UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER

Computer Science Honours
Final Paper
2016

Title: A study of how well certain robotic features can teach various introductory Computer Science concepts

Author: Jeremy Coupland

Project Abbreviation: RASPIED

Supervisor(s): Gary Stewart

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	10
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	15
System Development and Implementation	0	15	15
Results, Findings and Conclusion	10	20	10
Aim Formulation and Background Work	10	15	10
Quality of Paper Writing and Presentation	10		10
Quality of Deliverables	10		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	
Total marks		**80**	

A study of how well certain robotic features can teach various introductory Computer Science concepts

Jeremy Coupland
University of Cape Town
CPLJER001@myuct.ac.za

ABSTRACT

This paper presents the RASPIED robot platform, which consists of a physical robot, built using a Raspberry Pi 3 and a robot chassis, and an interactive web app which can be used to program to robot via an API. Four additional features were added to the robot, an infrared sensor, an ultrasonic sensor, a camera, and three different coloured LED lights. The robot platform was tested over two phases on seven students, two Masters students and five Honours students, who are currently studying Computer Science at the University of Cape Town. Four Computer Science concepts were tested, conditionals, for loops, while loops, and functions, in an attempt to find out which feature can best be used to teach each concept in a cost effective manner. For conditionals, it was found that the infrared sensor was the best. For teaching for loops, the infrared sensor and the camera were both found to be suitable candidates. While loops identified suitable candidates to be both the infrared sensor and the LEDs. For functions, it was found that the ultrasonic sensor or the camera were satisfactory teaching tools. It was concluded that the most cost effective and versatile feature is the infrared sensor.

CCS Concepts

• **Computer systems organization → Embedded and cyber-physical systems → Robotics.**

Keywords

Educational robot, Education, Pedagogy, CS1

1. Introduction

Over the past few decades, Computer Science educators have noticed a decrease in the number of students majoring in Computer Science [1] [2] [3]. Researchers have identified many factors that may be contributing to this decrease, many of them being social factors. Students often view Computer Science as a very asocial subject, others view Computer Science as tedious, boring and irrelevant, with no room for creativity [1]. Other researchers have found that certain programming languages, and the syntax associated with those languages, have been deterring students [4]. A combination of these factors has resulted in a decrease in the number of students majoring in Computer Science, and also an increase in the number of students deciding to drop out or change majors [5]. Because of this, educators have been searching and experimenting with different methods of teaching Computer Science, in an attempt to change the students' and society's perspective.

Many educators have found that teaching Computer Science within a certain context can greatly improve a student's learning experience [6] [7]. This entails teaching students basic Computer Science concepts, but framing them around a real life application, so that the students can see how the concepts are directly applicable to real life problems. One such context that is being explored is robotics.

When the term *robot* is mentioned, it can conjure various ideas based on the reader's personal experience and exposure to the field of robotics. A robot is simply a machine that can be programmed by a computer to carry out a complex series of actions. There have been several different robots used in education, some made specifically by the institution (e.g. the *e-puck* [8]), and others which are commercially available at varying prices, including the *Electric Ray Robot* [9], the *Ridgesoft Intellibrain-Bot* [10], the *Scribbler* [11] [13] [16] [17], and the *LEGO Mindstorms* [12] [18] [19]. While each of these robots differ with regards to the microcontrollers and programming languages used, they all, with the exception of the *LEGO Mindstorms*, share similar characteristics. They are a similar size, roughly 20x15x10 centimeters, they have two wheels and an Omni wheel/ball to allow movement, and can be equipped with additional sensors to increase their uses.

For this project, the author and two other group members, Josh di Bona and Muhummad Patel, have built the RASPIED robot platform, using a commercially available robot chassis, a Raspberry Pi 3, and a small collection of features (two different sensors, three different coloured LED lights, and a camera). The target users of this robot platform are the first year participants of the University of Cape Town's Computer Science extended degree programme. To program the robot, an application programming interface (API) was created, along with a web application which allows the robot to be programmed remotely. This report will detail the construction of the robot, and how the robot has been evaluated with respect to how effective each feature was in teaching various introductory Computer Science concepts.

2. Related Work

There have been several studies that have tested a robot-based Computer Science curriculum on a large number of students. One such study [13] at the Georgia Institute of Technology and Bryn Mawr College involved around 1500 students, 144 of which were taught using the robot-based curriculum and the *Scribbler* robot. Other students in this experiment were students in one of three other introductory Computer Science courses. It was found that, out of these four courses, the students in the robotic-based course had the highest pass rate of 90.97%, the second highest pass rate of 85.71% was obtained by a course that taught the same concepts without robots. A final exam was administered to robotics and non-robotics students, examining the same concepts, and it was found that on average students in the robotics course performed 10% better than non-robotics students, a result that was proven to be statistically significant. It was also noted that the average number of students enrolled in the second year Computer Science course more than doubled after the robot-based course was introduced.

