Obstacle Course Challenge

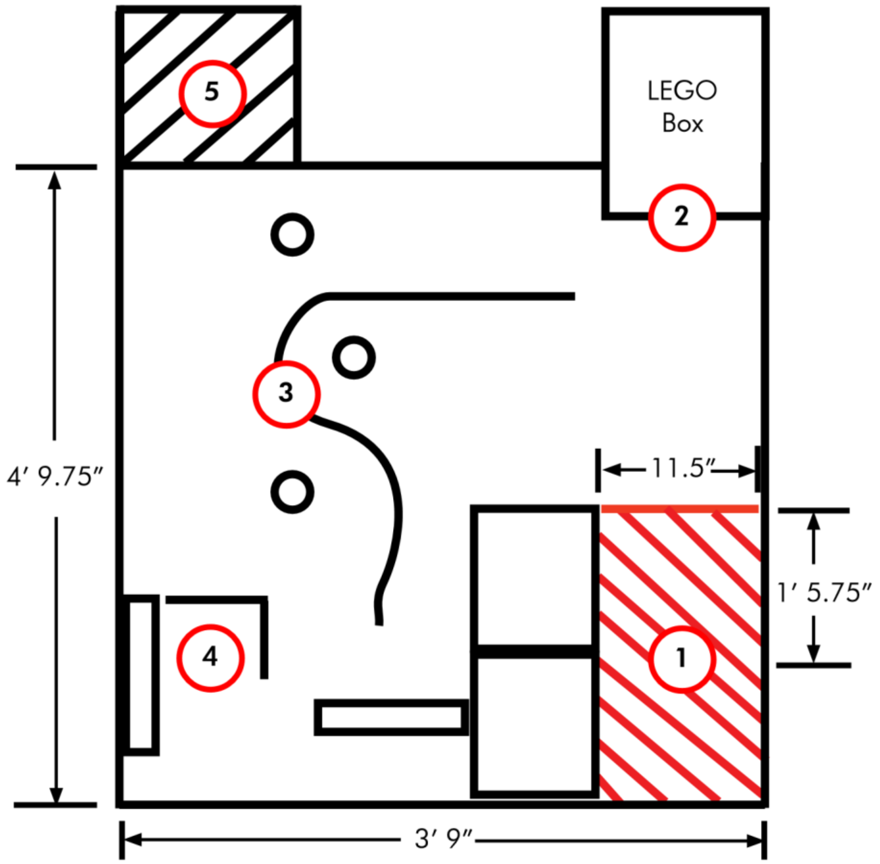
Challenge Description

This challenge includes obstacles which forces students to use sensors in order to get to the goal area. The robot must first begin at the starting area, touch the wall, follow a line, get to the calibration area using the Sonar, Touch, or Light sensor, and finally getting to the goal.

Materials Needed

- Black electrical tape
- Red electrical tape
- Scissors (or cutting tool)
- Ruler (or straight edge)
- 3 Styrofoam cups
- 4 Books
- 1 LEGO Box container

Board Specifications



Note: Diagrams are not drawn to scale

1 Starting area.

2 Touch the box.

3 Follow the line while avoiding obstacles.

4 Get to this calibration area.

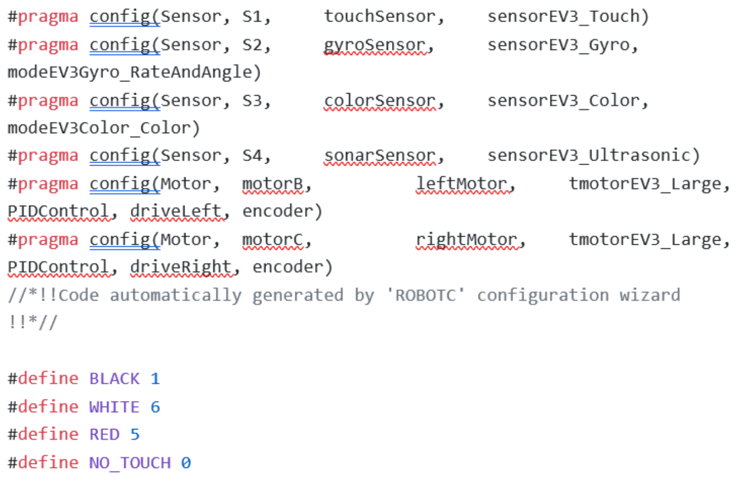
5 Goal area.

