

Daniel MacMillan

Dangnhi Ngo

Mobile Robotics I - COMP 4500

Spring 2018

Final Project Report

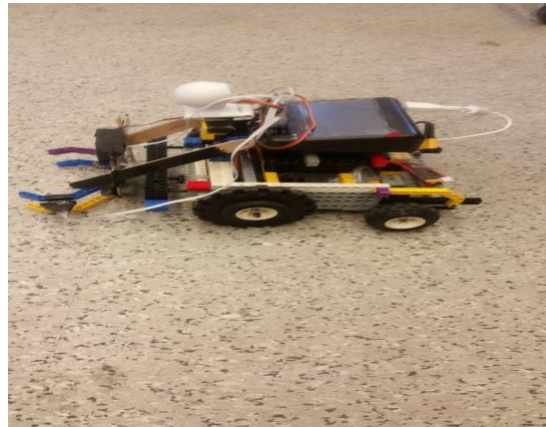
1. Description

BALL-E

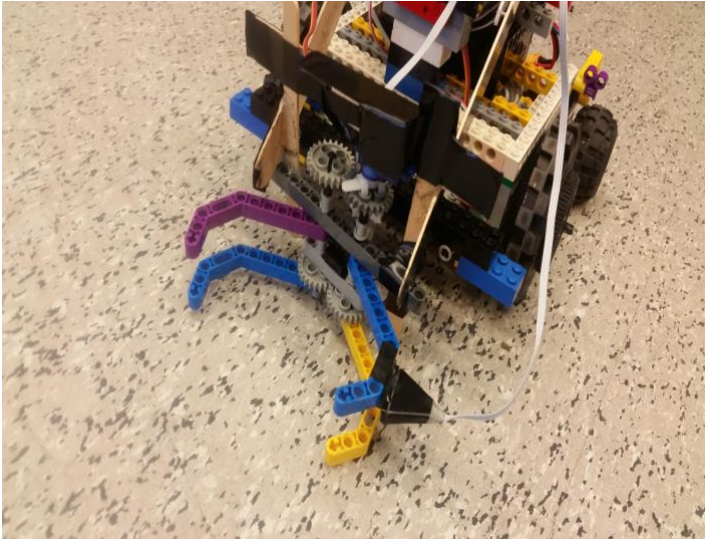
In this project, our robot uses the camera to approach green balls, grabs them and lifts them up with a claw, and then transitions to finding a yellow bin to drop it off in. Once it sees a yellow bin, the robot approaches it until a bumper is hit, which causes the robot to release the ball, turn around, put down its claw, and start looking for green balls again.

2. Design

Our robot uses a Wallaby Controller mounted on board the robot in order to interface with the sensors and to run the program. Its battery is also mounted on board underneath and is charged externally.



4 wheels keep the robot off of the ground. The two larger wheels near the front are powered by two motors mounted directly on each wheel which provides the ability to steer with two wheel drive.



One step servo mounted on top of the claw turns gears which keep the two arms of the claw closed or opened. Another two step servos lifts the claw and ball up over the camera and puts them down as well.

Two top hat sensors located on the claw arm and just behind the claw detect whether or not there is a ball inside of the claw. The top

hat sensors were well suited for the task of detecting objects because other distance sensors had too much variance and unpredictability when it came to gauging short distances. Two were used instead of just one in order to increase the odds of successfully detecting a ball in its claw.

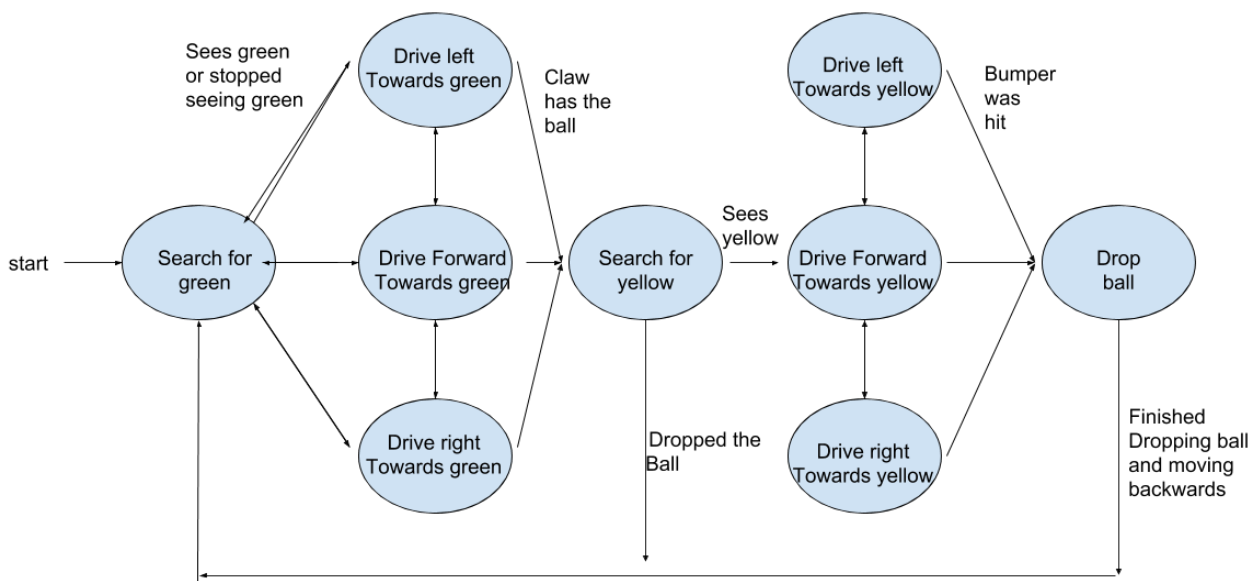
Other sensors on the robot include the camera on the top which looked for green and yellow objects, as well as the two bumper sensors on the front of the robot behind the claw and bumper. The bumper, once hit, would cause the claw to release the ball, move the robot backwards, and set the claw down for it to start searching for balls again.

