RASPIED Project Proposal

An Educational Robot Platform for the UCT CS1 Extended Degree Program

Muhummad Patel
PTLMUH006
University of Cape Town

Jeremy Coupland
CPLJER001
University of Cape Town

Josh di Bona
DBNJOS001
University of Cape Town

1. PROJECT DESCRIPTION

The use of robots as educational tools has gained traction in recent years with numerous studies being conducted in this area. Most of the studies surveyed either purchased personal robots for every student, or shared very few robots among the class. Studies using the latter approach generally resulted in poorer student performance than those with personal robots. The lack of access to robots outside the classroom and students' inability to experiment at home were identified as the biggest problems with the shared robot approach[5].

Many studies have found that teaching Computer Science within a certain context can greatly improve students' learning experience[15, 17]. In particular, teaching Computer Science within the context of robotics has been shown to be beneficial in many cases[16, 14]. The use of robots as educational tools is supported by the Constructionism theory of learning[10].

While the use of robots as educational tools has been shown to be successful, prohibitively high costs still serve as a significant barrier to entry. Some of the difficulties faced by robot-aided curricula include: prohibitively high cost of hardware, appropriate choice of features to support the curriculum, and sufficient access to the robot. This project will design and implement a complete educational robot platform using only easily accessible, commercially available hardware and software. This platform will be built for use in the UCT CS1 extended degree Computer Science course. Two problems that will have to be overcome and/or worked around is the fact that students will have already used Python and already been taught most of the basic concepts by the time the platform is ready for user testing. This will need to be accounted for when designing the user tests. While most related studies evaluate their success based on the difference in students' marks before and after the addition of robots to the curriculum, this project uses other metrics (viz. student motivation), as a long-running user study would be infeasible.

2. PROBLEM STATEMENT

This project will design and implement a complete educational robotic platform built using easily accessible and commercially available hardware and software. The project aims to answer two research questions and also consists of a software engineering component.

2.1 Research Question 1

Can an educational robot, along with a high-level programming interface, be used in the UCT CS1 extended degree program to increase the motivation of the students learning basic Computer Science concepts? Previous studies investigating the use of educational robots in CS1 courses have found that it successfully motivates students[13, 9]; however, these studies generally either do not use validated questionnaires or do not address all the aspects of motivation [8]. These studies also used a participant group which was not homogeneous and contained possible discrepancies in student ability which could affect motivation and interest when learning basic concepts with a robot.

2.2 Research Question 2

Given a selection of cheaply available features to be used on an CS educational robot, which ones would students be most motivated to use, and which ones do students think would best help them understand the topic? Would these features be the same? In a review of the available literature, no studies were found that evaluated the effect on motivation of the individual features of the robot. This research question should help strike a balance between cost and functionality of educational robots.

2.3 Software Engineering Component

The software engineering component of this project involves building a web app to support the robot. Students will use this web app to interactively program and observe the robot. Students will also be able to book time slots to access the robot remotely should they wish to experiment outside of class time.

3. PROCEDURES AND METHODS

The educational robot platform will consist of an inexpensive robot built using a Raspberry Pi[12] (and other commercially available components), and a supporting web app. The robot will support a number of features to support the curriculum.

3.1 Features and Implementation Strategy

The robot will be developed on the Raspberry Pi platform in Python and will be controlled using a high-level Python robot module inspired by the Python Turtle module[11]. The module will abstract low-level commands that control the motors and interact with the features. This will provide simplified methods for movement, robot world-state (e.g. direction, position, etc.), pathfinding, and finally - as a strech goal - pen control with methods to draw. Students will be able to program the robot by importing the module and invoking the required methods. Extensive documentation of the module will be provided to ensure ease of use.

The robot module will contain similar methods to those seen in the Python Turtle module[11], such as forward(), turn() and pendown(). In keeping with the conventions of the Python Turtle module, the turn method will allow students to turn in a broad range of angles. The robot will keep track of its position and direction at all times to simplify remote access to the robot, and allow for the robot to be returned to its original position.