COMP20170 Introduction to Robotics Challenge 1

Group Journal

24/01/20

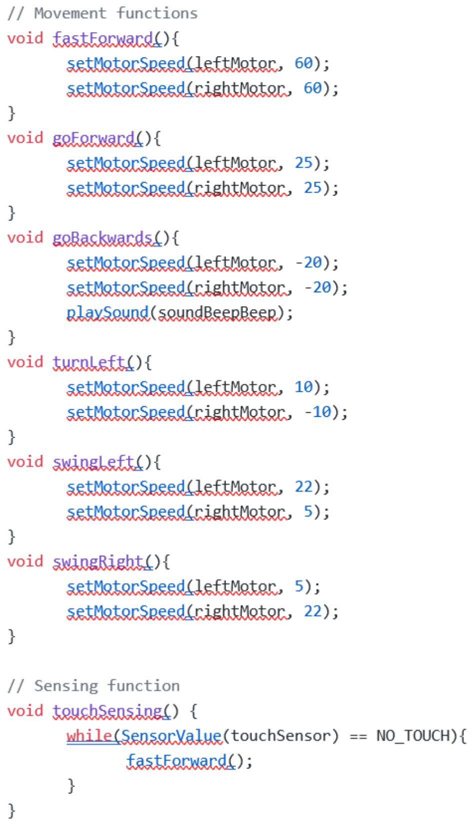
- Initially we had issues installing the firmware to the robot because conflicting programs were running. John eventually figured out the solution and resolved it the issue.
- John, Peter and Matthew built Derek (we named it). Conor and Finbar worked on installing firmware and software on the laptops we were using.
- ROBOTC was also installed on a Windows virtual machine running in Linux.
- When assembly: we connected two motors to EV3 and wheels, with a ball bearing on the front.
- Next, we attempted different variants of the sample code and worked out that turning the robot based on speed of motors (50 -50) rather than sleep time.
- We experimented with loading in custom sound effects etc.
- A while loop was used to repeat behaviour.
- We played around with attaching an arm to the robot to swing at the end of its movements to give Derek offensive capabilities.
- Conor took notes and logs of the progress of the robot to that point.



```
#pragma config(Sensor, S1,     touchSensor,    sensorEV3_Touch)
#pragma config(Sensor, S2,     gyroSensor,     sensorEV3_Gyro,
modeEV3Gyro_RateAndAngle)
#pragma config(Sensor, S3,     colorSensor,    sensorEV3_Color,
modeEV3Color_Color)
#pragma config(Sensor, S4,     sonarSensor,    sensorEV3_Ultrasonic)
#pragma config(Motor,  motorB,          leftMotor,     tmotorEV3_Large,
PIDControl, driveLeft, encoder)
#pragma config(Motor,  motorC,          rightMotor,    tmotorEV3_Large,
PIDControl, driveRight, encoder)
//*!!Code automatically generated by 'ROBOTC' configuration wizard
!!*//

#define BLACK 1
#define WHITE 6
#define RED 5
#define NO_TOUCH 0
```

31/01/20

- After assembling Derek last week, we added the more important parts today such as the sonar sensor, the touch sensor and the colour sensor.
- Daniel, John and Finbar went through the available Lego parts and collectively thought of the best ways to tackle attaching each sensor.
- Conor worked on adjusting and perfecting the code.
- It was important for every sensor to be in the best possible position on the robot to avoid any errors from physical obstruction of sensors. For example, we attached the sonar sensor first and we ran into an issue with the placement of the colour sensor.
- The touch sensor had to be beyond the sonar sensor, and it had to be close enough to the body to not compromise the reaction time of the robot. We found that any way we tried to attach it; the touch sensor would obstruct the sonar sensor as a result.

