Improving usability of a Web-based platform for teaching robotics engineering*

Lía García-Pérez[2][0000-0002-6519-9108], David Roldán[1][0000-0001-7049-7460], Enric Cervera[3][0000-0002-5386-8968], Pawan Wadhwani[4], and José M. Cañas[1][0000-0003-4179-2211]

[1] Universidad Rey Juan Carlos, Spain
[2] Universidad Complutense Madrid, Spain
[3] Universitat Jaume I, Spain
[4] SRM University, India

Abstract. Unibotics is an open online web framework for robotics engineering in higher education. It allows practical learning of robot programming from the web browser, and fits both distance and face to face education. It includes more than twenty robotics challenges illustrating mobile robotics concepts (navigation, control, localization, mapping, etc.), service robotics (drones, autonomous driving, vacuum cleaners, logistics), and DeepLearning (image classification, object detection). It is based on Python language, ROS middleware and Gazebo simulator. A new version has been released with internal refactoring and improved frontend, based on REACT library. A pilot study has been carried out with 110 real students from three different Spanish universities using it in three 12-week courses. The usability of the new Unibotics has been measured and compared to previous release. The questionnaire shows improvements in usability but still some inconsistencies in reliability. No scalability problems happened despite the growth in concurrent users. The platform has also allowed grad students with no prior robotics background to start with robotics engineering.

Keywords: Distance learning · Engineering education · Web-based and remote robotics · simulation.

1 Introduction

Robotics provides an increasing number of solutions and products for real-life problems, from autonomous driving to automatic cleaning, inspection and logistics. There is a growing industry and research behind. Several trends in robotics industry and research are also increasingly being introduced into the academic practice of robotics higher education.

First, in the last decade ROS middleware [6, 4] has become the de facto worldwide standard in the robot programming community. It proposes a distributed

* Supported by JdeRobot organization

component-based paradigm for robotics applications, they are composed of several ROS nodes which interoperate among them through typed messages called ROS topics. The ROS ecosystem includes a large community, several tools (Rviz, ROSbags, etc.), a collection of drivers and software pieces ready to reuse (Navigation stack, MoveIt stack for industrial robots, MAVROS for drones, etc...).

Second, the use of simulators has widely extended in the robotics research and industry. They allow to test robotics applications extensively before transferring them to physical robots. There are many general purpose robots simulators, several of them open source, such as Gazebo[3], Carla, Webots, CoppeliaSim, AirSim to name a few. For instance, they allow massive benchmarks and exhaustive testing of auto-pilot software in autonomous driving.

Third, there is also an increasing trend in the professional robotics world on using web technologies. One illustrative example is the RobotWebTools project [10], whose goal is to converge robot middleware with modern web and network technologies to enable a broadly accessible environment for robot front-end development, cloud robotics, and human interaction research. Foxglove Studio is an integrated visualization and diagnosis web tool for robotics, which has been used in the development and testing of autonomous racing technology. A few frameworks have recently appeared which support robot programming from the web browser and cloud computing [11]: IntrinsicAI Flowstate (from Google), Asimovo RoboDevOps or AWS RoboMaker [5] are good examples.

Those three trends in robotics research and industry are percolating through the academy, the robotics higher education. There are several private web online robotics training platforms; all of them use virtual laboratories. Robot Ignite Academy from The Construct is based on ROS, Gazebo and Jupyter, and provides several short courses. Riders.ai provides online robotics courses with real-world applications, including drones, and exciting competitions that students can participate in. Users only need a browser to access the integrated development environment where they edit code and run simulations. RobotBenchmark from Cyberbotics provides free access to a series of robotics exercises based on Webots simulations which are run in the cloud. Another noteworthy initiative is the Robot Programming Network (RPN) [2]. It extends existing remote robot laboratories with the flexibility and power of writing ROS code in a Web browser and running it on the remote robot on the server side with a single click.

Another web learning framework is Unibotics, which was born offline, then became ROS-based [8] and then online [7]. It currently includes more than twenty open source exercises (*https://jderobot.github.io/RoboticsAcademy/exercises/*) and several courses (for instance, for Intelligent Robotics and for Drones). This paper presents a pilot study with the latest release of Unibotics and 110 students from three different universities. The focus has been to test scalability with a larger user community, measure the impact on usability of the last refactoring, and to test its use from students with Computer Science profile without any prior background on robotics.

