Research in Engineering Education Symposium &
Australasian Association for Engineering Education Conference
5 – 8 December, 2021 - Perth, WA

Robot Simulation for Teaching Engineering Concepts

Michael Finn, Travis Povey, Joel Frewin, Thomas Bräunl
The University of Western Australia
tb@ee.uwa.edu.au

ABSTRACT

CONTEXT

Teaching Robotics to Engineering students is at the same time exciting as well as a challenge. Students come from different areas of Engineering (Mechanical, Electrical or Software) and have therefore different previous knowledge. Also, visual geometric understanding and making the link between a line of software and the movement of a robotic vehicle are key elements.

While we are using real mobile robots for the student labs, the lab time is quite limited, so students can only prepare “in theory” at home. Also, real robots have real problems, such as battery problems, connection problems, sensor and actuator problems, and so on. These will reduce the available lab time and therefore diminish the actual learning outcome.

PURPOSE OR GOAL

We designed and implemented the EyeSim simulator for our EyeBot mobile robots. This simulation system is source code compatible, so robot programs on the simulator will run on the real robots without having to change a single line of code. The simulation system is free and allows students to implement the complete lab solution at home – without having to cope with any of the potential problems mentioned above.

APPROACH OR METHODOLOGY/METHODS

We conducted anonymous online surveys of students using the simulation system for the UWA units in Embedded Systems and in Robotics.

ACTUAL OR ANTICIPATED OUTCOMES

The vast majority of students reported that the simulation system was easy to use and that it benefited their understanding of robotics principles. They highly rated the ability to prepare, test and debug their software solutions on the simulator before they run it on the real robots.

CONCLUSIONS/RECOMMENDATIONS/SUMMARY

The use of simulation systems is seen as very beneficial for teaching robotics concepts. Students can practically prepare algorithms for labs and do not have to rely on theoretical approaches. Since the simulated robots do not suffer from battery, sensor or other failures, more time can be dedicated to robotics concepts and a higher level of complexity can be achieved.

KEYWORDS

Robot simulation, engineering concepts

Problem and motivation

Robotics is, at its core, a sub-discipline of Engineering that focuses on combining software and hardware to affect physical change in the world around us. With a diverse range of applications and new opportunities to explore, this burgeoning field is attractive to students who wish to learn and shape the technology of the future.

Vitko et al. (2010) argue that due to the complexity of the subject and its dependency on a variety of existing technologies it is difficult to teach robotics concepts through theory alone. Students must engage with robot systems that can actively sense and interact with their world in order to progress towards a more complete understanding of the field. In the UWA Robotics course, this engagement is primarily driven through weekly lab assessments.

Over-reliance on physical robots as teaching aids

As the primary goal of learning robotics is to be able to apply the knowledge to real-life applications, the use of physical robots as teaching tools is advantageous. However, there are several drawbacks to relying on real robots. Robotics hardware can be expensive, often requires regular maintenance and must be upgraded as the field of robotics itself evolves. There is also the risk that there may be too many students for the equipment available, or conversely that the purchased equipment is underutilized.

Due to logistical challenges, risk of physical harm and/or prohibitive cost, students may also have limited access to larger scale robotics systems such as industrial manipulators, autonomous vehicles and submersibles. Additionally, the increasing need for remote working arrangements (often at short notice) means that access to smaller scale robots may also be limited without warning during a teaching semester.

When using physical robots, a multitude of factors such as battery health, sensor degradation and manufacturing defects can create unwanted variance in behaviour, as found by Kumar (2004). This inconsistency can influence how the same student's code solution performs across different robots, potentially leading to unfair assessment.

Some students may also wish to design their own robots, but with typically limited time and resources this undertaking may not be feasible in a physical capacity.

Robotics lab assessments

Prior to 2017, lab assessments leveraged a fleet of custom-made teaching robots called EyeBots. The EyeBot is a small differential-drive robot equipped with simple sensors, running the RoBIOS operating system (Bräunl, 2003).

