Extreme Olympics: Curling and Maze Events

Fall 2018/2019

Cooper G. Cunningham B00805690 CSCI 1108 Sec 01 Faculty of Computer Science cp945886@dal.ca	**Christina Fisher-Bell** B00797239 CSCI 1108 Sec 01 Faculty of Computer Science ch747645@dal.ca	**Abdullah al Mukaddim** B00804319 CSCI 1108 Sec 01 Faculty of Computer Science ab843512@dal.ca	**Aoxiang Zhang** B00794930 CSCI 1108 Sec 01 Faculty of Computer Science ax706429@dal.ca

Abstract

The purpose of this project is to program a Thymio robot using the Aseba framework to successfully compete in two Robotics Olympics events. The curling event consists of two robots battling to be the closest to the black dot inside a ring. During this complicated event, the robot completes the task using its sensors, localization theory, and actuators. The Search and Rescue event involves utilizing the horizontal sensors of Thymio and Markov localization theory to determine the location of the robot. During this competition the robot also must be able to navigate throughout the maze. Our robot successfully completed the search and rescue event and barely qualified in the curling event.

Introduction

Students at Dalhousie University are competing in two different robotics challenges. The first event, the curling program requires competitors to overcome the other team in reaching a certain location on a table. Robots have only one minute to be as close as possible to the defined location than the other team. The Search and Rescue program requires robots to identify their location using deductive reasoning [2] and indications from the environment based on sensor readings [1].

The definition of robotics is the science of perceiving and manipulating the physical world through computer-controlled devices [1]. General topics utilized in both events extensively include localization, sensors and modeling, simulator and robotics environments, actuators, kinematics, and odometry, state transition diagrams, robotics overview and ASEBA intro, search, object recognition, and debugging. The two most recurring topics are localization and sensors and modeling. The robotics platform utilized for the project was Thymio, a robot possessing two ground proximity sensors and seven horizontal proximity sensors, and the programming language used was ASEBA [1]. These sensors proved extremely useful for detecting the barcodes, the black dot, the grey ring, and the black lines. The conclusions of this research will assist students in understanding robotics concepts of localization which proves itself a feature in the future applications of robotics.

In the curling event, our robot had issues with veering in the incorrect direction and falling off the table, as well as failing to read the destination ring. Our robot only managed to stop on the dot after some configurations in a non-battle version of the event. In the search and rescue event, our robot successfully read and displayed the barcodes using its LEDs and completed the maze. Both events were completed fairly effectively, though some issues were ran into.

Robotics is a field that works on the development of machines that can replace humans in various activities [3]. Development in robotics are applicable to many domains of life and as technology advances exponentially the number of robots that assist humankind are also developing at a rapid rate. The future of robotics is promising and the technology that will be developed will likely be surprisingly advanced from the perspective of individuals unlearned in the field of computer science.

Throughout the duration of the project, our project team encountered several unexpected errors that forced us to change the entire direction of our strategies. For the first event, we attempted a distance calculation performed by the robot that would make the robot stop at the middle and turn to drive directly onto the inner dot. However, while testing this, it became clear that the position in which our robot would start would greatly impact the outcome of this strategy, giving it either a 50% chance of driving off the table or a 50% chance of success. There were also bugs encountered such as the robot refusing to do a lawn mower scan correctly, the robot deviating from the desired end point, and the robot refusing to turn correctly. While testing the second event, the robot had issues reading the pattern correctly, displaying the correct number of LEDs, and traversing the maze overall.

All of these problems, minor or major, were fixed utilizing new strategies, trial and error, and the analyzation of the code for both of the olympic events [7]. These methods allowed our team to better understand the ASEBA language and the behaviour and potential functions of the robot. Both Olympic events required reconfiguration, slight or major, in order to perfect the program, but success was eventually achieved through extensive group discussion and intensive coding.