An additional method will be created to help introduce students to more complex concepts, in this case A* pathfinding. Students can enter parameters such as the positions of the obstacles and the destination. This method will return the shortest path in the form of a list of commands which can be run by the robot using another method.

The robot will make use of a number of features including: infrared sensors, an ultrasonic range sensor, a colour camera, and a 16x16 LED matrix. Students will have access to these features through the robot module mentioned above. The low-level sensor reading and interpretation will be handled by a separate features module that will be used by the robot module. The robot- and feature-specific code has been separated into individual modules such that these components can be easily developed in parallel.

The supporting web app component of this project will allow students to write Python programs for the robot, upload their code to the robot, and observe the behaviour of the robot via a live video stream. Furthermore, the web app will allow students to book time slots to use the robot outside the classroom. As a stretch goal, the web app will also allow students to run their code on a robot simulator that replicates the functionality of the physical robot. This should help reduce the frustration caused by wait times for access to the shared robot.

The robot will be placed in a designated enclosure in a supervised location available to students. The Computer Science Hotseat would be a prime location for the robot as this would allow students to physically interact with the robot during the Hotseat hours. A Raspberry Pi with an attached webcam will be mounted above the enclosure to capture and stream a live video feed of the robot. This will allow students to observe the robot remotely outside of the Hotseat operating hours.

The educational robot platform has been divided into three components to be implemented by each of the team members. The first component involves constructing the robot and implementing the robot module. The second component involves the implementation of the features module such that the robot module can utilise the features. The third component is the software engineering component and consists of the supporting web app and programming interface which will allow students to write and upload code to the robot.

The software engineering component of this project will be developed using the Scrum development methodology. This iterative approach will allow for team members to regularly touch-base and ensure that all components are compatible. The web app, robot, and associated features will be developed in 1 week sprints with weekly group meetings to evaluate the previous sprint, re-prioritize the task queue, and plan the next sprint. The first sprint will investigate available web-frameworks to determine which would best serve the requirements of this project. The remaining sprints will be used to develop an initial proof-of-concept and then slowly add on and improve the required features. Student feedback will be used to guide the design of the various functionalities and user interfaces.

One of the major challenges in implementing this project is maintaining a consistent view of the system among team members and ensuring that all components are compatible. To reduce the risk of the various components being incompatible, all inter-component communication will be carried out through agreed-upon interfaces (i.e. the Programming Interface and the Features Module). Furthermore, the weekly team meeting will be used to ensure that all team members' view of the system remains consistent and the components are still compatible.

The inherent inaccuracies of the hardware will pose some interesting design challenges for the robot and features components of the project. For example, the construction of the robot and implementation of the robot module will have to account for the possibility that the robot's actual world-state disagrees with its internal world-state as a result of external forces.

3.2 Evaluation and Testing

User Testing will be done to test for an increase in the levels of students' motivation when using the robot. The CSC1010H class will be used for the testing, possibly two at a time. In order to test for an increase in motivation, a control group will be needed, where students are given a questionnaire to test their current motivation using the tools they currently use (including Python Turtle). The experimental group will not be given a pre-task questionnaire so as not to influence the answers given after the task. Students in the experimental group will have to complete a set of tasks given to them after which they will be given a questionnaire measuring their motivation as well as a section allowing them to enter any additional comments about the robot and the module. The questionnaire that will be used has been developed by McGill and is targeted at measuring student motivation in robot-aided Computer Science courses[8]. This questionnaire measures the different components of motivation and will be used in evaluating the robot component of the project.

The success of the web app will be evaluated based on how well it meets the initial requirements. A successful web app will allow students to remotely write programs for the robot, upload code to the robot, observe the robot through a live video stream, and book time slots to use the robot. The successful web app should, at the very least, not serve as a hindrance and, at best, enhance the experience of programming the robot. This will be measured through user testing where students will use the robot directly (no web app) and then also through the web app. The students will then be surveyed to help determine to what degree the web app has succeeded in fulfilling the initial specification.