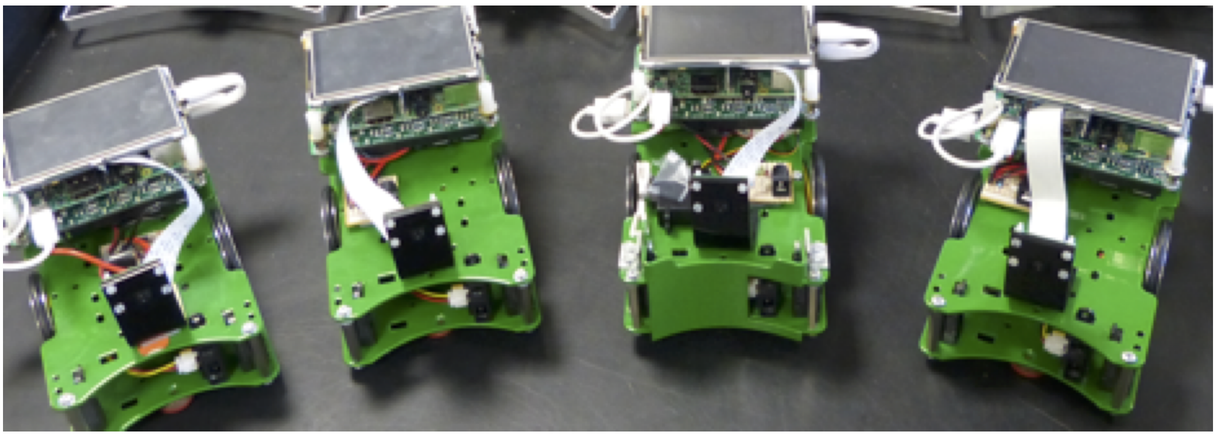

Figure 1: Four EyeBot 7 robots (the latest model)

Typical problems that students would be tasked with include driving a simple lawnmower pattern (Bräunl, 2021a), to using the built-in EyeBot camera to seek and transport objects (Bräunl, 2021b).

Since the EyeBots could not be made available outside of the prescribed time due to resource limitations, the majority of the work would have to be done in theory at home, with the limited lab assessment time used to actually run and debug the code. Additionally, as these EyeBots are used and maintained by students, they are susceptible to defects that can severely impact valuable lab time.

The EyeSim robot simulator

To address these issues, it was determined that a robot simulation tool could be leveraged. A basic multi-robot simulator, the original EyeSim, already existed and had been in use at UWA as a research tool, but its rudimentary graphics and lack of a realistic physics engine meant that a significant divide remained between the simulated and real-life robots. More popular solutions such as Gazebo (Koenig & Howard, 2004) and V-REP (Rohmer, Singh & Freese, 2013) had rich feature sets and realistic physical simulation, but were deemed unsuitable due to the relatively steep learning curve and lack of interoperability with RoBIOS.

Instead, the new EyeSim simulator (Bräunl, 2020) was developed. EyeSim is a free robot simulation platform that students can use at any time to rapidly deploy their code solutions at scale and with significantly more deterministic results than on physical robots.