Another large scale study was performed by Barry Fagin, Laurence Merkle and Thomas Eggers. This study was performed and documented in 2001 [12] and the following year the results were published [14]. It was found that out of the 948 students involved in the experiment, 175 of which were taught using the *LEGO Mindstorms* robot, and remaining students were taught using the conventional methods. The study recorded both quantitative and qualitative results from students. They found that with respect to results, the robotics students performed worse than non-robotics students in every instance where a difference was detected. Qualitative measures of this experiment included a survey administered to all students asking them to rate the course as a whole, the relevance and usefulness of the course, the amount that they learned and the effectiveness of their instructors on a Likert scale. In all four questions, the robotics students rated lower values than the non-robotics students. The study found that the biggest disadvantage of the robot-based curriculum was that students did not have access to the robot at all times, as the robots were required to remain in the labs. Without access to the robots, students were only able to work on assignments during designated lab sessions, which the authors suspect is the reason for the discrepancy between the robotics and non-robotics students' results.

One study created their own robot, the *e-puck* [8], and used it to teach 273 students C and C++ programming over the course of three years, starting in 2005. At the end of each course, students were asked two Likert scale questions: "The e-puck robot is a good tool to illustrate the concepts of the course" and "The e-puck is performing well". Over the three years, ratings have improved and are mostly positive, but no statistical analysis or quantitative data was recorded.

3. Research Question

Since there is no formal evaluation method for measuring the qualitative feedback of the students after the completion of the courses, educators have chosen to create their own questionnaires or surveys, or simply interview the students and make notes on their experiences [8] [12]. In previous studies, these results have been presented as *course ratings increased* [8] or *robot student ratings were lower than those of non-robot students* [12]. While these results do provide some useful information, they provide little insight as to what worked well and what did not, or how to improve the course so that it would be better received by students. With this is mind, the following research question has been formulated:

- Given a selection of cheaply available features to be used on a 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?

This research question will help provide insight to future educators who may be interested in creating a robot-based curriculum. The educators will be able to tell which features work well as well as allowing them to find a balance between the cost and functionality of the robot.

4. System Design

The entire robot platform, consisting of the physical robot and the web application, was created by the author and two group members, Josh di Bona and Muhummad Patel. The web application, created by Muhummad Patel, allows the robot to be programmed in Python through an interactive terminal. The web application collects the students' code and executes it on the robot, and shows the student a live video stream of the robots actions, as well as giving them an output terminal displaying the output of their program. This web application is hosted on a laptop and connects to another Raspberry Pi 3 which runs a video stream server and uses the Raspberry Pi Camera Module to capture the live video. The movement and construction of the robot was handled by Josh di Bona, with a specific movement API. The robot was made to operate in a grid environment. The movement functions involve moving forward/backward one or more blocks, or turning in 45 degree steps. The author was responsible for enhancing the capabilities of the robot by adding certain features. The features that were chosen were an infrared sensor, and ultrasonic sensor, a camera and three different colour LED lights, each one accessible via a features API. The movement and features APIs were made utilizing the Raspberry Pi GPIO Python module, which gives access to the pins on the Raspberry Pi. Documentation of the necessary movement and feature functions were made available to the students via a GitHub wiki.

4.1 Robot Design

The robot was made using a Raspberry Pi 3 and the Magician chassis. This chassis comes with two DC motors, and two anchored wheels, capable of reversing and accelerating, and an Omni wheel (ball) which allows turning. Attached to the Raspberry Pi is a motor board, which allows control over the motors. To wirelessly power the Raspberry Pi and the motor board, a power bank was used. The Raspberry Pi 3 has 40 pins, 26 of which were used by the motor board. This left 14 pins free which could be used to power the features. The features that were selected for the robot were an infrared (IR) sensor, an ultrasonic (US) sensor, a camera, and three LED lights.

4.1.1 The Infrared Sensor

The IR sensor works by having a small LED light that emits infrared light and a receiving tube that detects infrared light. If an object passes in front of the IR sensor, the infrared light is reflected back into the receiving tube, letting the sensor know that there is an obstacle nearby. This feature was placed on the front of the robot to allow the robot to avoid obstacles. The IR sensor has two functions in the features API: *ir.detect_obstacle()* which will return True of False when run depending on whether there is an obstacle present or not, and *ir.wait_for_obstacle()* which returns True if an obstacle is detected within 10 seconds, and False otherwise. The infrared sensor used cost R44,95.

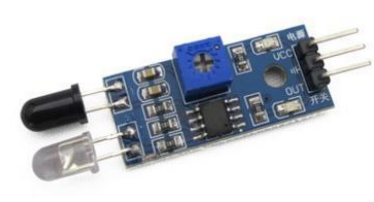

Image 1: Infrared Sensor

4.1.2 The Ultrasonic Sensor

The US sensor works by emitting a high frequency, inaudible sound pulse and timing how long it takes for the echo to be reflected back. Using the distance = speed * time equation, using the speed of sound and the time for the echo to be received, the sensor can be used to calculate the distance to the nearest obstacle.

This sensor was placed on the front of the robot, alongside the IR sensor, so that it could also be used to avoid and detect obstacles. The US sensor only has one function in the features API: *us.get_distance()* which returns a float, the distance to the nearest obstacle in centimeters. The ultrasonic sensor used cost R124,95.