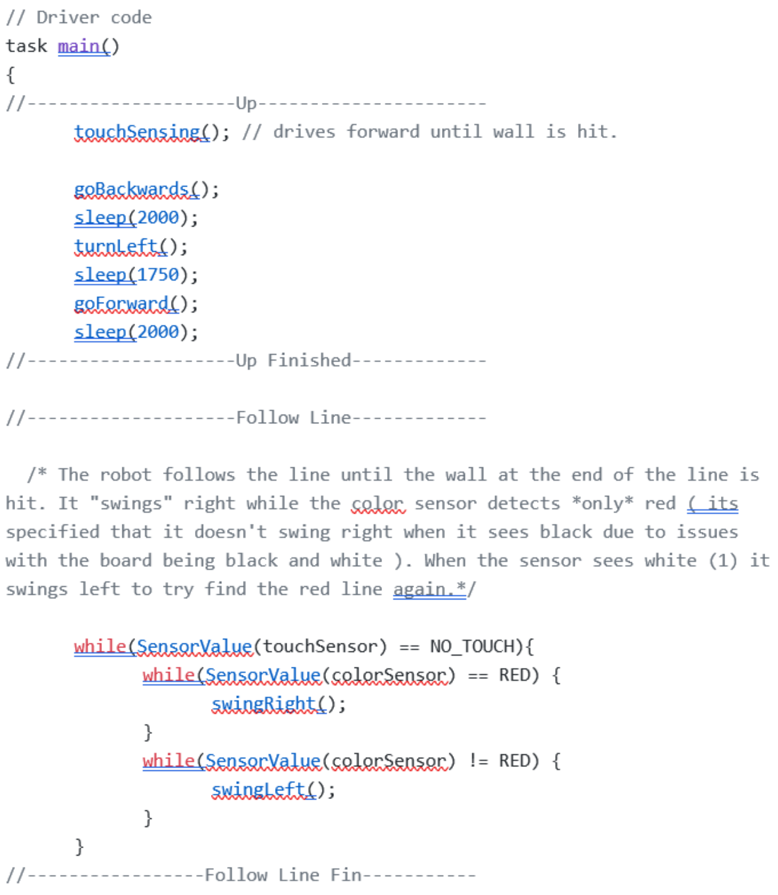


```
// Movement functions
void fastForward(){
	setMotorSpeed(leftMotor, 60);
	setMotorSpeed(rightMotor, 60);
}
void goForward(){
	setMotorSpeed(leftMotor, 25);
	setMotorSpeed(rightMotor, 25);
}
void goBackwards(){
	setMotorSpeed(leftMotor, -20);
	setMotorSpeed(rightMotor, -20);
	playSound(soundBeepBeep);
}
void turnLeft(){
	setMotorSpeed(leftMotor, 10);
	setMotorSpeed(rightMotor, -10);
}
void swingLeft(){
	setMotorSpeed(leftMotor, 22);
	setMotorSpeed(rightMotor, 5);
}
void swingRight(){
	setMotorSpeed(leftMotor, 5);
	setMotorSpeed(rightMotor, 22);
}

// Sensing function
void touchSensing() {
	while(SensorValue(touchSensor) == NO_TOUCH){
		fastForward();
	}
}
```

- We thought about solutions to resolve this issue and decided that extending arms out, slightly further than the sonar sensor, would be a better way to attach the touch sensor while also not affecting the reaction time of the robot's time.
- Setting up a mini trail on our table using the same red tape, we tested out the code that had been written.
- The robot followed the red line very well on the first try but then gave up towards the end and nearly fell off the table.
- By tweaking the code, we adjusted the robot to make sure it followed the red tape for the full length.
- Derek wobbled a lot on the second attempt, so we thought the code that determined his rotations were probably needed to be reduced.
- Daniel took down notes and logs of what the robot was doing and what was needed to improve and then wrote more detailed notes.

4/02/20

- John, Peter, Finbar and Conor met up at 9am and tried lots and lots of different things and nothing worked for the first while.
- We attempted to use an arc as a work around for the issue of Derek not following the red tape properly
- Once it would go off the line it would turn in an arc that kept getting bigger until it found the red line, it turned out to be very inefficient as it would've taken a very long time for it to successfully correct itself.
- Multiple loops with a break + without
- Adjusting position of sensor
- Daniel joined the group and we put Derek on the course and tried time after time again to get him to follow the line in its entirety
- Daniel took notes and logs of Derek's progress
- Derek refuses to follow the red line after the second bend.
- Derek would follow the black lines without issue but not the red line.
- We changed the value of the offset and changed the integers to doubles for more accurate tweaking.