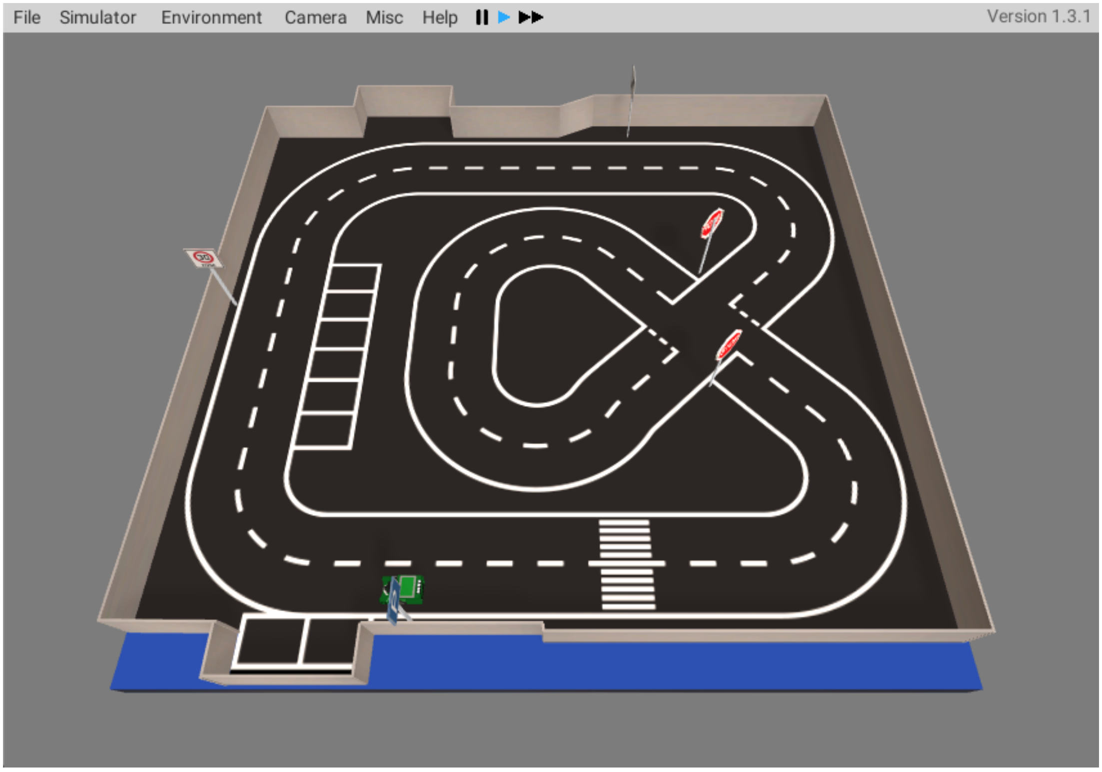

Figure 2: An example EyeSim scene

A 'scene' containing robots, objects and terrain (as shown in Figure 2) can be set up by hand, or via a specialised SIM file defining how a scene should be initialised that can be opened and processed within seconds. The ability to set up a scene this way, as opposed to manually resetting a physical environment, allows robot software development to be done faster and more iteratively.

The SIM file in Figure 3 creates the scene shown in Figure 2. The 'world' command selects the driving environment and defines its size, walls and any textures. The 'object' command introduces passive objects, such as the stop signs, which can then be included multiple times into the scene by using their respective keywords. Finally the keyword ‘S4’ adds a certain type of active mobile robot to the scene and specifies its starting position (x, y), its starting angle (phi) and its control program.

```
# Environment
world world/Carolo-lab.wld

# Settings
settings TRACE

# Objects
object ../../objects/StopSign/stopsign.esObj
StopSign 2300 1900 -45
StopSign 2280 1250 135

object ../../objects/ParkingSign/ParkingSign.esObj
ParkingSign 1050 230 180

object ../../objects/SpeedLimitSign/speedlimitsign.esObj
SpeedLimitSign 50 1700 90

object
../../objects/SpeedLimitSign/cancelspeedlimitsign.esObj
CancelSpeedLimitSign 2000 2900 0

# robotname x y phi
S4 1000 340 180 carolo.x
```

Figure 3: An example SIM file

An important feature of EyeSim is that the robot source code, written in C/C++ or Python, uses a standardised control API from the RoBIOS library (Bräunl, 2020). This means that a program can be written for a simulated robot, then run on a physical RoBIOS-based robot and vice-versa without requiring any code changes. This gives students the ability to develop code quickly and safely in simulation before deploying it on a physical robot, a feature that is especially powerful in terms of robots requiring lengthy initialisation/tear-down procedures, or those that are shared by many students at once.