Background

Students at Dalhousie University are competing for a project in Dalhousie first year Experimental Robotics course taught by William Stone. In this competition, entitled Robot Olympics, there will be two events: Curling and Search and Rescue. The students will be using the ASEBA language to program Thymio robots. ASEBA is an event-driven programming language which functions based on information gathered from sensor readings [1]. Students model the ongoing stream of sensor information and program the robots to act in certain ways based on the events that occurs due to sensor values [1]. The Thymio robot is equipped with five proximity sensors at the front of the robot, two ground sensors and two rear proximity sensors [1]. The horizontal proximity sensors measures the distance to objects utilizing infra-red light while the ground proximity sensors measure light from the ground [1]. The Thymio robot possesses two wheels, five buttons located on the top of the device, and a port to connect it to a computer [3]. Thymio also possesses seven LED display lights surrounding the buttons [3]. Utilizing concepts such as search, object recognition, and localization, the robots' programs was coded. Many of these concepts needed to be combined in order to satisfy the criteria of performance of the robot. The teams conducting this project have previous experience in programming robots to move through an obstacle course.

The Curling Program

The Robotics Olympic Curling Event involves two robots competing to have a final location as close to the target as possible; as in the game of curling, points are earned from pieces being placed in the right location. Two robots start from the same end of the table, on opposite ends of the side. Competing teams are challenged to direct their robot to a location near the other end of the table. Our team strategized to program the robot to solve three different tasks in order. The robot must: determine which side it is on, get to the target location and after arriving at the target location the robot must dominate the spot. In case of error, the robot has also been programmed to have various recovery mechanisms [7].

To attempt to determine which side of the table the robot is located, it begins to turn left until it either encounters black tape with the ground proximity sensors or after the robot has been turning for two seconds. This algorithm determines the robots state based on the side of the table it starts on. Using this information, our robot scans the area following a certain path that will yield a high probability of entering into the target spot. After the robot has turned left, it reverses to face towards the target location at the end of the table.

To find the spot, the robot moves until just before the ideal location and searches for the black spot by scanning the entire area similar to how a lawn mower would mow a field of grass. The robot follows an algorithm moving a certain distance and then completes alternating turns to execute the 'lawn mower strategy'. The robot begins turning in a left or right direction based on the 'hit' variable which determines whether or not the robot has noticed a black line. Because our robot begins by turning left whether or not the robot has hit a black line, it will determine which side of the table it has started on. The robot will begin lawn mowing to the right if starting on the left side of the table and begin lawn mowing to the left if starting on the right side of the table. The strategy was chosen to ensure that, eventually, our robot would have a greater chance of arriving at the correct location. However, the robot may not be as quick as other robots that take an approach with a higher degree of risk. The strategy behind this solution was determined with the goal of eliminating as many sources of error as possible to create the most consistent results.

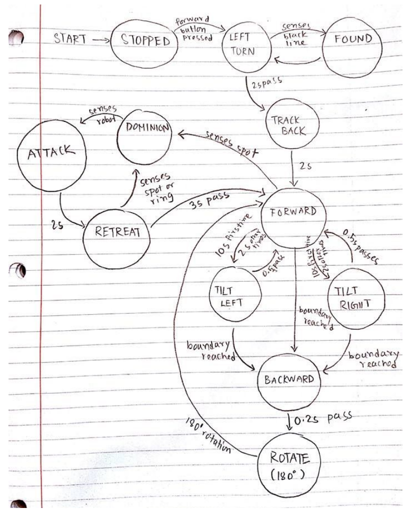
Once the robot has entered the target location, it rotates in circles to allow the horizontal proximity sensors to identify if an enemy robot is approaching. We call this the 'domination state'. If the robot approaches the threshold value, our robot will attempt to push the other robot off by accelerating forward. The robot will then retreat to the spot after two seconds of pushing. It will dominate the spot after the grey ring is sensed. In testing, we found that, if our robot is able to enter into the target location initially, was successful in defending against enemies attempting to enter into the target location. Though our solution takes a long time, the safety it provides will give our team a competitive edge due to other robots' errors and inability to deal with our robot's algorithms.

In case of unexpected issues, we have developed and implemented various recovery mechanisms in both programs. If our robot is unable to refind the circle after attempting to push the opposing enemy's robot out, our robot will begin to search for the spot by performing a lawn mower scan. However, we are unable to determine in which direction the robot should lawn mow as we do not have means to determine the robot's location on the course. This recovery mechanism gives us approximately a coin flip probability of arriving back at the target location after leaving it.