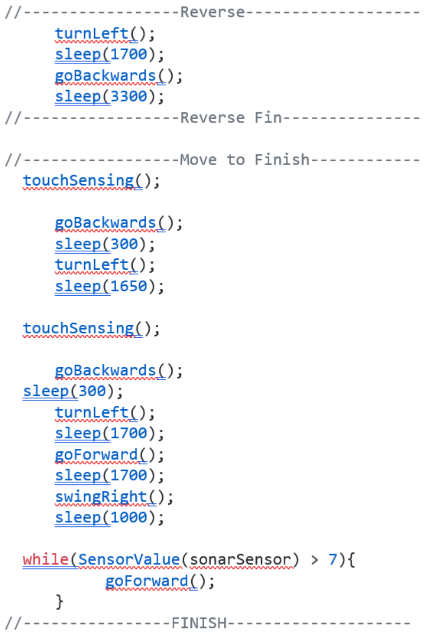
```
// Driver code
task main()
{
//--------------------Up---------------------
	touchSensing(); // drives forward until wall is hit.

	goBackwards();
	sleep(2000);
	turnLeft();
	sleep(1750);
	goForward();
	sleep(2000);
//--------------------Up Finished------------

//--------------------Follow Line------------

  /* The robot follows the line until the wall at the end of the line is
hit. It "swings" right while the color sensor detects *only* red ( its
specified that it doesn't swing right when it sees black due to issues
with the board being black and white ). When the sensor sees white (1) it
swings left to try find the red line again.*/

	while(SensorValue(touchSensor) == NO_TOUCH){
		while(SensorValue(colorSensor) == RED) {
			swingRight();
		}
		while(SensorValue(colorSensor) != RED) {
			swingLeft();
		}
	}
//----------------Follow Line Fin-----------
```

- Conor removed a conditional statement and Derek finally started following the red line after the bend.
- Peter implemented the code for the last part of the obstacle

- We downloaded the code onto Derek and tried it, Derek would move successfully until it hit into a wall and then stopped.

5/02/20

- John, Peter, Daniel, Finbar and Conor met up to sort out the remaining bugs in Derek's code.
- We all brainstormed ideas to get around the issues at hand
- Derek was doing the first 80% of the obstacle course nicely at this stage but was still failing the last part.
- Conor added some code in to more efficiently execute some parts while the rest of us tried out different ideas.
- Conor and Peter came up with an idea to improve the code and it was implemented and downloading onto the robot.
- Derek made one successful trip around the obstacle and we recorded it.



```
//-----------------Reverse------------------
      turnLeft();
      sleep(1700);
      goBackwards();
      sleep(3300);
//-----------------Reverse Fin---------------

//-----------------Move to Finish------------
  touchSensing();

      goBackwards();
      sleep(300);
      turnLeft();
      sleep(1650);

  touchSensing();

      goBackwards();
  sleep(300);
      turnLeft();
      sleep(1700);
      goForward();
      sleep(1700);
      swingRight();
      sleep(1000);

  while(SensorValue(sonarSensor) > 7){
          goForward();
      }
//----------------FINISH-------------------
```

7/02/20

- John, Peter, Daniel, Finbar and Conor met up to sort out the final tweaks to be made to Derek.
- We brushed over the code to make look as clean as possible all while Daniel made the finishing touches to the learning journal.
- Daniel then added the final code to be submitted alongside the journal.

Team member Jobs & Roles:

- **John:** helped building the robot, code tweaking and taking down Derek's inconsistencies and errors on the obstacle course
- **Peter**: wrote the code for the last part of the obstacle course and helped tweak and adjust any errors present
- **Daniel:** Helped finish building the robot. Wrote the learning journal and took notes every meetup, also helped provide insight into code errors and inconsistencies on the obstacle course.
- **Finbar:** Helped build the robot initially and helped when tweaking code or brainstorming better methods.
- **Conor:** Wrote the majority of the code. Tweaked and adjusted the code to make Derek run better.
- **Matthew:** Filmed Derek's progress and helped build Derek.