To test and evaluate the advanced features, students will be given a simple task using the relevant feature. This task will let the students understand how the feature works and see how it may be applicable in future educational assignments. The students will then be asked which feature they preferred using, and which feature they felt best demonstrated the Computer Science principle that the task addresses. In the cases where the students have not yet covered these Computer Science principles in class and therefore lack the knowledge to complete a task, or if the robot is not yet functioning and able to be used for the tasks, the features will simply be demonstrated. This will still allow the students to see how the features work. The results will then be captured and a statistical analysis will be performed to see if the results are significant.

4. ETHICAL, PROFESSIONAL, AND LEGAL ISSUES

The project will make use of user testing with participants from the CSC1010H class, as a result ethical clearance will be needed from the Science Faculty Ethics Committee and Student Affairs. The robotics platform created will be licensed as open source software. Furthermore, all team members will be required to maintain a professional demeanour when carrying out user tests.

5. RELATED WORK

Much of the previous work in the area of educational robotics has made use of the Lego Mindstorms platform[3, 6]. The Lego Mindstorms platform provides a number of input sensors including touch, light, and rotation sensors. Output devices such as servo motors are also provided. The feature set supported by the Lego Mindstorms hardware does not appear to be a limiting factor and it has been successfully used to teach advanced concepts such as Artificial Intelligence[6]. The Lego Mindstorms sensors and motors come packaged in convenient plug-and-play modules that have been made to work within the Mindstorms platform; however, the official Mindstorms modules are far more expensive than the raw electronic components.

The major limitations of the Lego Mindstorms platform include the limited on-board memory and the lack of support for useful wireless communication protocols[6]. Another common criticism of the Lego Mindstorms platform - particularly in the context of tertiary-level courses - is that the official graphical (drag-and-drop) programming language is too simple[6]. Prior studies have addressed this issue of the simplicity of the default programming language by making use of third party programming tools such as the Not Quite C (NQC)[1] programming environment[6], and the Ada/Mindstorms 2.0[3] programming environment[4].

Furthermore, while a previous study investigating the suitability of the Lego Mindstorms platform concludes that it is a cost-effective, viable option for the inclusion of a robotic element in Computer Science courses[6]; the cost of the platform is arguably still a non-trivial barrier to entry.

In a survey of related works, one study was found that made use of a Raspberry Pi server to allow students to program robots through a web-interface[13]. This study was aimed at 3rd and 4th grade students and did not allow for remote access to the robot as the web-tool was only made available in the classroom during instruction time. Another study, carried out in a senior Computer Science course at California State University, used a remotely accessible web server and a webcam to allow students to view the robot and send commands to it remotely[2]. This setup used a single robot and scheduled students' access to the robot using the web app. The web app being developed as part of this project aims to merge the full programmability afforded by the first study, with the webcam and scheduling provided in the second study.

When looking at an interface module for the robot platform, a few papers were found that made use of either Pyro or Myro[14, 7], which is an application program interface (API) developed to help students write robot-independent programs. It includes a set of robot control functions and additional helper functions to introduce students to more complex concepts as well as the basic concepts. Its main aims were it being platform independent and it currently works with a few off the shelf robots. For the analysis of the effectiveness of the robots in increasing the student's motivation, McGill does an in depth review of previous studies and comes to the conclusion that validated tools do exist, although most papers don't use validated tools or do any sort of quantitative analysis of their results and that motivation is made up of four parts consisting of attention, relevance, confidence and satisfaction. She suggests and uses a combined questionnaire made up from different validated tools and focuses on motivation in CS1 with the use of a robot along with the relevant statistical analysis.

6. ANTICIPATED OUTCOMES

The following section outlines the expected deliverables of this project. This entails the entire educational robot platform's individual components, the impact that this platform will have, and how the success of each component will be measured.

6.1 System

One of the main deliverables of this project is a programmable educational robot equipped with interesting and useful features. The robot will be capable of basic movement as well as advanced features such as obstacle detection. The movement and features will be programmable using a high-level Python module. A list of features will also be produced, showing which feature can best be used to increase a student's motivation and learning experience for a specific Computer Science concept. Along with the robot, a web app will be produced, allowing students to program and observe the robot remotely. Figure 1 shows an overview of the educational robot platform components and how the development of these components is split between team members.