An error has been identified as our robot is unable to read a consistent difference between values of the external boundary of the course and the boundary to the target location. Our robot, when encountering the boundary of the target circle, would begin to turn 180 degrees and start lawn mowing in the opposite direction as how it is intended to respond in the case of a proximity sensor reading of the external boundary. This feature was omitted from our program because it was interfering with the robot's ability to enter into the target location, which is of the highest necessity. This choice was made to best program the robot using the equipment that we had access to. The robot's sensors do not always produce accurate results and the programmer must work with the robot to produce the best result possible.

There is a possibility that the competitor robot will interfere with our path and in this case we must equip our robot with strategies to deal with this situation. If we are outside of the circle while the other robot is inside, then we will attempt to push them out and get in as close as possible if we are unable to completely. If we notice the other robot is in our proximity outside of the circle, then our robot is programmed to stop until the other robot has cleared our way and then continue moving. There may be an issue with our solution if the other robot is able to get into the location faster and is able to keep themselves in the correct location despite our robot's efforts at moving the opponent out.

Reference Appendix A to view the program.



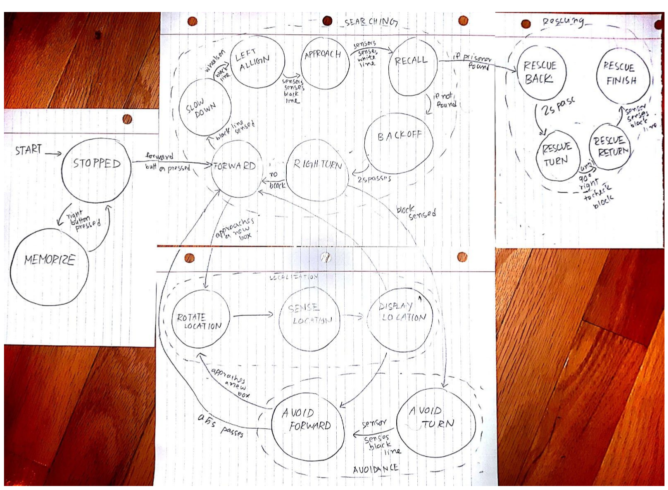
The Search and Rescue Program

The Search and Rescue program requires a robot to navigate a maze and recognize its location using sensor readings from various walls labelled with color combinations. The robot initially begins in a stopped state until it is placed into a block of the maze. It attempts to identify its location within the maze and if it's facing a wall orientation. The robot chooses to move in a certain direction based on what it determines from these sensor values. It approaches until it is one centimeter from the block and analyzes the pattern. The robot's algorithms predict its location based on the patterns. If the robot is unable to find the block, it will reverse, turn right, and move forward until it senses a barcode. If the robot senses a block without pattern, it attempts to reorient itself by directing its horizontal sensors towards other sides of the block.

The following text is intended to guide the reader to understand the various states described in the state transition diagram. Dotted lines signify the most important elements of the state transition diagram. The memorize state holds an array named prisoner, which stores the LED values for each color of the prisoner pattern, which holds the RGB values ex: white (32,32,32). When the right button is pressed, the Thymio enters into the memorize state. There is a group of search states. In order, we: search, localize, avoid a block if one is present, and rescue. In the search state, we move forward. If we detect a black line, the robot slows down until its wheels are on the line. It then aligns left to be parallel with the line. The robot approaches the pattern until it enters the white space directly outside of the box. The robot enters the recall state in which it identifies the barcode and determines if the prisoner state is equivalent to the barcode. After the recall state, the robot enters into the back off state or rescue. It enters the rescue state if the prisoner is found. Else, the robot continues to search the patterns. If a block is sensed, the robot enters into the avoidance group of states to avoid the obstacle. If a block is not sensed, the robot continues in the forward state. If the robot senses a block, it turns right to align with the black line. After, the robot enters the avoid forward state due to being continuously on the black line. The traditional forward state will trigger the robot to slow. The robot then enters into the rescue group of states. The rescue state is entered when the prisoner is identified and the robot is then confronted with the task of entering the center of the maze. The robot turns 90 degrees and searches for a block, then re-aligning itself with the line and reversing into the center of the maze. The robot, finally, checks for a block to determine its location.

