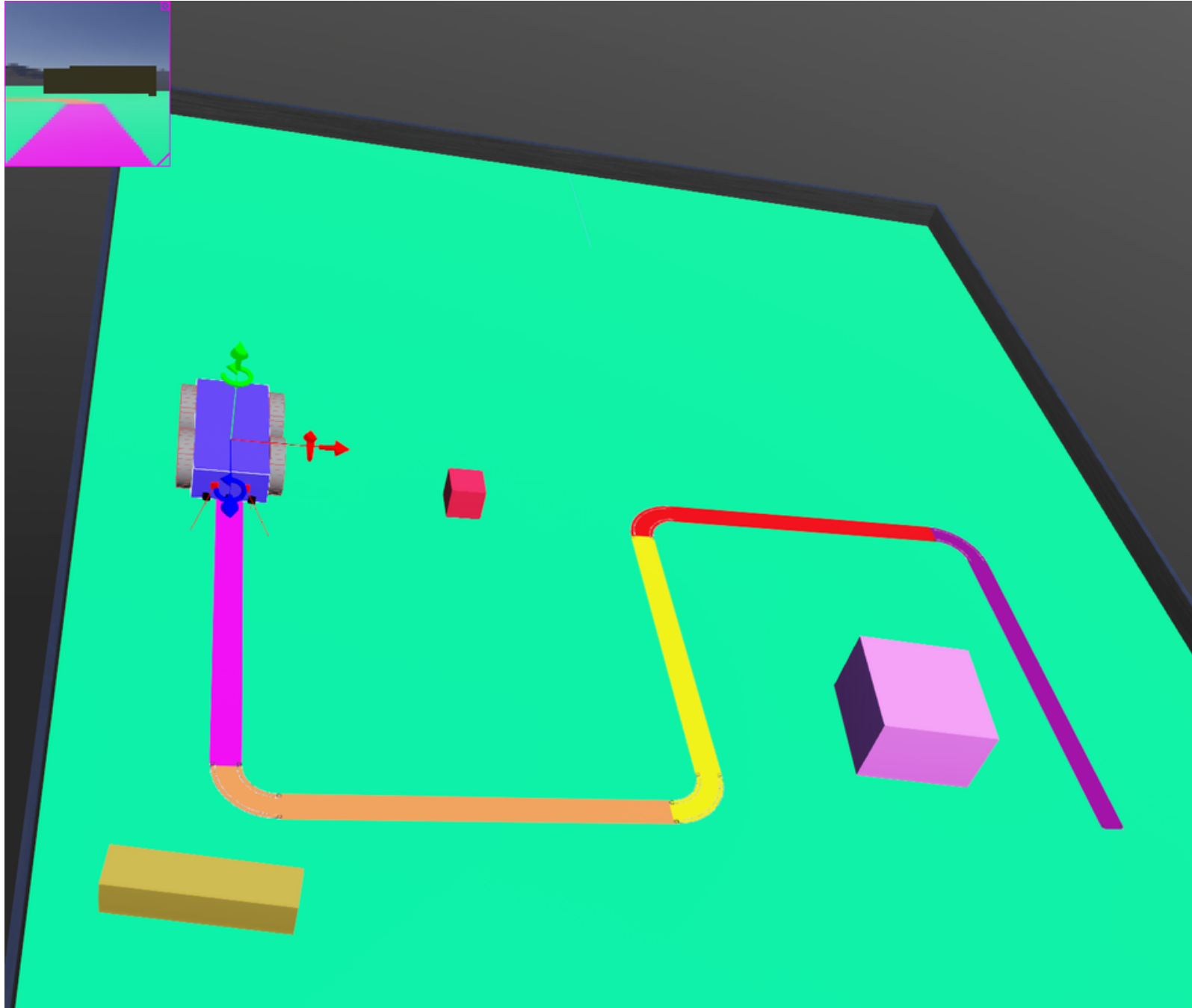
1. Three Infrared Sensors
2. Two Distance Sensors