2 Software improvements in the Unibotics platform

In this section we summarize the last improvements , mainly focused on enhancing the maintainability and scalability of the platform by improving the user experience. The Unibotics design has been detailed in [7] and its architecture is shown in Figure 1. In runtime there are two main components: the web browser and a docker container. The browser acts as the graphic user interface (GUI) of the application, allowing the user to select one exercise in which he/she will have to solve one task by programming a robot. Once the user chooses one exercise, the GUI of that exercise is shown, as it can be seen in Figure 3. This web page allows the user to edit the source code of her/his robotics application, and to show that application running. There are several widgets to display the robotics simulated world, the robot in action, a text console, some auxiliary widgets for debugging and particular widgets that depend on each exercise.

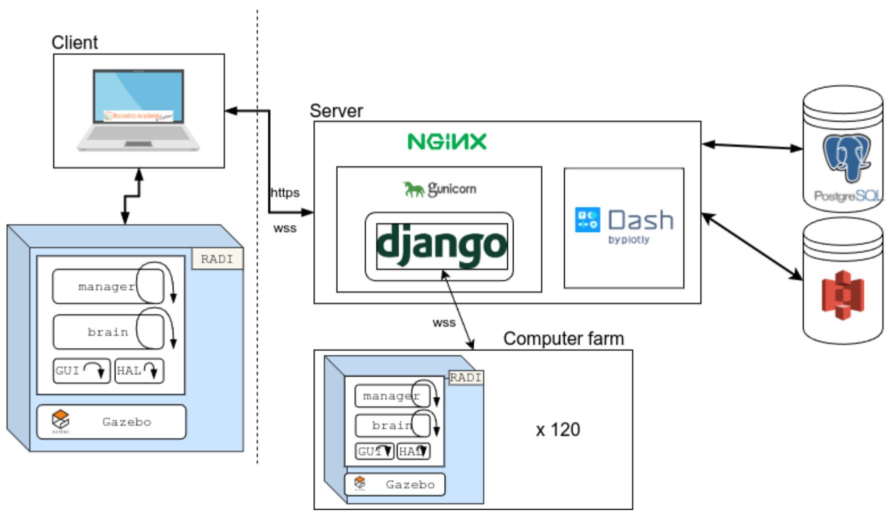

Fig. 1. Architecture of Unibotics platform.

The robot simulator and the robotics application itself both run inside a Docker container, named Robotics Academy Docker Image (RADI), which the user has to download from a public repository and has to launch it before using Unibotics. It is the *robotics backend* of the system, and all the robotics dependencies are preinstalled there. Therefore, the user does not have to install them on the local computer which greatly reduces the amount of time needed to start programming robots. Inside the RADI there is a Robot Application Manager (RAM) which acts as a bridge between the browser and the robotics applications inside the RADI. When the user runs the code through the browser, this code is sent to the RAM, which at the same time sends it to the robotics application template. The robotics application executes that code and then sends back to the RAM and then to the browser the results of the execution of that code (images, data, simulation result, etc).

2.1 REACT-based frontend

The major change in usability is intended with the complete redesign of the web pages of the exercises. In the previous version, the GUI of Unibotics was created using simple HTML templates. Each exercise had is own template, even when all of them shared the same widgets such as the buttons, the code editor or the gazebo widget. They were copied in each exercise folder. Since Unibotics is continuously growing and new exercises are added every year, this way of creating the templates for the exercises made the maintainability of the platform difficult. Therefore, we decided to change the frontend system for a method that would allow us to create components that could be reused in the different exercises that share it. In this way, if a component needs to be modified, the modification would be automatically extended to all the exercises that incorporate it.

To do so, we decided to evolve from static HTML templates to a more advanced GUI by using React. React is a Javascript based framework that allows programming web and native user interfaces. First, React enables efficient application state management, which facilitates the development of dynamic and interactive interfaces. In addition, React's inherent modularity facilitates component reuse, which speeds up the development process and improves code maintainability. DOM virtualization in React optimizes performance by updating only the necessary parts of the interface instead of reloading the entire page, resulting in a smoother user experience. In addition, React's component-based architecture facilitates collaboration between development teams, as different parts of the interface can be developed independently. In summary, adopting React for building web interfaces brings efficiency, scalability and improved user experience.