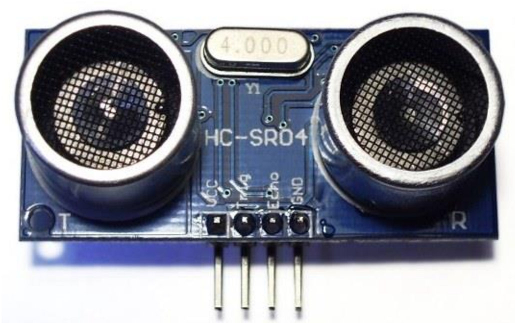

Image 2: Ultrasonic Sensor

4.1.3 The Camera

The camera used was one of the camera modules made specifically for the Raspberry Pi. The camera has two functions: *camera.take_photo()* and *camera.detect_colour()*. The *camera.take_photo()* function uses the Raspberry Pi PiCamera Python module to take a photograph and save the image as a .jpg file on the Raspberry Pi's hard drive. The *camera.detect_colour()* function takes a photograph and then uses the Python Imaging Library (PIL) to calculate the average red, green, and blue colour channel values of the image, and uses them to calculate the colour of the image, returning a string value of "red", "green", or "blue". The camera is placed on the front of the robot, facing towards the ground, so that the robot can identify the colour of the grid block which it currently occupies. The camera module used cost R489,95.

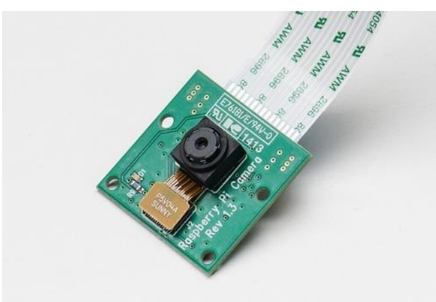

Image 3: Raspberry Pi Camera Module

4.1.4 The LEDs

There are three different colour LED lights on the robot: red, blue, and yellow, which can be turned on, off, or flashed independently. Users can access these functions via the features API by calling *led.red_on()*, *led.red_off()*, or *led.red_flash()* for each colour. The LEDs are placed in a breadboard, which is stuck to the left hand side of the robot chassis with double-sided tape. The LEDs can be bought individually, but come in packs for approximately R25,00.

Image 4: LED Light

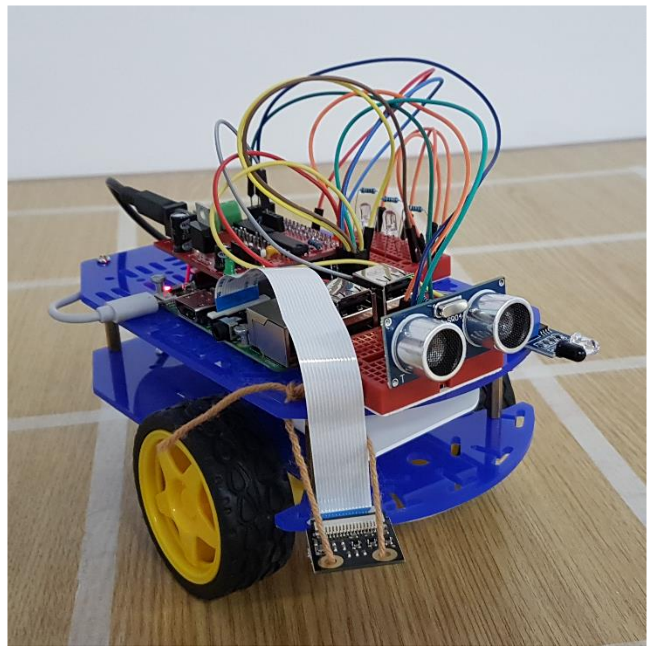

Image 5: The Final Robot

4.2 Task Design

To test the robot, a set of tasks were created. Each set of tasks uses a single feature of the robot to examine a specific Computer Science concept. The concepts that were examined were conditionals (*if*, *elif* and *else* statements), for loops, while loops, and functions. These concepts were chosen to correspond with the concepts taught in the previous years extended degree programme's tutorial sessions. More concepts were taught throughout the year, but due to time constraints, only four concepts could be tested. Each task was designed to be completed by a first year Computer Science student within 10 minutes and was focused around demonstrating the capabilities and uses of each feature. The tasks were as follows:

4.2.1 Conditionals

4.2.1.1 Infrared Sensor

Write a program that will move the robot forward 3 blocks and then use the Infrared sensor to check for an obstacle. If an obstacle is detected, turn right 2 steps and move forward 1 block otherwise, reverse 1 block.

4.2.1.2 LEDs

Write a program that will move the robot to position (4,5) in the grid, and then generate a random number in the range [1,3]. If the number is 1, move the robot forward 1 block and flash the red LED light. If the number is 2, leave the robot where it is, and flash the blue LED light. If the number is 3, make the robot reverse 1 block and flash the yellow LED light.

4.2.1.3 Ultrasonic Sensor

Write a program that will use the Ultrasonic sensor to detect the distance to the nearest object. If the object is further than 25cm, move forward 1 block. If the object is closer than 25cm, turn right 2 steps and move forward 1 block.

4.2.1.4 Camera

Write a program that will pathfind to the following positions: [0,3], [1,5], and [4,5]. At each of these positions, use the camera to take a photo of the grid block and then detect the colour of the photo. If the photo is red, print "This is red!" to the terminal, if the photo is green, print "This is green!" and if the photo is blue, print "This is blue!".