A design challenge faced by the project is that since the web app, the basic robot, and support the advanced features are being developed in parallel, there was a need to test each component of the project individually and irrespective to one another. This will be accounted for by providing method stubs, which can be used by each component in place of the actual method which may or may not have been developed yet in the other component. Method stubs will also allow students to test their code for compilation errors with a mock robot module before running their code on the robot. Another design challenge faced by both the features

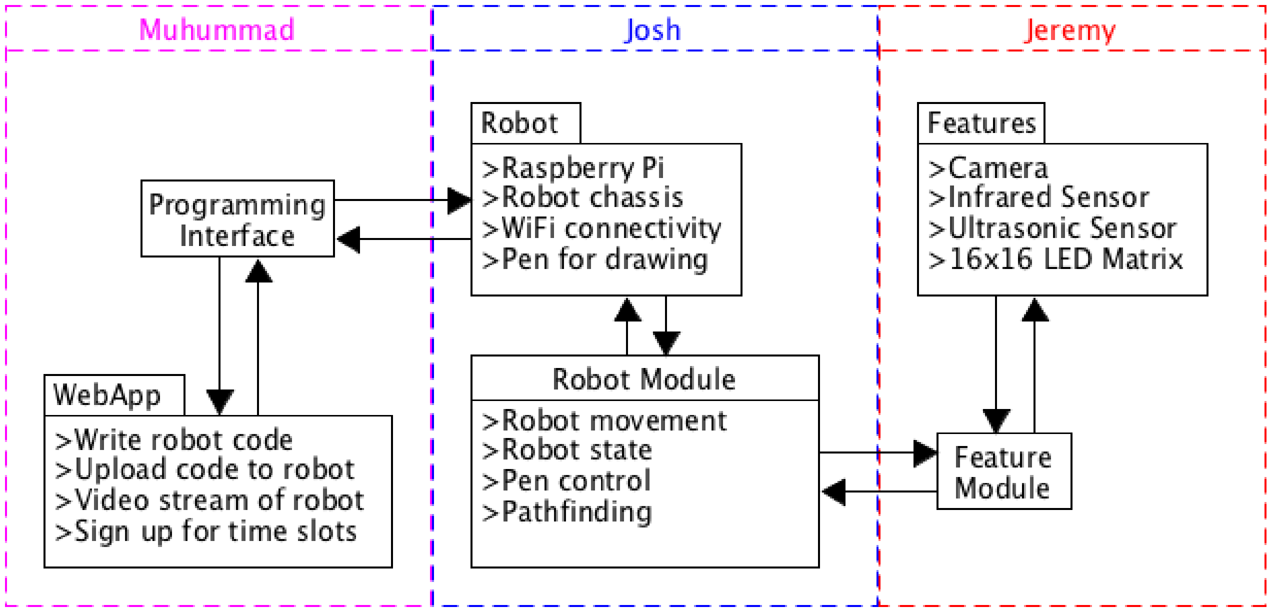

Figure 1: Overview of the Educational Robot Platform showing the relationship between the components

and robot module sections is the fact that software cannot cater for any hardware inaccuracies, which could occur from external sources such as students moving the robot or the motors not performing the same as when the power being supplied to the robot is not always constant. This will be prevented by using a permanent power supply wired to the robot instead of a battery pack, which will also reduce the costs and maintenance involved in running the robot. The inaccuracies of the hardware will need to be accounted for when implementing the return-to-origin method and turning methods of the robot movement module. A grid based system will be used to help manage the position of the robot and serve as a visual reference for students. The stretch goal of pen control and drawing brings about many challenges such as a reliable mechanism to remotely erase the drawings made by students.

6.2 Impact

It is anticipated that the educational robot platform will increase the students' motivation. Along with the robot supporting a variety of features, the project will also produce a list of which feature most excites and motivates the students with regard to a specific Computer Science concept, as well as which feature students think best illustrates the concept. Whether these features would be the same remains to be seen. This list of features could help future educators who may be exploring the idea of using a robot for Computer Science education by highlighting which features would be best received by students.

