HSB
Hochschule Bremen
City University of Applied Sciences

Hochschule Bremen
Fakultät für Elektrotechnik und Informatik

RoboCup Fußball Simulation 2D

Guide

Friederike Korte 5129793
Denis Karlinski 5155538
Tristan Lilienthal 5058556
Nour Ahmed 5200991
Serkay-Günay Celik 5031519

Dozent
Prof. Dr. Förster

16. Dezember 2024

Inhaltsverzeichnis

Introduction

The RoboCup Soccer Simulation provides a platform for testing and developing AI agents that compete in a 2D soccer environment. This guide walks you through setting up the required tools, libraries, and teams to simulate matches between two engines. By following these steps, you will install essential software such as the RCSS server, monitor, and libraries like `librcsc`. Additionally, you will set up two competitive teams: Helios Base and Cyrus Base, which serve as AI agents in the simulation. Whether you are a researcher, student, or AI enthusiast, this guide will help you establish a fully functional environment to run RoboCup Soccer matches.

1 System Preparation

1. Open the Microsoft Store.
2. Search for **Ubuntu 20.04.06 LTS**, download, and open it.
3. Enter a **username** and **password** when prompted.
4. Update your system:

```
sudo apt update
sudo apt upgrade
```

5. Create a project folder structure:

```
mkdir rc
cd rc
mkdir monitor server teams tools
```

2 RCSS Server Setup

1. Install dependencies:

```
sudo apt install build-essential automake autoconf libtool flex
   bison libboost-all-dev
```

2. Download the server:
   (a) Visit `https://github.com/rcsoccersim/rcssserver/releases/tag/rcssserver-18.1.3`.
   (b) Download `rcssserver-18.1.3.tar.gz`.
3. Move the file to the `server` folder and extract it:

```
tar xvzf rcssserver-18.1.3.tar.gz
cd rcssserver-18.1.3
./configure
make
```

3 RCSS Monitor Setup

1. Install dependencies:

```
sudo apt install build-essential qt5-default libfontconfig1-dev
   libaudio-dev libxt-dev libglib2.0-dev libxi-dev libxrender-dev
```

2. Download the monitor:

   (a) Visit `https://github.com/rcsoccersim/rcssmonitor/releases/tag/rcssmonitor-18.0.0`.

   (b) Download `rcssmonitor-18.0.0.tar.gz`.

3. Move the file to the `monitor` folder and extract it:

```
tar xvzf rcssmonitor-18.0.0.tar.gz
cd rcssmonitor-18.0.0
./configure
make
sudo make install
```

4 Librcsc Setup

1. Install dependencies:

```
sudo apt install build-essential libboost-all-dev autoconf automake
   libtool
```

2. Download and install `librcsc` into the `tools` folder:

```
tar xvzf librcsc-<version>.tar.gz
cd librcsc-<version>
./bootstrap
./configure
make
sudo make install
```

5 SoccerWindow2 (Better Visualization)

1. Install dependencies:

```
sudo apt install build-essential automake autoconf libtool libboost-
   all-dev qt5-default libfontconfig1-dev libaudio-dev libxt-dev
   libglib2.0-dev libxi-dev libxrender-dev
```

2. Download and install `soccerwindow2`:

```
tar xvzf soccerwindow2-support-v18.tar.gz
cd soccerwindow2-support-v18
./bootstrap
./configure
make
sudo make install
```

3. Update library paths:

```
sudo sh -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/librcsc.conf'
sudo ldconfig
```

6 Helios Base Setup (First Team)

1. Install dependencies:

```
sudo apt install build-essential libboost-all-dev
```

2. Download and install Helios Base into the `teams` folder:

```
tar xvzf helios-base-support-v18.tar.gz
cd helios-base-support-v18
./bootstrap
./configure
make
```

7 Cyrus Base Setup (Second Team)

1. Install dependencies:

```
sudo apt install build-essential libboost-all-dev cmake
```

2. Install required libraries:

   (a) Install `librcsc` into the `tools` folder:

```
git clone https://github.com/helios-base/librcsc.git
cd librcsc
mkdir build
cd build
cmake ..
make
sudo make install
```

   (b) Install Eigen3:

```
sudo apt install libeigen3-dev
```

   (c) Install `CppDNN` into the `tools` folder:

```
git clone https://github.com/Cyrus2D/CppDNN.git
cd CppDNN
mkdir build
cd build
cmake ..
make
sudo make install
```

3. Install Cyrus Base into the `teams` folder:

```
git clone https://github.com/Cyrus2D/Cyrus2DBase.git
cd Cyrus2DBase
mkdir build
cd build
cmake ..
make
```

8 Automating the Setup with a Bash Script

To simplify starting the server, both teams, and the monitor, you can create a bash script. Below is a sample script:

```
#!/bin/bash

# Start the RCSS Server
echo "Starting RCSS Server..."
cd /home/robo/rc/server/rcssserver-18.1.3/src
./rcssserver &

# Short pause to ensure the server is running
sleep 2

# Start the first team (Cyrus)
echo "Starting Team Cyrus..."
cd /home/robo/rc/teams/Cyrus2DBase/build/bin
./start.sh &

# Start the second team (Helios)
echo "Starting Team Helios..."
cd /home/robo/rc/teams/helios-base-support-v18/src
./start.sh &

# Start the Monitor
echo "Starting RCSS Monitor..."
cd /home/robo/rc/monitor/soccerwindow2-support-v18/src
./sswindow2 &
```

1. Create a new file in the `rc` folder:

   ```
   nano start_match.sh
   ```

2. Insert the above code and save the file.

3. Make the file executable:

   ```
   chmod +x start_match.sh
   ```

4. Run the script:

   ```
   ./start_match.sh
   ```

9 Fixing Bad Interpreter Errors

If you encounter the error `/bin/bash^M: bad interpreter: No such file or directory`, it means the script was created with Windows-style line endings (CRLF) instead of Unix-style (LF). This can be fixed using one of the following methods:

1. Install `dos2unix` and fix the script:

```
sudo apt install dos2unix
dos2unix start_match.sh
```

2. Alternatively, use `sed` to remove the carriage returns:

```
sed -i 's/\r$//' start_match.sh
```

3. Make the script executable again:

```
chmod +x start_match.sh
```

After applying one of the fixes, you can execute the script as usual:

```
./start_match.sh
```

10 Final Notes

Follow the steps carefully, and ensure all dependencies and libraries are correctly installed. Once set up, you can run matches between the two teams using the `rcssserver`, `rcssmonitor`, and the respective team binaries.