Pathfinding hint: follow_path(pathfind([get_pos()], [goal x, goal y]))

4.2.2 For Loops

4.2.2.1 Infrared Sensor

Write a program that will loop 4 times. At each iteration:

- Use the Infrared sensor to check for an obstacle
- If you detect an obstacle, reverse 1 block
- Otherwise, if there is no obstacle, move forward 1 block
- Finally, wait for one second before continuing on to the next iteration.

Once the loop has finished, if an obstacle has been detected at any time, reverse 1 block, and if not, move forward one block.

Hint: Use time.sleep(1) to insert a 1 second wait (don't forget to 'import time')

4.2.2.2 LEDs

Write a program that will loop 4 times. At each iteration, move the robot forward, turn right 2 steps, and flash the LEDs in the order red -> blue -> yellow.

4.2.2.3 Ultrasonic Sensor

Write a program that will loop 5 times. At each iteration use the Ultrasonic sensor to calculate the distance to the nearest object. Once all measurements have collected, calculate the average. If the average is greater than 30cm, move forward one block, otherwise reverse one block.

4.2.2.4 Camera

Write a program that will loop 3 times. At each iteration, use the camera to take a photo and then detect the colour of the photo. Print the colour to the terminal, and then move the robot forward one block.

4.2.3 While Loops

4.2.3.1 Infrared Sensor

Write a program that will use a while loop. At each iteration, move the robot 1 block forward and use the Infrared sensor to detect obstacles. If an obstacle is detected, turn the robot 2 steps to the left. Once 2 obstacles have been detected, terminate the loop.

4.2.3.2 LEDs

Write a program that will use a while loop. At each iteration, generate a random number in the range [1, 3]. If the number is 1, flash the red LED, if the number is 2, flash the blue LED, if the number is 3, flash the yellow LED. Terminate the loop once the red LED has been flashed 5 times.

4.2.3.3 Ultrasonic Sensor

Write a program that will use a while loop. At each iteration, move the robot 1 block forward and use the Ultrasonic sensor to calculate the distance to the nearest obstacle. If the obstacle is closer than 30cm, terminate the loop.

4.2.3.4 Camera

Write a program that will use a while loop. At each iteration, move the robot 1 block forward, use the camera to take a photo, and then detect the colour of the photo, printing the colour to the terminal each time. Once a red block has been detected, terminate the loop.

4.2.4 Functions

4.2.4.1 Infrared Sensor

Write a function checkSafe(x) which will take an integer x as a parameter. The function must use the Infrared sensor to check for obstacles, wait for x seconds, and then check again. If an obstacle was detected both times, make the robot reverse one block, otherwise move forward one block.

4.2.4.2 LEDs

Write a function lightShow() that will turn the LEDs on and off in a specific order. The order is entirely up to you, just try to keep your function under 25 lines of code.

4.2.4.3 Ultrasonic Sensor

Write a function calcBlocks(x) which takes a float x as a parameter. Your program must measure distance measurement to the nearest obstacle using the Ultrasonic Sensor, and then pass the value to the function, which will then calculate the number of blocks to this obstacle, and then move forward that many blocks.

Hint: Each block is 20cm x 20cm big

4.2.4.4 Camera

Write a function colourMove(x) which takes a string parameter. Your program must use the camera to detect the colour of the current block, and then pass this colour to the function to decide which action to take. If the image is red, move forward one block, if the image is blue, turn right 2 steps, and if the image is green, reverse one block.

5. Testing Metrics

Two surveys were used to test and evaluate the features side of the robot platform. The first survey is a System Usability Scale (SUS) questionnaire [15]. This questionnaire assesses the usability of a system using ten Likert scale questions, and was used to evaluate the usability of the features API and the documentation available on the wiki. The second survey was created specifically for this project. The survey asks the user to rate each feature on a five point Likert scale in terms of how well that feature taught the Computer Science concept being tested in relation to the other features. The final two questions ask the user to pick which feature they most enjoyed using, and which feature best demonstrated the concept being tested. Both of these surveys are attached under Appendix A at the end of this paper.

6. Methods

The following section will detail the participants of this study, the testing environment and the procedure followed while testing.

6.1 Participants

The participants of the study consisted of two Masters students and five Honours students from the University of Cape Town's Computer Science Department. The planned participants of the study were the targeted end users of the robot platform, class members of the current Computer Science extended degree programme course. Unfortunately this plan was unfeasible. At the time of testing, students were protesting on campus, resulting in the closure of campus for three weeks, and conditions on campus being uncertain for the weeks thereafter. As a result of this, testing was performed with expert users, rather than end users. The two Masters students were recruited by our project supervisor, Gary Stewart, and the remaining Honours students were recruited via social media.

6.2 Testing Environment

The testing environment was a corner in the Honours Laboratory on the University of Cape Town's Upper Campus. Two tables were pushed together to make a surface for the robot. Using masking tape, the tables were divided into a 6x8 block grid, with each block being 20x20 centimeters in size. Six coloured pages were placed on the grid at different positions: two green, two blue, two red pages. The coloured pages were placed to coincide with the tasks that the users were expected to complete. The Raspberry Pi responsible for hosting the web server and capturing the live video was stuck to a wall, overlooking the tables. A table and chair was placed next to the grid table for participants to sit and work.