Figure 2 shows the old GUI for the Localized Vacuum Cleaner exercise, based on HTML and Javascript. Figure 3 shows its new GUI, based on REACT.

2.2 PostgreSQL Database

In the previous version of Unibotics, we used a MySQL database to store the structured data of the platform: users, exercises and farm machines available; and Elasticsearch to store information about the users and their interactions with the platform: operative systems, browsers, locations, scores in each exercise, time spent in the platform and time spent in an exercise.

Having two different databases to store the information increased the development and maintenance costs of Unibotics. To improve this aspect, we decided to migrate our storage system to a single one, PostgreSQL (as can be seen in Figure 1). Migrating from a MySQL database to PostgreSQL brings several substantial advantages. First, PostgreSQL is known for its robustness and compliance with SQL standards, resulting in greater data integrity and more accurate transaction handling. PostgreSQL offers support for advanced features such as stored

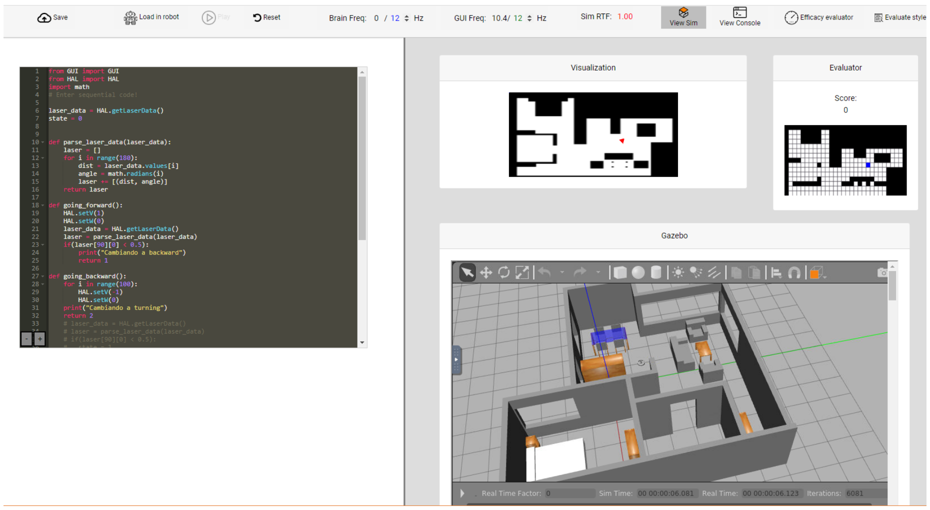

Fig. 2. Old GUI based on HTML and Javascript.

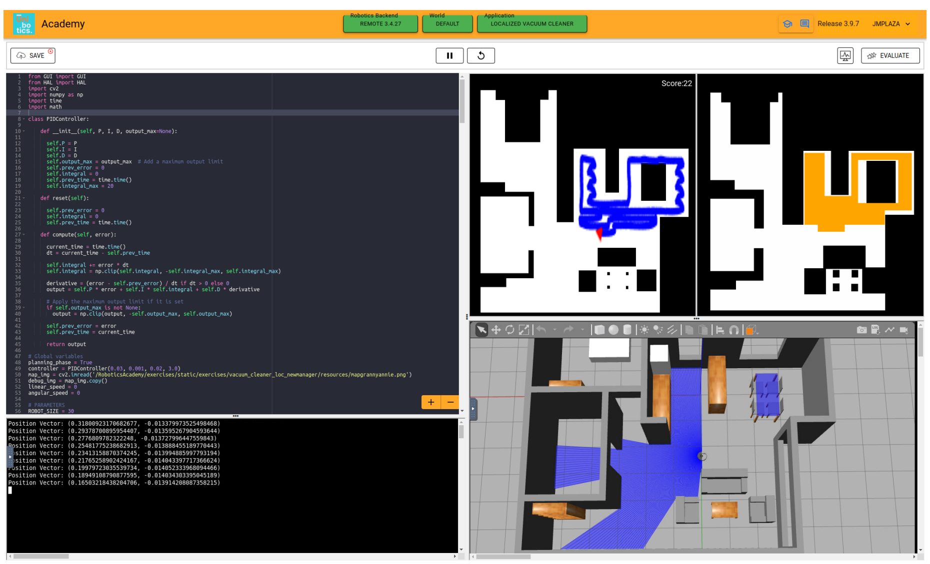

Fig. 3. New GUI based on React.