This interoperability also means that students can be far more confident in the veracity of their assessment results, as the performance of their code will not be subject to varying physical factors. The high confidence in reproducibility also dispels the “it worked on my machine” excuse commonly used by students.

Using specialised ROBI robot definition files, students are also able to modify existing robots or create new ones from scratch within minutes. This provides the ability to easily prototype vehicles and manipulators before committing resources to a physical build.

Currently, the majority of lab assessments use EyeSim, while some assessments still focus on using EyeBots to provide students with a modicum of hands-on experience. The time requirement for assessments still exists, but in both EyeSim and EyeBot labs students now have the ability to debug their code from home using the simulator, shifting much of the workload away from lab time.

Survey of EyeSim users

Students that had used EyeSim in their units were surveyed once in Semester 2, 2017 and again in Semester 1, 2021 for a total of 25 respondents. The survey focused on determining the perceived impact of EyeSim on their experience of studying robotics.

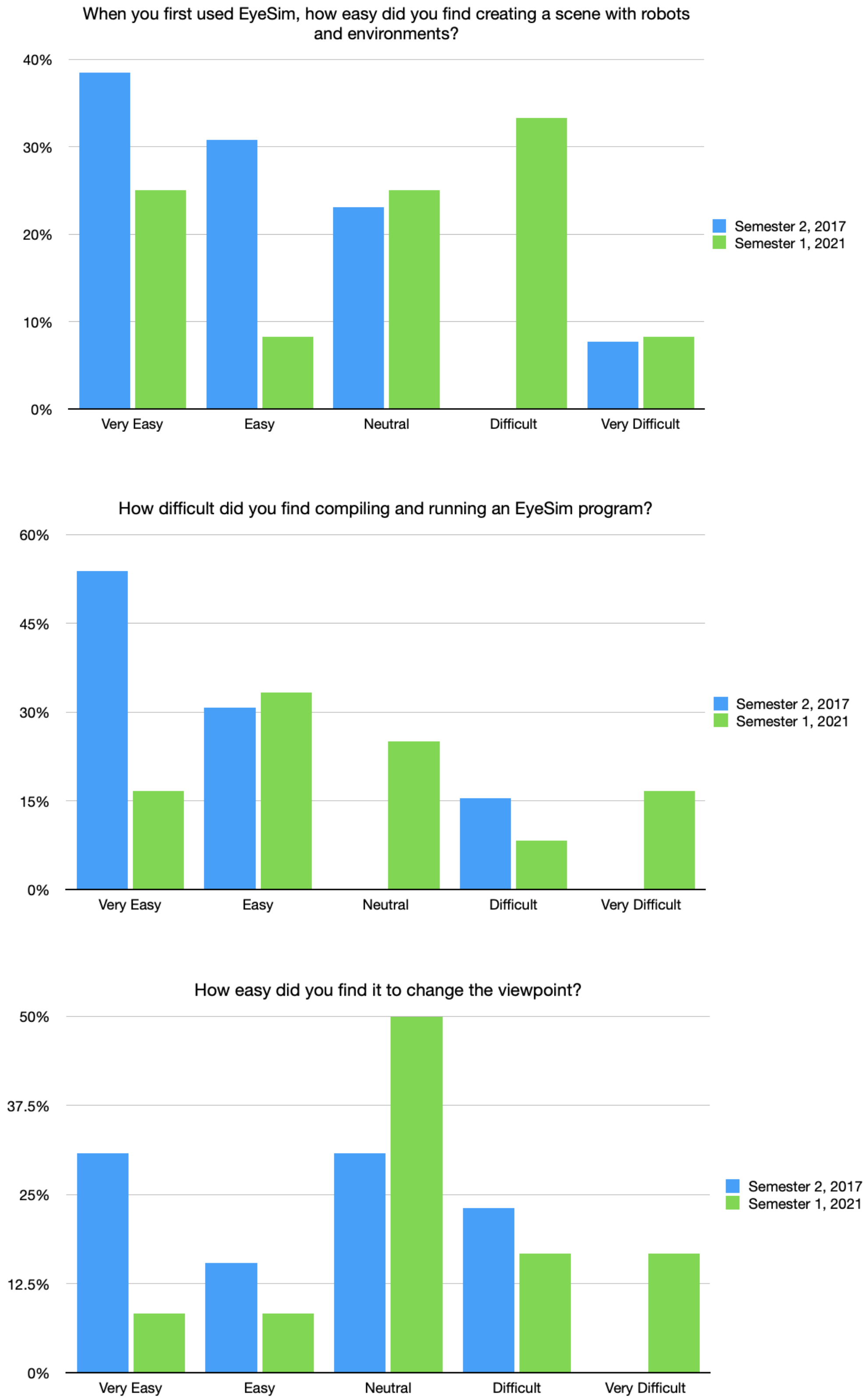

Figure 4: Survey questions relating to ease of use

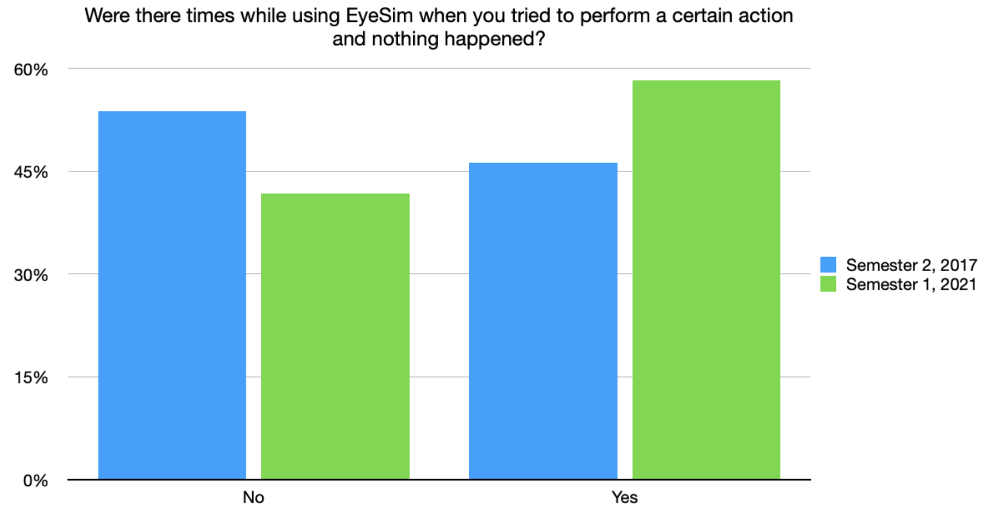

Figure 5: "Were there times while using EyeSim when you tried to perform a certain action and nothing happened?"