**Note:** All the concepts related to driving the robot and reading the sensors will be explained in the Tutorial!



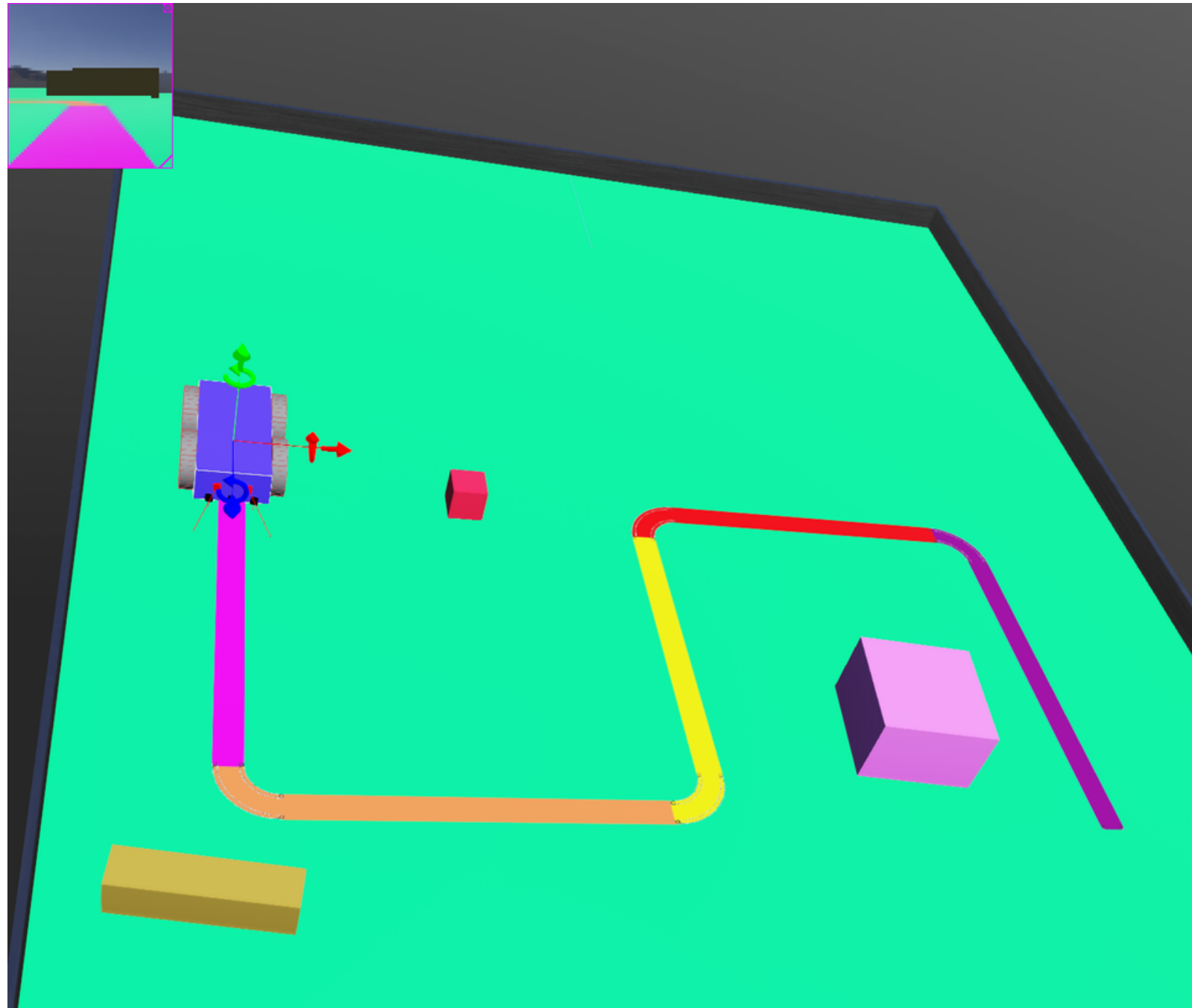
# The Task

---



Together, Stronger!