procedures, functions and triggers, providing greater flexibility in business logic implemented directly in the database. PostgreSQL's scalability is remarkable, allowing the efficient handling of larger and more complex data sets. In addition, its support for custom data types and extensions provides additional versatility. PostgreSQL's active and engaged community ensures frequent updates and ongoing support.

Including information that was previously stored in Elasticsearch into PostgreSQL has helped to simplify infrastructure and consolidate technologies. It reduced complexity by having a single database that handles both search needs and transactional operations. Moreover, PostgreSQL is easier to administer and in our case, it required less hardware resources to operate.

2.3 From ROS1 to ROS2 robotics middleware

In the previous version of Unibotics, the robotic exercises were designed using ROS1 Noetic. A few years ago, the ROS community designed the new generation, named ROS2, after years of experience and learnt lessons from widely using ROS1 in many robotics contexts. In order to take advantage of ROS2 improvements, we have also decided to update our exercises to ROS2 Humble.

The transition from ROS1 to ROS2 offers several significant advantages that improve efficiency and flexibility in the development of robotic systems. ROS2 introduces a more robust and versatile communication model, enabling greater scalability and performance compared to its predecessor [4]. This change facilitates the management of more complex distributed systems in varied environments.

In addition, ROS2 incorporates improvements in terms of security and reliability [9]. Its architecture enables more efficient resource management, reducing bottlenecks and improving fault tolerance. Emphasis has also been placed on interoperability with industry standards, facilitating the integration of hardware and software components from different vendors.

3 Teaching Experiences

In this study, the Unibotics platform has been tested in three different Spanish universities, with students from very different backgrounds. At the Jaume I University (UJI), it has been used with master's degree students. At the Universidad Rey Juan Carlos (URJC) and the Universidad Complutense de Madrid (UCM) the platform has been used in undergraduate courses.

3.1 M.Sc. in Intelligent Systems at UJI

In the autumn semester of academic year 2023/24 we have used for the first time the Unibotics platform at UJI, within the Master of Science in Intelligent Systems, in the laboratory works of a course on Cyber-Physical and Robotic Intelligent Systems. 24 were involved in the course, and the works consisted of six weekly sessions of one and a half hours each in the laboratory, and fifteen extra hours of personal work at home. The students had to solve two challenges at the Unibotics platform: "Follow Line", and "Rescue People".

From the point of view of the teacher, preparing the laboratory sessions was straightforward: it only required the installation of the Docker software and an Internet browser. The Unibotics Docker images were downloaded at the beginning of each session, the students launched a container locally, and connected to it with the browser. In this way, new versions of the Unibotics software with enhancements and bug fixes were used when available.

As for the students' opinions, they mostly like the platform though they complain about some buggy behaviors, especially when restarting the exercises. Nonetheless the results of the questionnaire are very similar to the rest of the students, with few slight differences. For example, only 14.29% of UJI students agree at some extent with the statement "We had to learn a lot of things before using the platform", whereas 33.95% of the rest of students agreed with it. The reason could be that the more complete background of M.Sc. students is an advantage compared to B.Sc. students when using the platform.

It is worth noting that the students have barely no background in robotics since most of them hold a B.Sc. in Computer Science. Nonetheless, they were able to understand the basic concepts and start programming robot behaviors easily.

3.2 Robotics in Computer Science Grades at UCM

The subject of Robotics in the Faculty of Computer Science at the UCM is an optional subject offered to third and fourth year students of the degrees of Software Engineering, Computer Engineering and Video Game Development. The aim of the course is to introduce some of the basic concepts of robotics to students with no previous knowledge of robotics, control or sensors. As an optional subject offered to different degrees, the students' starting backgrounds can be very diverse.

