# Simulation Project RoboHub Eindhoven



## **Backstory:**

Since the RoboHub started there was always a team working on the robot for the RoboCup (Suii) physically. But because the team has grown and not everyone can be present where the robot is. Working on the software for Suii is getting harder since you can not test it on the complete system. Suii's capabilities are growing but for the growth to be faster we can not rely on only the physical robot. This is why we want a full simulation with all Suii's sensors, dynamic capabilities and future improvements.

#### What do we want:

For the developers to **work on Suii from home** or when Suii is under construction, we need a simulator that can behave like Suii would in the real world. All the necessary components that make Suii navigate and manipulate need simulators.

- Manipulation:
  - **3D Camera** that generates a depth map (Must)
  - 2D Camera (Must)
  - A moving UR3 arm (Must) (highly complicated, advised to just use the Universal Simulator)
  - A moving and "gripping" gripper (Should)
  - Collision detection (Should)
  - Autonomous object segmentation for future AI training (Could)
  - Complete dynamics simulation with gripping forces and weights (Could)

- Navigation:
  - Lidar data (Must)
  - The ability to stage a **RoboCup circuit** (Must)
  - **Odometry** from the wheels movement (Must) (highly complicated dynamic problems, could just use a mock simulator the returns the command velocities)
  - Collision detection (Should)
  - Complete dynamics simulation with drag, torque and speed simulation. (Could)
  - Ultrasonic data (Could) (For seeing glass windows)

## The end product:

- A working simulation that can handle the basics of Suii
- A **Tutorial** on how to build this kind of Simulation and use the software
- A small **research** on why there was chosen for Webots or Gazebo
- If you build any API that we can work with there needs to be API documentation
- The **documentation** has to be done in Markdown or Asciidoc so it can be publised on **GitHub**.
- A class and data flow diagram of the software.
- The end product has to be on GitHub where the actual **workflow** can be followed.
- A **demo** for the team when the project is finished.

## **Research before starting:**

There are a dozen 3D robot simulators on the market but there are two that are looking as a good way to go for our system and that are **Gazebo** and **Webots**, Gazebo is an open-source project focused on **ROS** robots and Webots is also just released as an open-source project but has been developed without **ROS** in mind, but there was done a lot of work paid by big companies so the product is pretty mature.

#### **Known hard parts:**

- For a simulator to work properly the 3D robot model has to be as close as possible to the real-world robot. Getting that proper can take up allot of the time needed to get this working.
- For **dynamic behavior** there are a lot of configurable variables that need to be set. This is tedious work, don't put to much time in it.
- The robot software (Navigation, Vision algorithms and state machines) is all running on the robots computer. It already has some computation power issues. Running that on a laptop where there is also a 3D Simulator with physics engine running can be a hard task. So "Less is more" when we are talking about necessary computational resource needed for the simulation.
- This is a **very software intensive project**, it needs to be run on a Linux system with **ROS** communication. Preferably written in python with the possibility to write computational heavy code in C++.

## Motivational pitch for the project:

**You will learn a lot** about the growing mainstream robotics software. Working on simulation is a big topic that **every engineering company is looking for**, as they call it their "**Digital Twin**". This is the perfect project to kick start your knowledge into the robotic software development. It also gets you in touch with the RoboHub Eindhoven Team where you are able to join the **RoboCup tournaments all over the world**.

# **Project requirements from Fontys:**

- Money: Expected 0 eu
- To get it working you will need 4 motivated people that want to learn Software. Or at least 2-3 experienced software people.
- It should be doable on the laptops of the students.