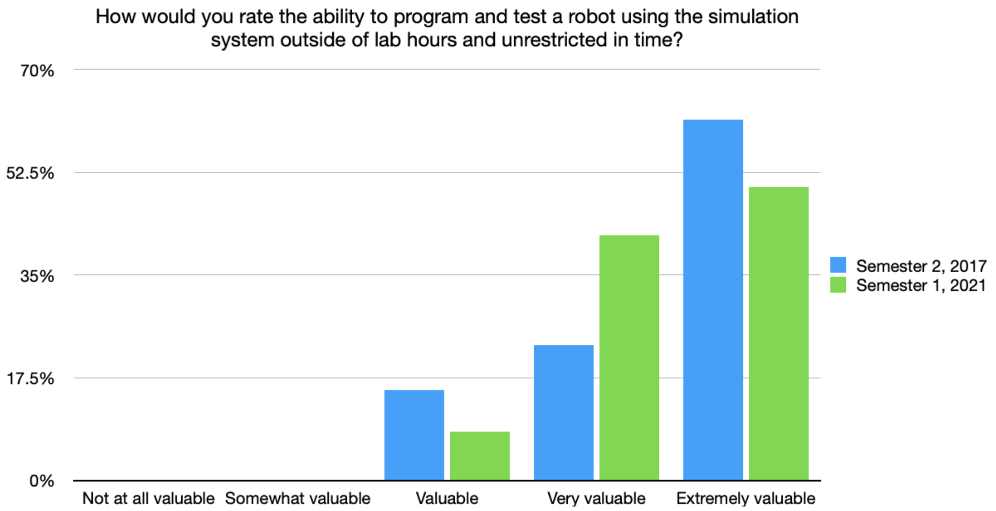

Figure 6: "How would you rate the ability to program and test a robot using the simulation system outside of lab hours and unrestricted in time?"

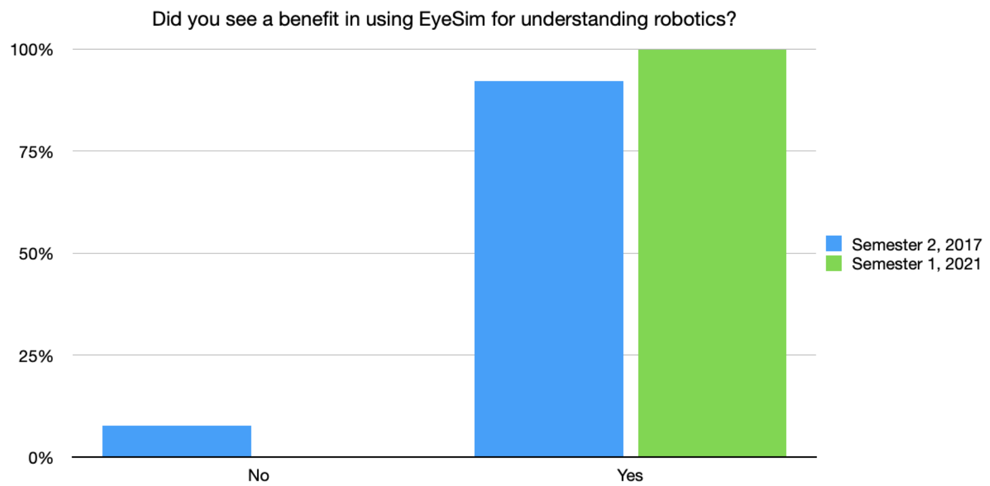

Figure 7: "Do you see a benefit in using EyeSim for understanding robotics?"

Discussion of results

The survey questions relating to ease of use (Figure 4) depict a generally positive response. Across each question in this set, it can be observed that responses from the 2017 survey round trend more clearly towards a positive response, whereas responses from the 2021 round trend towards a more mixed response. This shift correlates with several significant changes made to the EyeSim system between the two survey periods. The first question addresses scene creation, which was initially simple on the release of EyeSim, but became more complex as new features such as terrain, water and submersible robots were added as options. Similarly, the second question pertains to the difficulty of running control programs in EyeSim, which was initially constrained to C programs but had been expanded to also allow C++ and Python programs by the 2021 survey round. As EyeSim assessments begin within the first few weeks of the semester, students have limited time to familiarise themselves with the system, and it is suggested that increasing the feature complexity of EyeSim had a proportionally deleterious effect on ease of use. Since EyeSim is a research tool, new features will continue to be added, and so instead of reducing the feature set this issue could be addressed through more comprehensive documentation and training.