The Robotics course is taught in the autumn semester with two weekly sessions of 1h40m duration. One of these sessions is taught in the lab. Traditionally, the practice of this subject was only oriented towards the assembly and programming of a small mobile robot. Working with the robot is one of the main attractions of the subject and students appreciate learning how to assemble and program the hardware, despite the heavy workload.

In the Autumn 2023-24 course, the robot assembly and programming exercises were completed with 3 programming exercises using the Unibotics environment: Formula 1 "Follow Line", "Basic vacuum cleaner" and the "Obstacle Avoidance". These exercises allow students to program more complex robotic behaviours than those allowed by the hardware they have for the physical robot. 29 students were enrolled in this course.

Each exercise allowed students to explore a common programming technique in robotics: a PID controller in the "Follow Line" exercise, a state machine in the "Basic vacuum cleaner", and potential fields in the "Obstacle avoidance" exercise.

In general, students have rated very positively the possibility of programming more complex robots in a simple way. Some of the students' comments along these lines: "easy implementation of test code without having to install ROS", "the platform allows you to learn to program and simulate a wide variety of things. The tutorials are very useful.", or "the platform is great". Negative comments focus on the difficulty of controlling the cameras and the map.

3.3 Robotics Software Engineering degree at URJC

60 undergrad students participated in this pilot study at the URJC: 35 from the Mobile robotics course at third year, with 5 exercises: "Basic Vacuum Cleaner", "Follow Line", "Obstacle Avoidance", "Global Navigation" and "MonteCarlo Laser Localization"; And 25 from the Service Robotics course at fourth year, with 4 exercises: "Localized Vacuum Cleaner", "Drone Rescue People", "AutoParking" and "Amazon warehouse".

Both courses last 13 weeks, 4 class hours a week (2h for theory classes and 2h for practical classes with Unibotics), and their focus is on the robotics algorithms and the final applications, not in the ROS topics management which are taught in other courses. All the students had ROS prior knowledge.

Several minor problems arouse along the course, mainly with the new GUI, and they were solved in two working days at most, but the users were frequently required to update their RADI images. This was annoying and in the final survey many students complained about those stability problems in Unibotics. One aspect they valuate is the possibility of using Unibotics from home, not only from the classroom.

4 Measuring usability

At the end of the semester, the students were given a satisfaction questionnaire that they needed to complete for us to be able to assess Unibotics usability. This questionnaire is based on the System Usability Scale questionnaire [1]. It is composed of 13 questions, of which 10 allow students to evaluate the usability of the platform and 3 of them were questions about the specifications of the computers they were using to run the RADI (if they were using their own computers), one question about using ROS indirectly through the platform and an open question to express their opinions on the platform. In Table 1 we present the questions made to the students. In them, students were asked to show their agreement with the following statements (1 means completely disagree and 5 means completely agree).

4.1 Discussion

In 2023 we have 67 participants on the volunteer and anonymous survey (out of 110 students) compared to 22 participants in 2022.

Table 1. Satisfaction questionnaire questions

Id	Question
Q1	Tell us about the specifications of your PC
Q2	I would like to use Unibotics more frequently
Q3	Unibotics is too difficult to use
Q4	The platform is easy to use
Q5	I need technical help to be able to use Unibotics
Q6	The functionality of Unibotics is integrated smoothly
Q7	Unibotics is too inconsistent
Q8	I believe that most of the students would be able to use Unibotics quickly
Q9	Unibotics is cumbersome to use
Q10	Unibotics was comfortable to use
Q11	I needed a lot of previous knowledge to start using Unibotics
Q12	Tell us about the advantages and shortcomings of Unibotics
Q13	(Only if you have experience on ROS:) Programming robots with Unibotics is better than directly with ROS?

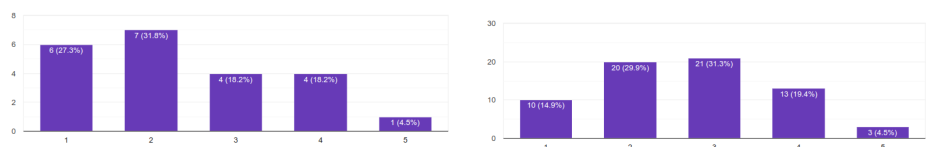

Fig. 4. Q10: 2022 (left) and 2023 (right)

Figures 4 and 5 show the comparison between the answers to questions Q10 and Q4 in 2022 and 2023. We can see an improvement in the answers, showing that students in 2023 feel more comfortable and find Unibotics easier to use than students in 2022 (the center of mass is closer to the 5 bin).