We chose this solution to maintain consistency. Our solution possessed a more challenging element as the robot attempts to align itself with the black line. The solution, however, was ineffective as the robot had difficulty aligning with the line since we were limited to one ground sensor. Issues emerged in our solution due to localization problems. If our robot began in a light grey block, it would complete the event without difficulty. However, if we started on a dark grey block, our robot would not conduct itself properly. Within the simulator, the robot would conduct itself properly because there were no issues with sensors and accurate readings. We could have created models to better account for issues in the reality of experimental robotics. Our simulator was not entirely accurate to the model that was used in the class on competition day. If the arena had a greater gap between the pattern and the block, our program would have operated far more efficiently. Our team had issues as we constantly had to adjust the arena that we were working on, causing a great amount of wasted time and energy for the team to create an effective prototype. Yet, our solution gives the perfect color output if the left align state operates effectively. Our program can make perfect ¼ turns and is extremely accurate in pattern recognition. The solution was highly effective for prisoner rescue. The program was able to complete the competition in far less time then is provided for the robot.

Reference Appendix B to view the program.



Results

Based on the performance of our robot in both competitive Olympic events, it can be concluded that the team could have greatly utilized more time to further develop the robot's programs. However, for the most part, the robot performed exceptionally within the two Olympic events.

In the curling Olympic event, our robot encountered issues in its performance, especially in comparison to other groups. Even though the robot identifies which side it's on, it may turn the incorrect way and fall off the table. Our robot's ground proximity sensors may have had too low of a threshold that allowed it to have a difficult time detecting the black dot or grey ring. Our robot did not have an effective defensive system, concluding with it getting amptly pushed off the paper surface. Our robot ultimately had issues identifying the side in which it was on and also issues with finding the black dot. When not battling another robot, and after a couple of adjustments, our robot completed the curling event successfully and with ease. However, other groups had much success in battling with another robot, where ours did not. In comparison to these groups, our robot failed.

In the search and rescue Olympic event, our robot correctly read the barcodes and displayed the right order and number of LEDs directly afterwards. Our program's ability to map the robot's current locations allowed for a successful event. However, our robot completed the event rather slowly in conjunction with the others, though it proved necessary for completion. Overall, our robot performed spectacularly given the coding hiccups encountered during the event. Altogether, the robot performed exceptionally when compared to a majority of other robots who could not manage to detect the barcodes, showcase the LEDs, and travel competently around the maze. The search and rescue Olympic event proved highly successful when paralleled with expectations and the performance of our competitors.

Conclusion and Future Work

If given a wider timeframe to complete and flesh out our programs extensively, we would have changed various aspects within our code.

For the curling Olympic event, we would have allocated more time to testing our strategy for the event. Ultimately, we would have completely reconstructed our original strategy and devised a new plan of attack that would be more effective. Perhaps we would include more timers and a lawn mower scanning attempt if given more time to reflect upon our decisions. Important aspects of our solutions included initial localization, timers, and proximity sensor capabilities.

For the search and rescue Olympic event, we would have developed the code further by slightly increasing the speed to enhance the robot's efficiency. The team would have also strategically planned the program more according the the initial position of the robot. Our team would have taken the opportunity to allocate an increased amount of time to consult with the TAs to ensure that or robot was prepared to start the event in the correct orientation. Important aspects of the solution for the search and rescue Olympic event was the localization strategies, the barcode reading abilities, and the LED displays.

In conclusion, extra allocated time would allow our team to be complete the given events faster, more efficiently, and more effectively.

References

[1] Stone, William. *Sensors and Modeling*. Dalhousie University, 2018. Accessed 22 November 2018.

[2] Stone, William. *Localization*. Dalhousie University, 2018. Accessed 22 November 2018.

[3] Stone, William. *Actuators, Kinematic and Odometry*. Dalhousie University, 2018. Accessed 24 November 2018.

[4] Stone, William. *Object recognition*. Dalhousie University, 2018. Accessed 26 November 2018.

[5] Stone, William. *Search*. Dalhousie University, 2018. Accessed 26 November 2018.

[6] Stone, William. *Report Writing*. Dalhousie University, 2018. Accessed 8 November 2018.

[7] Stone, William. *Debugging.* Dalhousie University, 2018. Accessed 8 November 2018.

Appendices

Appendix A

Event #1

https://docs.google.com/document/d/1Wgc1MtvxbNYzZ3UcUb5pc9YMxtrR7LvoIGdDx-_iudk/edit?usp=sharing

Appendix B

Event #2

https://docs.google.com/document/d/14oktbV4e1HaxBdwzO_czZ-pVUt5NCp39HrFOK9HwUQo/edit?usp=sharing