# The Task



To write a controller for the robot in the Webots simulator such that it follows the track from start to finish and avoids collisions with any obstacles on the track.

# The Task

---

This task is divided into **two** parts.

## Part 1: Line Following

- Develop the logic for turning left or right on the given track.
- Stop the robot when it reaches the end of the track.

## Part 2: Obstacle detection

- Stop the robot when there is an obstacle on the track within a distance of 4 cm from the robot.



# Task File Structure

---

You will use the following files for the task:

## Part 1: Line Following

- The world file: worlds > [hackathon\\_linefollow.wbt](#)

## Part 2: Obstacle detection

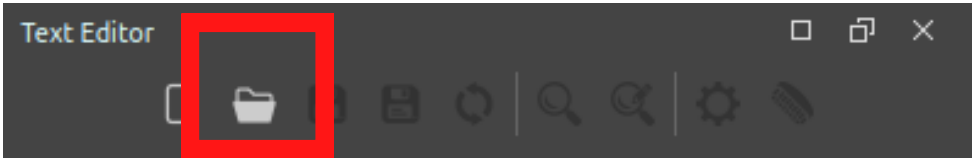

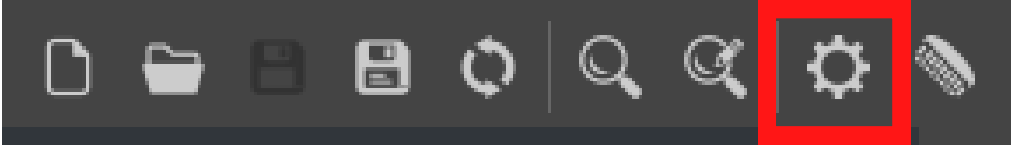
- The world file: worlds > [hackathon\\_obstacle\\_detection.wbt](#)