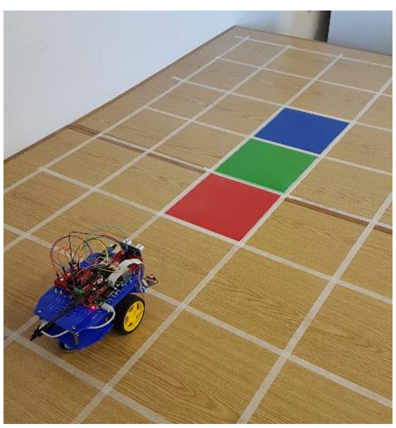

Image 6: Grid Setup

6.3 Testing Procedure

Testing was performed in two phases. At the beginning of the first testing session, the participant was greeted and made to sit in front of a laptop on the table. The participant was briefed on what was expected of them (that they would have to complete a series of tasks using the robot platform), and were then given a consent form to sign. They were then made to fill out a pre-survey, as per the needs of a group member's research. Once complete, they were required to sign up and acquaint themselves with the web app. When they were ready to proceed, they were provided with a series of tasks, each one using each feature to test a specific Computer Science concept. For the tasks that required obstacles, a box was placed on the grid in the necessary positions. Once the tasks were complete, the participants were made to fill out the remaining surveys and questionnaires. Finally, they were asked if they had any general comments about the platform, and afterwards they were thanked for their participation and allowed to leave.

After the first testing phase was complete, the feedback gathered from students was reviewed. The platform was then changed and improved based on this feedback before the second phase of testing began. The second phase of testing took the same form as the first, with the same participants (minus one Masters student) who then used the improved platform and completing a set of tasks testing a different Computer Science concept.

Throughout both testing phases, notes were taken to document any general comments or problem areas that the participants experienced. Because learnability was not being tested, we were able to interact with and help the participants if they ran into any difficulties.

7. Results

The following section reports the results of the usability survey as well as the participants' rankings of how well each robot feature taught the various Computer Science concepts.

7.1 Usability Results

The usability survey results were recorded and then scored following a defined method [15] to give a score out of 100. The usability scores after the first phase of testing was 84.6, and increased to 87.9 after the second phase of testing, indicating an increase in usability, however this was not proven to be statistically significant.

7.2 Feature and Concept Results

To analyse the results of the various features performance for each Computer Science concept, the modal and median scores for each feature were used, and not the mean. This is because the data is ordinal, and the values simply represent a "*greater than*" or "*less than*" relationship. Since there are so few participants in each response group ($n <= 4$), very few statistical tests can be performed and receive reliable results.

7.2.1 Conditionals

Three participants completed the series of tasks to test conditionals. The infrared and ultrasonic sensors both had a median score of 4, while the camera and LED both had a median score of 3. The only feature to have a modal score was the ultrasonic sensor, with a score of 4. Two participants indicated that they enjoyed using the infrared sensor the most, and the other participant preferred the LEDs. The same participants indicated that the same feature they most enjoyed using also best demonstrated the concept, in their opinion.

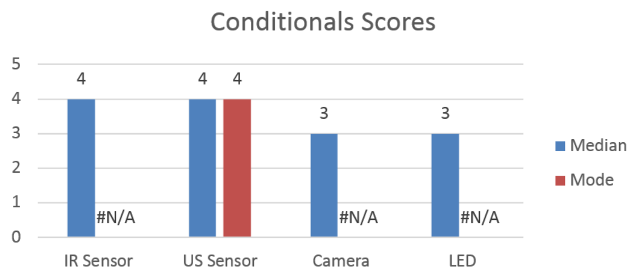

Figure 2: Feature ratings for Conditionals

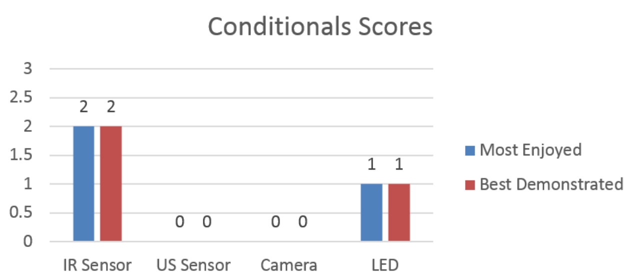

Figure 2: Best features for Conditionals

7.2.2 For Loops

Similar to conditionals, three participants completed the series of tasks to test for loops. The infrared sensor and camera received a median score of 4, the LEDs a median score of 3, and the ultrasonic sensor a median score of 2. The only two features to have a modal score were the camera, with a score of 4, and the LEDs, with a score of 3. Two participants indicated that they most enjoyed using the infrared sensor, and thought the infrared sensor best demonstrated the concept, while the remaining participant indicated that they most enjoyed using the LEDs, but felt that the camera best demonstrated the concept.