All software produced by this project, including the web app, will be released under an Open Source License and will be available for use by anyone wishing to implement a similar system. This should help speed up and simplify the process of integrating robots into Computer Science curricula in the future.

6.3 Measuring Success

Students are able to complete selected tasks using the high-level interface, while writing and running their code using the web app. All features on the robot are usable and suggestions can be made for future feature/component purchases (for features/components that support the curriculum) and robot design. Students show higher levels of motivation from using the robot than the students in the control group.

7. PROJECT PLAN

This section explains the project plan/schedule, and how the workload is split between team members. Included in this section is a risk analysis matrix (see Figure 2), a Gantt chart showing the project schedule including important milestones (see Figure 3), and a list of the resources that will be required during the project.

The project will make use of the following resources:

- 2x Raspberry Pi (Robot and Web-cam controller)
- Robot chassis and motors (2x DC/Stepper, 1x Servo) with driver circuits
- 2x WiFi dongle (Robot and Server)
- 2x SD card (Robot and Server)
- 4x infrared sensor
- 1x ultrasonic range sensor
- 2x colour camera/webcam
- 1x 16x16 LED matrix

Risk	Probability	Impact	Mitigation
Components fail to arrive on schedule.	Low	Medium	Seek local alternative replacements/substitutions.
Components break, get lost or are stolen.	High	Medium	Purchase spare parts.
Milestones/deadlines being missed	Medium	High	Adjust the schedule to work through weekends.
Team member drops out.	Low	High	Reduce the scope of the project.
Ethical clearance denied.	Low	High	Adjust the aims and methods of the project.
Insufficient users for user testing	Medium	High	Open user testing to the entire CS1 year instead of just the CSC1010H course.
Web framework lacks some features	Medium	Low	Adjust requirements or pivot to use another web framework.

Figure 2: Risk analysis matrix outlining the major risks associated with the project

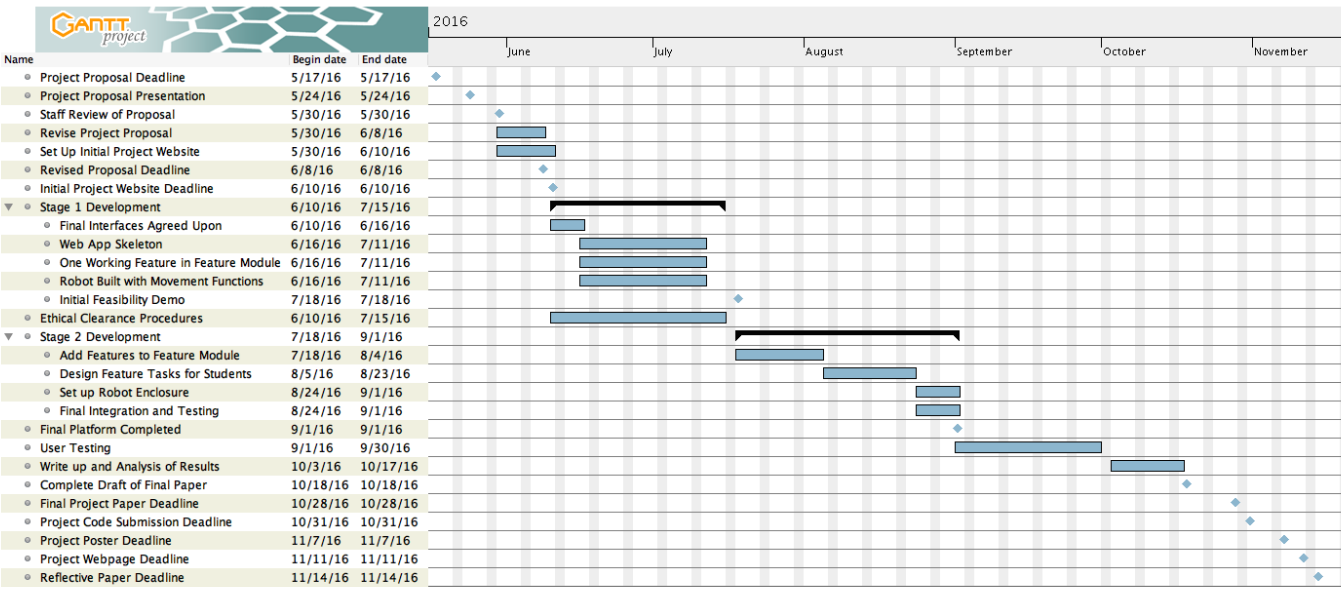

Figure 3: Gantt chart showing project schedule and major milestones

- Items to build the enclosure and the floor grid
- 2x Batteries (4x AA rechargeable) or a micro USB cable
- Students for user testing