They also display that inconsistencies still are a problem, as shown by the answers to question Q7 (Fig. 6). According to Fig. 4, 45% of students think the platform is uncomfortable to use, and another 31% are not sure. We think the reasons for this dissatisfaction are the buggy behaviors detected along the pilot studies. Some of them were fixed but they required the students to update their RADI frequently, which is uncomfortable. In addition, a new refactoring of the dockerized execution for higher reliability is on its way with the next Unibotics release.

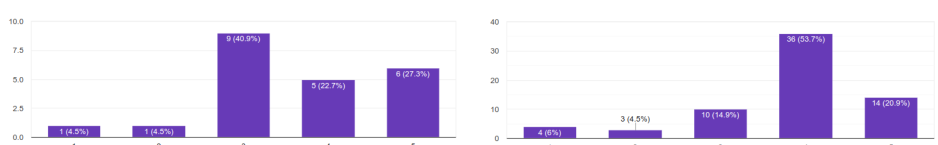

Fig. 5. Q4: 2022 (left) and 2023 (right)

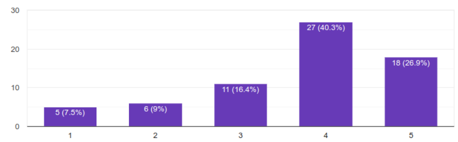

Fig. 6. Q7 2023

4.2 Unibotics usability in terms of students' prior knowledge

This study involved students with very different backgrounds in robotics. This allowed us to analyse the students' perception of the usability of the platform according to their previous education. The underlying research question is: *Is Unibotics an easy-to-use platform for students new to robotics?*. We have analysed the answers to questions Q10 and Q11 in Table 1, separating the answers of those students who have declared that they have no prior knowledge of ROS (33%) and those who do (67%), according to Q13 (Fig. 7). The preliminary data show that the platform is easy to use even for novice robotics users.

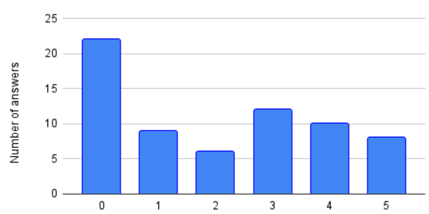

Fig. 7. Previous ROS background

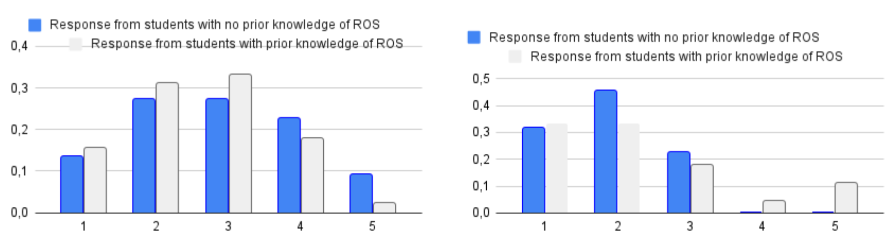

Fig. 8. Relation between ROS background and easy-to-use: Q10 (left), Q11 (right)

It can be seen from the Fig. 8 that the results (expressed in relative frequency) for both groups of students are very similar. In fact, students with no previous knowledge of ROS feel slightly more comfortable with Unibotics. The proportion of students who agree (score 4) or strongly agree (score 5) with the statement "Unibotics was comfortable to use" is slightly higher.

Regarding the statement "I needed a lot of previous knowledge to start using Unibotics" the results are similar. The proportion of students with no prior knowledge of ROS who disagree with this statement is higher than in the case of students with prior knowledge, although the difference is also small.

5 Conclusions

Over the past year, major changes have been made to Unibotics to improve both the usability and maintainability of the platform. These changes have been a clear enhancement for developers in terms of maintenance. Scalability has also been demonstrated as the number of users has doubled since last year (from 50 in 2022 to 110 in 2023).