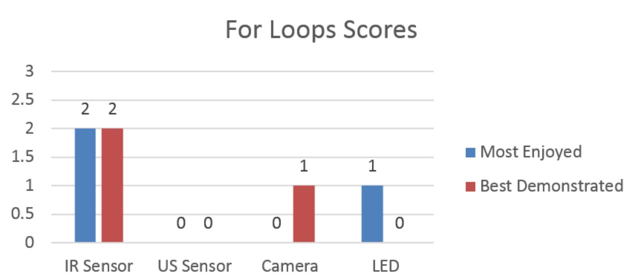

Figure 3: Feature ratings for For Loops

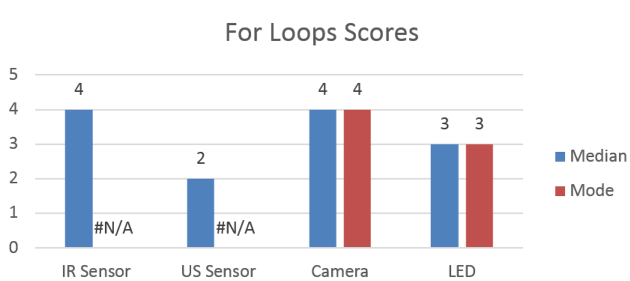

Figure 4: Best features for For Loops

7.2.3 While Loops

To test while loops, four participants completed the series of tasks. From the responses, the median score for infrared sensor was 4, the LEDs 3.5, and the ultrasonic sensor and camera both received a score of 3. The LEDs were the only feature to not have a modal score, while the infrared sensor received a score of 5, and the ultrasonic sensor and camera both received a score of 3. One participant indicated that they most enjoyed using the ultrasonic sensor, and that it best demonstrated the concept. The remaining three participants indicated that they most enjoyed using the infrared sensor, but only one of them indicated that they thought it best demonstrated the concept. The remaining two thought the LEDs best demonstrated the concept.

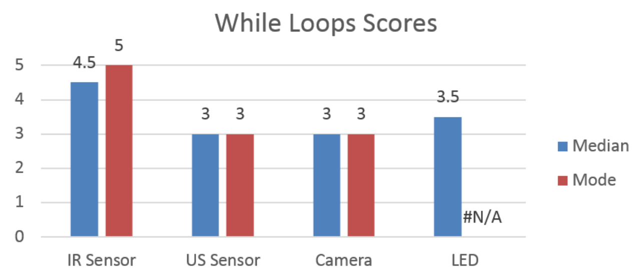

Figure 5: Feature ratings for While Loops

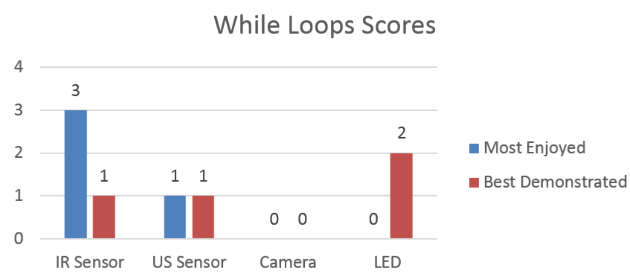

Figure 6: Best features for While Loops

7.2.4 Functions

Three participants completed the series of tasks to test their knowledge of functions. The ultrasonic sensor and the camera both received a median score of 4, the infrared sensor a score of 3, and the LEDs a score of 2. The infrared sensor did not have a modal score, however the camera and the ultrasonic sensor received a modal score of 4, and the LED a modal score of 2. All three students indicated that they most enjoyed using the ultrasonic sensor, and two of them thought it best demonstrated the concept, with the remaining student indicating that they thought the camera was better.

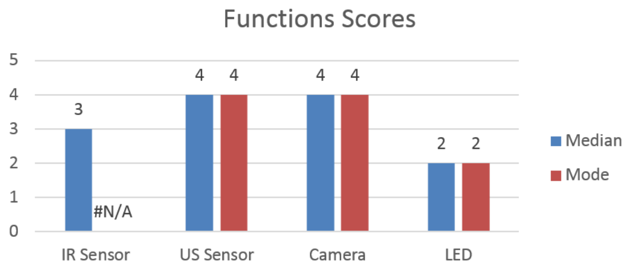

Figure 7: Feature ratings for Functions

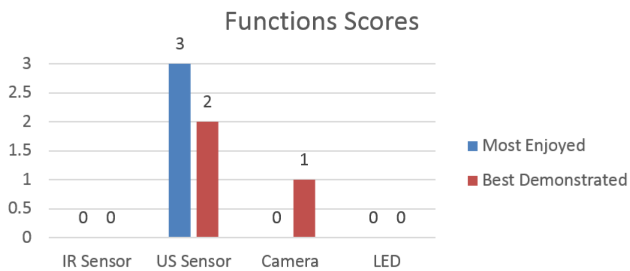

Figure 8: Best features for Functions

8. Analysis

8.1 Usability

The results of the usability survey do show a slight increase in the usability score. This was to be expected, as the second phase of testing had incorporated feedback gathered during the first phase. While the difference in usability score is rather low between the two phases, the participants did acknowledge that the platform was "much better" than it was previously. Qualitatively, most students felt that after using the platform during the second testing phase was far more intuitive and easy, but also stated that they had not had much difficulty during the first phase of testing either.

8.2 Features and Concepts

8.2.1 Conditionals

For the conditionals results, there are two pairs of features with equal ratings: the infrared and ultrasonic sensor, and the camera and LEDs. While this does put the two sensors slightly ahead of the camera and the LEDs, the results of the feature which participants most enjoyed using and best demonstrates the concept of Conditionals seems to favour the infrared sensor. When participants were asked to provide a reason for their choice of the infrared sensor, they replied that they valued the simplicity of the sensor, that they could easily understand how it worked. This is ideal, as the infrared sensor is one of the least expensive features examined.