Figure 5 shows responses to the question “Were there times while using EyeSim when you tried to perform a certain action and nothing happened?”, and a roughly 50/50 split can be observed for both survey rounds, with a slight trend towards ‘Yes’ in the later round. This question was designed to assess the proportion of students that encountered defects, and many of these defects were enumerated in the long-form accessory section of this question. With about half of all students encountering defects, this defect rate is significant. Moreover, it is suggested that the higher defect rate in 2021 implied by the trend towards ‘Yes’ answers is consistent with the increased feature complexity; a greater number of interconnected features lends itself to a higher defect probability. As EyeSim is developed and maintained in the course of student projects, it may not be possible to reduce this defect rate in the future without acquiring funding for professional software development resources.

Figure 6 shows responses to the question “How would you rate the ability to program and test a robot using the simulation system outside of lab hours and unrestricted in time?”, with both survey rounds indicating a significantly positive response. As respondents had indicated that each lab assessment required on average 4-6 hours of preparation time prior to the lab session, it is unsurprising that they found EyeSim to be a useful tool in this capacity. Additionally, a majority of survey respondents had previously participated in lab assessments using the physical EyeBot robots and would therefore appreciate being able to run code outside of a lab session from this perspective.

The final question, “Do you see a benefit in using EyeSim for understanding robotics?”, directly asked respondents whether they felt EyeSim was a valuable teaching tool. An overwhelmingly positive response in both survey rounds can be observed, indicating that EyeSim had a positive effect on the perceived learning outcomes of the vast majority of students.

It is acknowledged that there is a time difference of four years between the 2017 and 2021 survey populations, however the structure of the UWA Robotics unit did not change significantly during this period. It is also important to note that the 2021 survey was performed during the COVID-19 pandemic, at which time UWA mandated mask wearing on campus, but student contact hours were unaffected.

It is also acknowledged that the questions in Figure 4 regarding ease of use are phrased in a leading manner that may have influenced a disproportionately positive response. This will be remedied in future surveys.

Summary

Overall, reasonably consistent results were observed across both survey rounds. The inconsistency in the ease of use section can be explained by the feature set expansion that took place between 2017 and 2021, and identifies a need to improve documentation and training processes for future cohorts. Additionally, the high proportion of students encountering defects is evidence that more resources need to be dedicated to quality control processes as new features are added.

The sample size of the survey was limited by the number of students in the unit, and although a larger number would have been desirable, the results seem to confirm our hypothesis that the use of a mobile robot simulator like EyeSim improves motivation and learning outcomes. We conclude that the ability to put theoretical robotics concepts into practice at any time, without the limitations and risks that accompany physical robots, is highly advantageous for robotics students. We expect that EyeSim will continue to mature and provide value to students in the years to come.

This research was made possible by a grant from the Education Futures Scholarship Program from the UWA Centre for Education Futures.

References

Bräunl, T. (2008). RoBIOS Operating System. In *Embedded Robotics,* 3rd Ed. (pp. 13-15). Springer, Berlin, Heidelberg.

Bräunl, T. (2020). *Robot adventures in Python and C* (pp. 1-183). Berlin: Springer.

Bräunl, T. (2021a). Lab Assignment 1 - Robot Driving. Retrieved 2 August 2021, from https://robotics.ee.uwa.edu.au/courses/robotics/labs/lab1-lawnmower.pdf

Bräunl, T. (2021b). Lab Assignment 9 - Robot Driving and Object Detection. Retrieved 2 August 2021, from https://robotics.ee.uwa.edu.au/courses/des/labs/lab9-C-cans.pdf

Koenig, N., & Howard, A. (2004, September). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)* (Vol. 3, pp. 2149-2154). IEEE.

Kumar, A. N. (2004). Three years of using robots in an artificial intelligence course: lessons learned. *Journal on Educational Resources in Computing (JERIC)*, *4*(3), 2-es.

Rohmer, E., Singh, S. P., & Freese, M. (2013, November). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1321-1326). IEEE.

Vitko, A., Jurisica, L., Babinec, A., Duchon, F., & Klúcik, M. (2010). Some didactic aspects of teaching robotics. *AT&P Journal Plus. ISSN*, 1336-5010.