Simulation environment

When choosing the simulation environment for the project, we focused mainly on the return of investment in terms of learning the simulation environment. In addition to this it was important that the chosen environment had support for Arduino. A last factor was that we needed a software which was free, or atleast had free student licenses. In general, we decided that we wanted to use ROS (Robotic Operating System) as the robotic framework for our system. As a result, the simulation environment to be used had to support ROS. *Figure 1* shows a comparison chart of the environments discussed in the group.

	OS	Language	Tutorials	License	ROS connectivity	Arduino
V-REP	Windows, Linux, MacOS	C/C++, Python, Matlab	Yes	Commercial - free educational version	Plugin exists	Yes
Gazebo	Li nux	C/C++, Python	Yes	Open source	Maintained alongside ROS	Yes
Webots	Windows, Linux, MacOS	C/C++, Python, Matlab	Yes	Commercial - educational costs money	Has ROS interface	Yes
Player/Stage	**Player:** Windows, Linux, MacOS. **Stage:** Linux, MacOS	**Player:** Any that supports TCP sockets. **Stage:** C/C++, Python	Yes	Open source	Has ROS interface	Yes

Figure 1: Comparison chart for simulation environments

In the process of choosing the simulation environment, we mainly discussed V-Rep and Gazebo. Since we established the fact that we wanted to use ROS, the most natural choice for the environment became Gazebo, since this is maintained with ROS. We also considered the return of investment in terms of future usage. The two factors which kept us hesitant when deciding for ROS was that it is Linux based (none of the group members had computers set up with Linux), and the fact that we were informed by the teacher in the subject that the learning curve is steep. The challenge with no Linux-based computers was solved by installing Linux on two old computers.