8.2.2 For Loops

The for loops results showed two leading features, the infrared sensor and the camera. These two features were tied with median score ratings of 4, however the camera did had a modal score of 4, while the infrared sensor did not have a modal score. This means that the camera was consistently rated higher than the infrared sensor. While this provides evidence that the camera may be the better feature, two of the participants still indicated that they most enjoyed using the infrared sensor, and that it best demonstrated the concept. Only one participant felt that the camera best demonstrated the concept, but also indicated that they most enjoyed using the LEDs. Based on these results, it can be argued that the infrared sensor or the camera could be a suitable candidate for teaching the concept of for loops. The camera is, however, significantly more expensive than the infrared sensor.

8.2.3 While Loops

The while loops results yield some interesting results. The infrared sensor is rated the best out of all the other features, with a median score of 4.5, and a modal score of 5, with the LEDs following behind with a median score of 3.5. Of the feedback, three out of the four participants indicated that they most enjoyed using the infrared sensor, but only one felt that it best demonstrated the concept. The feature that most participants thought demonstrated the concept best were the LEDs. These results could be used differently in various situations. If the goal of an educator is to increase the motivation of the students, they may decide to use an infrared sensor, while if the goal was to teach the concept as effectively as possible, the LEDs may be a better choice. With respect to cost, the LEDs can be purchased individually for a lower price, but the price of the infrared sensor is still affordable even on a low budget.

8.2.4 Functions

For the functions results, the ultrasonic sensor and the camera were tied, with both median and modal scores of 4 each. This shows that participants consistently rated both features highly. However, when asked which feature they most enjoyed using, the ultrasonic sensor received all the votes. The ultrasonic sensor also received two votes for being the feature that best demonstrated the concept, while the camera only received one. When asked why, the participants replied that the ultrasonic sensor was simpler, more intuitive, and more reliable to use than the camera. This may be due to the fact that the camera's ability to detect the colour of an image is reliant on the current lighting conditions when the image is captured. In certain conditions, colours could be incorrectly identified, resulting in unexpected behavior of the participants programs. Taking this into account, the ultrasonic sensor seems to be the superior feature for this concept. In addition to the reliability of the ultrasonic sensor, it is also significantly cheaper than the camera.

9. Conclusions

While in-depth statistical analysis of these results could not be performed due to the low number of participants, there are some clear preferences in the results. Of the four Computer Science concepts that were investigated (conditionals, for loops, while loops, and functions), one or two features can be identified which can best be used to teach the concept. For teaching the concept of conditionals, the best feature appears to be the infrared sensor. Participants valued the simplicity of the feature, making the sensor a very cost effective choice.

For teaching the concept of for loops, participants showed interest in both the infrared sensor and the camera. While both of these features can be used together, the camera is significantly more expensive than the infrared sensor, and would be the less cost effective purchase.

The concept of while loops indicated two features as possible candidates: the infrared sensor, and the LEDs. This is ideal, as the LEDs are the least expensive feature, followed closely by the infrared sensor. If the focus of an educator is to teach while loops in particular, the cost effective choice would be the LEDs, however, the infrared sensor has been consistently rated higher across multiple concepts than the LEDs, and would make an overall more versatile feature.

For teaching the concept of functions, participants identified the ultrasonic sensor and the camera as being worthy candidates. While the ratings for these two features were close, it can be concluded that the ultrasonic sensor is a better choice. This is not only because the ultrasonic sensor is significantly less expensive, but also due to the fact that it is far more reliable. The performance of the camera can vary widely depending on current lighting conditions, which may lead to confusion or frustration of the users.

10. Problems, Challenges, and Limitations

This section describes some of the problems experienced during the creation of the robot platform, as well as the problems faced and the limitations of the study as a whole.

10.1 Problems

There were only two problems identified. The first, as mentioned before, was that the performance of the camera depended on the lighting conditions in the room. In darker lighting conditions, the camera would sometimes incorrectly identify the lighter colour red/green block as a darker colour blue block. This problem can be mitigated by placing the robot in an environment where the lighting conditions are not likely to change.

Another problem was that the wireless internet signal boosters placed all around the University of Cape Town's various buildings produced incorrect readings on the infrared sensor. Whenever the infrared sensor is pointed towards one of the signal boosters, it falsely detects that there is an object. This problem was mitigated by designing the tasks such that whenever the infrared sensor was used, the robot was always in a position that faced away from the signal boosters.

10.2 Challenges

The only significant challenge arose when working with the ultrasonic sensor. Having next to no experience with electronics prior to this project, troubleshooting issues with the various features was difficult. After days of struggling with the ultrasonic sensor, it was tested by an engineer and found to be faulty. A replacement ultrasonic sensor was found soon after, and worked as intended throughout the duration of the project.

10.3 Limitations

There were several limitations of this study. With only access to one robot, and each testing session taking approximately one hour to complete, only seven participants were tested per testing phase. This low number of participants were then further divided between the Computer Science concepts being investigated, lowering the number of participants in each group even more. As a result of this, only four concepts were investigated, and the effect of the order in which the features were used could not be investigated at all.

There were also several limitations with respect to the hardware used. There are only a limited number of available pins on the Raspberry Pi after the motor board is plugged in, which puts a limit on the number of features that could be used. The Raspberry Pi and motor board also took up a lot of space on the robot chassis, limiting the space where additional features could be placed.