**Note 1:** The controller file for both worlds is the same, i.e. [linefollow\\_with\\_obstacle\\_detection.cpp](#)

**Note 2:** The drive link to the task files will be posted on Discord.

**!** **Important:** Please refer to the instructions on page 2-3 to start with Webots in Webots.

# If the robot doesn't move.....

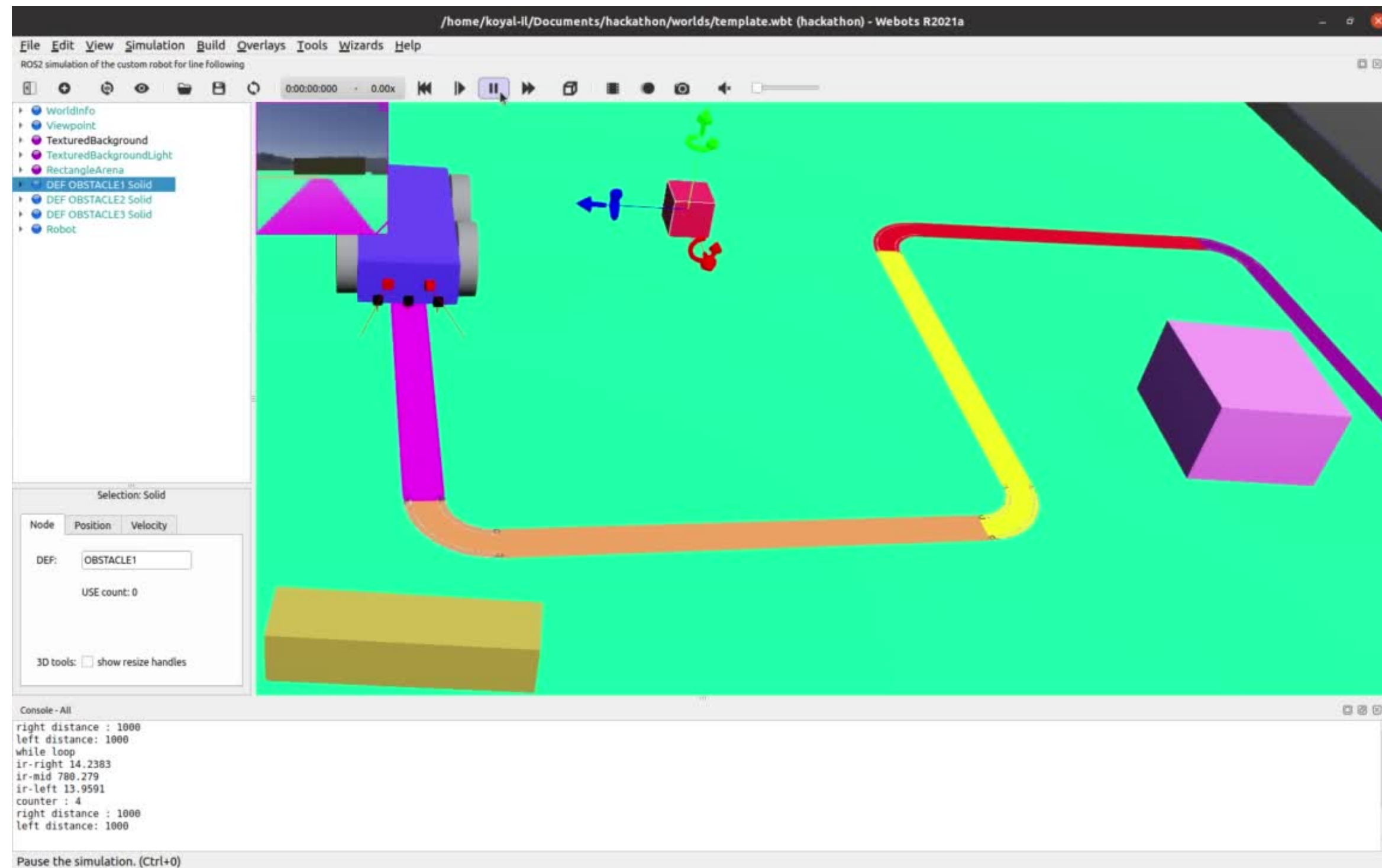
- Generally after loading the world, you can see the **robot** in the **middle window** as well as the **controller** in the **right window**.
- Sometimes it might happen that you can see the robot but not the controller code and there is a warning in the console : `WARNING: linefollow_with_obstacle_detection: This is not an executable file, try to change its permissions.`
- In this case,
  - You need to first open the controller file by clicking on  (The folder icon in the Text Editor toolbar is highlighted with a red box.)
  - Go to **controllers > linefollow\_with\_obstacle\_detection > linefollow\_with\_obstacle\_detection.cpp**
  - Click on the 'comb' symbol  (The comb icon in the toolbar is highlighted with a red box.)
  - Click on the 'wheel' symbol  (The wheel icon in the toolbar is highlighted with a red box.)
  - Click on 'Reset'
- The robot should now start moving!

# Controller Template

---

Together, Stronger!

# Expected robot behavior



Together, Stronger!



# Results Submission

---

- **Deadline** – 12:00 pm CEST, 26th June
- Late submissions from the teams after the deadline will **NOT** be considered!
- Each team will be given a **drive link** to upload the results.

## What should be uploaded?

- Controller code (C++ file)
- Screen recording of the robot performing the given tasks

## What if you finish this task early (way before the submission deadline)?

- You can also submit the results anytime before the deadline.
- Teams that do this, will be given a bonus task to solve!
- Completing the bonus task will fetch you extra points!

# Results Presentation

---

- Teams need to make a **5-minute presentation** describing their approach to solving the task (not the bonus task).

## General guidelines for the presentation

- Keep the text in the slides as less as possible
- Screenshots/screen recordings/animations can be included to make it more interactive

# Note

---

- The task description (filename 'Main task description.pdf'), as well as this presentation, will be shared with interested participants on Discord for reference.

**Questions?**  
**Thank you for your attention!**