Materials used in the final robot design include 2 top hat sensors, 3 step servos, 2 bumpers, 1 camera, 4 wheels and 2 motors mentioned above, legos, popsicle sticks (large and small), zip ties, tape, and hot glue.

3. Code Design

In our code, we have subroutines for turning left/right, driving forward/backward, lifting the claw up/down, and opening/closing the claw.

Behavior was modeled after a finite state machine with only one process running.



4. Source Code

```
/* FINAL PROJECT */  
  
/* Using the camera to approach green balls and then transition to finding a  
* yellow bin. Use claw controlled by single servo to grip and release ball  
* once it enters the opened claw. Use two other servos to lift the claw and  
* ball overhead, and once bumper is hit (presumably by the yellow bin),  
* release the ball.  
*/  
  
/* Daniel MacMillan and Dangnhi Ngo*/  
  
#include <kipr/botball.h>  
  
#define MAX_VEL 30  
#define CLAW_CLOSE 400  
#define CLAW_OPEN 0  
  
//function to make the robot turn left  
void turn_left()  
{  
    motor(0, MAX_VEL*2);  
    motor(1, -MAX_VEL*2);  
    //printf("turn left\n");  
    msleep(20);  
}  
  
//function to make the robot turn right  
void turn_right()  
{  
    motor(0, -MAX_VEL*2);
```

```

        motor(1, MAX_VEL*3);
        //printf("turn right\n");
        msleep(20);
    }

//function to make the robot go straight
void go_straight()
{
    motor(0,MAX_VEL/2*3);
    motor(1,MAX_VEL/2*3);
    //printf("go straight\n");
}

//function to make the robot go backward
void go_back()
{
    motor(0,-MAX_VEL/2*3);
    motor(1,-MAX_VEL/2*3);
    //printf("go back\n");
}

/* open_claw: sets the claw on the ground and opens the claw to its max width
*/
void open_claw()
{
    set_servo_position(2, CLAW_OPEN);
    printf("claw opened\n");
}

/* close_claw(): closes the claw around ball and lifts the claw up
*/

```

```

void close_claw()
{
    set_servo_position(2, CLAW_CLOSE);
    printf("claw closed\n");
}

/* arm_up(): set servos to move the claw up over the camera
*/
void arm_up()
{
    set_servo_position(0, 1024);
    set_servo_position(1, 1024);
}

/* arm_down(): set servos to set the claw down on the ground
*/
void arm_down()
{
    set_servo_position(0, 0);
    set_servo_position(1, 2047);
}

int main()
{
    // Start state of the robot
    camera_open();
    enable_servos();
    arm_down();
    open_claw();
    int can_feel_ball = 0;
    while(!c_button_clicked())

```

```

{
camera_update();
int obj_x = get_object_center_x(0,0);
int bin_x = get_object_center_x(1,0);
int analog0 = analog(0);
int analog1 = analog(1);
int bumped = (digital(0) || digital(1));

// check if the ball is in the claw, if so - grip
if(can_feel_ball == 1)
{
    if((analog0 > 100 && analog1 > 90))
    {
        can_feel_ball = 0;
        arm_down();
        msleep(20);
        open_claw();
    }
//hit the trash bin, drop the ball off
if(bumped == 1)
{
    open_claw();
    go_back();
    msleep(1000);
    turn_left();
    msleep(500);
    arm_down();
    can_feel_ball = 0;
}

// look for the bin

```

```

        if (bin_x > 70 && bin_x < 90)
        {
            printf("x: %d\n", bin_x);
            go_straight(); // we see the bin
        }
//object is on the right, turn right
else if (bin_x > 100)
{
    printf("x: %d\n", bin_x);
    turn_right();
}
//object is on the left, turn left
else if (bin_x < 50)
{
    printf("x: %d\n", bin_x);
    turn_left();
}
}
if(can_feel_ball == 0)
{
    if(analog0 < 90 || analog1 < 70)
    {
        can_feel_ball = 1;
        close_claw();
        msleep(20);
        arm_up();
    }

    // look for the ball
    if (obj_x > 70 && obj_x < 80)
    {

```



```

        go_straight();
    }
    //object is on the right, turn right
    else if (obj_x > 100)
    {
        turn_right();
    }
    //object is on the left, turn left
    else if (obj_x < 50)
    {
        turn_left();
    }
    }
    }
    open_claw();
    disable_servos();
    camera_close();
    return 0;
}

```

5. “Did your robot behaved as planned during the demo?”

The robot behaved as planned when demoing to the cameramen. When the professor stopped by the on board camera glitched and could not pick up any data around it. This behavior has been previously encountered - a quick restart was necessary to get it working again. By then, however, a ball had already been placed in the bin and was seen as a sufficient demo for behavior.

6. “If you were to redesign your project, what changes would you make?”

I would have the bin be attached to the back of the robot, making it so we don’t need to worry about the occasional ball falling away from the bin. Also, reaching the stretch goals of closing the lab door would have been fun.

7. “What changes would you make to this course?”

Daniel:

I don't think any changes are necessary - this course was a blast!

Dangnhi:

I have had a great experience with this course. I have learnt how to build a robot and do a lot of fun things with it. I would highly recommend CS students to take this class for elective course.

Nothing needs to be changed.