11. Future Work

The future work of this study would definitely involve testing with more participants. This would allow more reliable information to be gathered, as well as other variables such as order effects to be efficiently analysed. Along with more participants, a wider array of features and concepts could also be investigated. Possible features include light sensors, temperature sensors, accelerometers, and buzzers. Other concepts to be investigated could be recursion, data structures, and file input/output. Future work could also include multiple sets of tasks for each concept, to eliminate any bias of one task being more enjoyable than another, resulting in compromised results.

12. Acknowledgements

I would like to acknowledge my project supervisor, Gary Stewart, my second reader, Maria Keet, and my group members, Josh di Bona and Muhummad Patel, it has been a pleasure being able to work so closely with you. I would also like to acknowledge all the students who volunteered to be test subjects, this project would not have been possible without you.

13. References

[1]. FACTS, Q. (2008). Designing Personal Robots for Education: Hardware, Software, and Curriculum.

[2]. Markham, S. A., & King, K. N. (2010, June). 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* (pp. 204-208). ACM.

[3]. Lenox, T. L., Woratschek, C. R., & Davis, G. A. (2008). Exploring Declining CS/IS/IT Enrollments. *Information Systems Education Journal*, *6*(44), 1-11.

[4]. Rahul, R., Whitchurch, A., & Rao, M. (2014, December). An Open Source Graphical Robot Programming Environment in Introductory Programming Curriculum for Undergraduates. In *MOOC, Innovation and Technology in Education (MITE), 2014 IEEE International Conference on* (pp. 96-100). IEEE.

[5]. Yadin, A. (2011). Reducing the Dropout Rate in an Introductory Programming Course. *ACM inroads*, *2*(4), 71-76.

[6]. Xu, D., Blank, D., & Kumar, D. (2008, February). 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* (pp. 66-70). ACM.

[7]. Yarosh, S., & Guzdial, M. (2008). Narrating data structures: The Role of Context in CS2. *Journal on Educational Resources in Computing (JERIC)*, *7*(4), 6.

[8]. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A. & Martinoli, A. (2009). The e-puck, a Robot Designed for Education in Engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (Vol. 1, No. LIS-CONF-2009-004, pp. 59-65). IPCB: Instituto Politécnico de Castelo Branco.

[9]. Rahul, R., Whitchurch, A., & Rao, M. (2014, December). An Open Source Graphical Robot Programming Environment in Introductory Programming Curriculum for Undergraduates. In *MOOC, Innovation and Technology in Education (MITE), 2014 IEEE International Conference on* (pp. 96-100). IEEE.

[10]. Saad, A., Shuff, T., Loewen, G., & Burton, K. (2012, March). Supporting Undergraduate Computer Science Education Using Educational Robots. In *Proceedings of the 50th Annual Southeast Regional Conference* (pp. 343-344). ACM.

[11]. Markham, S. A., & King, K. N. (2010, June). 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* (pp. 204-208). ACM.

[12]. Fagin, B. S., Merkle, L. D., & Eggers, T. W. (2001). Teaching Computer Science with Robotics Using Ada/Mindstorms 2.0. *ACM SIG Ada Letters*, *21*(4), 73-78.

[13]. Summet, J., Kumar, D., O'Hara, K., Walker, D., Ni, L., Blank, D., & Balch, T. (2009, March). Personalizing CS1 with Robots. In *ACM SIGCSE Bulletin* (Vol. 41, No. 1, pp. 433-437). ACM.

[14]. Fagin, B. S., & Merkle, L. (2002). Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education. *Journal on Educational Resources in Computing (JERIC)*, *2*(4), 2.

[15]. Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4-7.

[16]. FACTS, Q. (2008). Designing Personal Robots for Education: Hardware, Software, and Curriculum.

[17]. Xu, D., Blank, D., & Kumar, D. (2008, February). 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* (pp. 66-70). ACM.

[18]. Schumacher, J., Welch, D., & Raymond, D. (2001). Teaching Introductory Programming, Problem Solving and Information Technology with Robots at West Point. In *Frontiers in Education Conference, 2001. 31st Annual* (Vol. 2, pp. F1B-2). IEEE.

[19]. Apiola, M., Lattu, M., & Pasanen, T. A. (2010, June). Creativity and Intrinsic Motivation in Computer Science Education: Experimenting with Robots. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 199-203). ACM.

14. Appendix A

14.1 Usability Survey

The following questions relate to the Features of the Robot, please rate each statement by placing a ✓ in the box of your choice.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Please write down any comments on the usability of the software or any recommendations for improvement.

14.2 Feature Rating Survey

Concept:

Please circle the appropriate number:

1. Compared to the other features, I felt the **infrared sensor** taught the concept:

Worse | The Same | Better

Worse		The Same		Better
1	2	3	4	5

2. Compared to the other features, I felt the **ultrasonic sensor** taught the concept:

Worse		The Same		Better
1	2	3	4	5

3. Compared to the other features, I felt the **camera** taught the concept:

Worse		The Same		Better
1	2	3	4	5

4. Compared to the other features, I felt the **LEDs** taught the concept:

Worse		The Same		Better
1	2	3	4	5

Please tick one feature per question:

1. Which feature do you think students will most enjoy using for this concept?	IR Sensor	US Sensor	Camera	LEDs
2. Which feature do you feel best demonstrated this concept?	IR Sensor	US Sensor	Camera	LEDs