Tunisian School (0717)

P2 Code Review Document

Introduction

As part of our work, coding should be reviewed in order to avoid mistakes during the competition, especially for our main robot DEKU who will do most of the work. Ahmad Samara is the programmer of DEKU's code. Azer Abdallah and Fadi Belgacem are responsible of the code review which will be conducted on 20/01/2019.

Best Practices Checklist

	Code uses functions for better organization
	Code contains comments documenting each function's purpose
	Code contains comments documenting each function's argument
	Code contains comments documenting each function's returns value
	All variable names are descriptive and convey use in code
	No unnamed constants other than 0,1 or 2
	Code is formatted to show flow control
	Comments do not contain old block of code

In our code we have constants that can't be named by numbers (0, 1, 2..). So it will be more easy to use them if they're named by familiar names (name that makes us remember to what it refers) instead of numbers. This will make our work easier and faster.

General code analysis

Reliability

While the wheels' movements are accurate to some extent, we wanted to have a redundant system to ensure that the robot moves straightforward without any deviation. To accomplish this out robot has two perpendicular *Tophat* sensor (facing the border of the black tape line). In our code we are checking the sensors' detection value, if both sensors indicate white color's value then the robot is moving correctly. If one of the sensors indicates black color's value the program runs a correction routine that does a weighted average to determine how to correct the position.
This code is currently pretty reliable, but the accuracy can be increased. The more we test the robot, the more we can fine tune the correction routine. Currently our accuracy is plus or minus 1 inches of the deviation, but we hope to lower that to 0,5 inch through future testing.

Maintainability

While Ahmad Samara has written most of the code, our team leader requires all members of the programming team to read through the code twice a week. We do this so they can have at least a rough understanding of the code. Because we do this weekly, our code has been easy to understand and modify for everyone. We used a lot of comments at the beginning of each function stating its task, and additional comments in the functions to clarify any complicated line and understand it better. We think that we are already have really good code maintainability. From here our next step would be to make a poster, in which we will write down all of the information that could be beneficial for us.

Effectiveness

Currently our code does effectively perform the task assigned. We go out and collect the food supplies and take them back to our starting box. This is all done in a matter of seconds. We are so efficient at the moment, that we are looking to add some more functionality to our robot.

```
int main()
{
   clear_motor_position_counter(0);
   cmpc(3);

   while(get_motor_position_counter(3) < 14000)
   {

     motor(3, 75);

     if(gmpc(0) < gmpc(3))
     {
       motor(0, 100);
     }
     else
     {
       motor(0, 50);
     }
   }

   ao();

   return 0;
}
```

We have to be careful when adding on to our robot, that we do not compromise the high efficiency that we already have. If we cannot find a way to score more points quickly and easily then we will go back to our current code. It is better to score a few points well, then to maybe score a lot.

```
#include <kipr/botball.h>
int main()
{
  printf ("Drive to the wall\n");

  while (digital(0) == 0)  // Touch sensor not touched
  {
     if (analog(0) <= 2700) // Far away drive forward
     {
        motor(0,80);
        motor(3,80);
     }
     if (analog(0) > 2701) // Too close back up
     {
        motor(0,-80);
        motor(3,-80);
     }
  }
  ao();
  return 0;
```