The following is a list of the anticipated deliverables for this project:

- A functioning robot with all the features included
- A grid for the robot to be used on as well as a Raspberry Pi and webcam above it
- Student-facing interface (as a Python module) for programming the robot
- A report of which feature could be best used to illustrate various concepts
- A report on the motivation of the students using the robot with applicable statistical analysis
- A report on the results of the software engineering, web app component

The work will be divided up among the group members as follows:

- *Muhummad*: Build the web app and set-up the Raspberry Pi that controls the webcam
- *Josh*: Build the robot, the module that contains high-level methods for movement, robot state, pathfinding and pen control and set up the grid the robots will move on
- *Jeremy*: Implement the different robot features (camera, infrared sensors, ultrasonic sensor, and LED matrix) and create a module that contains the high-level methods that utilise these features

8. REFERENCES

[1] D. Baum and J. Hansen. Nqc programmer's guide. 2003.

[2] J. Challinger. Efficient use of robots in the undergraduate curriculum. In *ACM SIGCSE Bulletin*, volume 37, pages 436–440. ACM, 2005.

[3] B. Fagin. An ada interface to lego mindstorms. *ACM SIGAda Ada Letters*, 20(3):20–40, 2000.

[4] B. Fagin and L. Merkle. Measuring the effectiveness of robots in teaching computer science. In *ACM SIGCSE Bulletin*, volume 35, pages 307–311. ACM, 2003.

[5] B. S. Fagin and L. Merkle. Quantitative analysis of the effects of robots on introductory computer science education. *Journal on Educational Resources in Computing (JERIC)*, 2(4):2, 2002.

[6] F. Klassner. A case study of lego mindstorms' suitability for artificial intelligence and robotics courses at the college level. In *ACM SIGCSE Bulletin*, volume 34, pages 8–12. ACM, 2002.

[7] S. A. Markham and K. King. Using personal robots in cs1: experiences, outcomes, and attitudinal influences. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, pages 204–208. ACM, 2010.

[8] M. M. McGill. Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education (TOCE)*, 12(1):4, 2012.

[9] S. McNamara, M. Cyr, C. Rogers, and B. Bratzel. Lego brick sculptures and robotics in education. In *ASEE Proc*, 1999.

[10] S. Papert and I. Harel. Situating constructionism. *Constructionism*, 36:1–11, 1991.

[11] Python Software Foundation. Python Turtle Documentation. https://docs.python.org/2/library/turtle.html.

[12] Raspberry Pi Foundation. Raspberry Pi Website. https://www.raspberrypi.org/.

[13] M. Saleiro, B. Carmo, J. M. Rodrigues, and J. H. du Buf. A low-cost classroom-oriented educational robotics system. In *Social Robotics*, pages 74–83. Springer, 2013.

[14] J. Summet, D. Kumar, K. O'Hara, D. Walker, L. Ni, D. Blank, and T. Balch. Personalizing cs1 with robots. In *ACM SIGCSE Bulletin*, volume 41, pages 433–437. ACM, 2009.

[15] D. Xu, D. Blank, and D. Kumar. Games, robots, and robot games: complementary contexts for introductory computing education. In *Proceedings of the 3rd international conference on Game development in computer science education*, pages 66–70. ACM, 2008.

[16] A. Yadin. Reducing the dropout rate in an introductory programming course. *ACM inroads*, 2(4):71–76, 2011.

[17] S. Yarosh and M. Guzdial. Narrating data structures: The role of context in cs2. *Journal on Educational Resources in Computing (JERIC)*, 7(4):6, 2008.