In addition, we conducted a wider study through satisfaction surveys. These surveys were aimed at finding out students' impressions of Unibotics. As we have surveys from two different courses 2022-23 and 2023-24, we have been able to confirm that the most recent users perceive the platform to be easier to use than those of the previous academic year. The slightly better numbers in the usability questionnaire reflect the improvements done in the user interface.

Comparing the use of Unibotics for robotics learning with the traditional assembly and programming of a small mobile robot, the students are not required to install locally the ROS middleware or understand its internals (topics, master, drivers...). This installation, set-up and ROS introduction lectures typically took more than two weeks of the course length. In addition, students with MS Windows computers, or even without any ROS experience, can also perform the exercises. They all are much focused in the robotics algorithms (perception, control, navigation) than in the hardware assembly or the middleware itself.

One of the goals of Unibotics is to be a robotics learning platform for untrained students. Analysing the results of the surveys we have seen that users with no prior knowledge of robotics find the platform slightly more comfortable to use than students with previous knowledge of robotics. This indicates that Unibotics is a suitable tool for students to get started in the world of robotics.

The platform is fully functional and is a tool that can be of great value to students and professors, as this study has shown. But unreliability is the worst rated aspect of the platform, the inconsistencies are still a problem to be solved. The results of the questionnaire also confirm that working on improving the reliability of Unibotics is one pending key point to enhance the user experience.

Comparing with the web based robotics learning frameworks presented in Section 1, Unibotics supports many robot models and provides a large collection exercises which cover many robotics areas: mobile robotics, drones, autonomous driving, service robotics, etc. In addition, they are completely open source (RoboticsAcademy). Thought it still does not cover industrial robot arms and does not yet support cloud based execution of the robotics applications.

As future lines, we would like to increase the number of users. We also plan to carry out learning analytics analysis, in particular a comprehensive study on gender differences in robotic students. Finally, technical improvements upgrading to Gazebo Harmonic LTS and using Unibotics as a digital twin of the physical robot. In this direction we already have the "Follow Person" exercise available both with a simulated TurtleBot2 robot and a real one.

Acknowledgments

This research was partially funded by the UNIBOTICS-GAM project Ref. TED2021-132632B-I00, from Proyectos de Transición Ecológica y Transición Digital 2021 call Spain. Authors also appreciate the help of Google for improving RoboticsAcademy through the Google Summer of Code program since 2017.

References

1. Brooke, J.: Sus: A quick and dirty usability scale. Usability Eval. Ind. **189** (11 1995)
2. Casañ, G.A., Cervera, E.: The experience of the robot programming network initiative. Journal of Robotics **2018** (2018)
3. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, pp. 2149–2154. IEEE (2004)
4. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics **7**(66), eabm6074 (2022)
5. Prabha, V.E.M., Udayan, J.D.: Continuous integration pipeline for ros-based application using aws robomaker. In: 2023 4th IEEE Global Conference for Advancement in Technology (GCAT). pp. 1–7. IEEE (2023)
6. Quigley, M., Gerkey, B., Smart, W.D.: Programming Robots with ROS: a practical introduction to the Robot Operating System. " O'Reilly Media, Inc." (2015)
7. Roldán-Álvarez, D., Cañas, J.M., Valladares, D., Arias-Perez, P., Mahna, S.: Unibotics: open ros-based online framework for practical learning of robotics in higher education. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-023-17514-z
8. Roldán-Álvarez, D., Mahna, S., Cañas, J.M.: A ROS-based Open Web Platform for Intelligent Robotics Education, pp. 243–255. International Conference on Robotics in Education (RiE), Springer (2021)
9. Stavrinos, G.: ROS2 For ROS1 Users, pp. 31–42. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-45956-7_2, https://doi.org/10.1007/978-3-030-45956-7_2
10. Toris, R., Kammerl, J., Lu, D.V., Lee, J., Jenkins, O.C., Osentoski, S., Wills, M., Chernova, S.: Robot web tools: Efficient messaging for cloud robotics. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 4530–4537. IEEE (2015)
11. Zhang, H., Zhang, L.: Cloud robotics architecture: Trends and challenges. In: 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE). pp. 362–3625 (2019). https://doi.org/10.1109/SOSE.2019.00061