# RobotC Virtual Worlds Challenges

**Group members:**

De’Paul University:

Seth Weber

William

UNESP:

Karen Gomes

Luis Guilherme Fernandes

Pius Oehling

Willian Mesquita

**Question 1.** Sensabot Challenge

#pragma config(StandardModel, "EV3\_REMBOT")

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

void moveStraight(int sleep\_time)

{

setMotorSpeed(motorB, 50);

setMotorSpeed(motorC, 50);

sleep(sleep\_time);

setMotorSpeed(motorB, 0);

setMotorSpeed(motorC, 0);

}

void moveBackward(int sleep\_time)

{

setMotorSpeed(motorB, -55);

setMotorSpeed(motorC, -55);

sleep(sleep\_time);

setMotorSpeed(motorB, 0);

setMotorSpeed(motorC, 0);

}

void inspect()

{

setMotorSpeed(motorA, -50);

sleep(300);

setMotorSpeed(motorA, 50);

sleep(300);

setMotorSpeed(motorA, 0);

}

task main()

{

moveStraight(1350);

inspect();

delay(300);

moveStraight(1350);

inspect();

delay(300);

moveStraight(1500);

inspect();

delay(300);

moveBackward(4500);

}

**Question 2.** Orchard Challenge 1.0

#pragma config(StandardModel, "EV3\_REMBOT")

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

void moving(int x)

{

moveMotorTarget(motorB, x\*214.1961471468032, 50);

moveMotorTarget(motorC, x\*214.1961471468032, 49.99999799999);

waitUntilMotorStop(motorB);

}

void turn(int x)

{

moveMotorTarget(motorB, x\*1.8888888664, 50);

moveMotorTarget(motorC, x\*-1.8888888664, 50);

waitUntilMotorStop(motorB);

}

task main()

{

moving(8);

turn(-90);

moving(2);

turn(-90);

moving(7);

turn(135);

moving(8.2);

turn(-90);

moving(2.5);

turn(-90);

moving(9);

turn(90);

moving(2.8);

turn(80);

moving(9);

}

**Comment:**

The first challenge was completed by the 15th of April classroom. And the second one, during the following week. On it, we discovered that the robot’s wheel diameter is not 5.6cm (as it is said), but 5.35cm. That is, if we consider pi=3.1415. And we found the right motor (motorC) to be slightly faster than the left motor (motorB), even though the software simulates a virtual world and should be subsequently “perfect”. The values used in the void turn function were found by trial and error. But in the void moving function, they were calculated and tested. With the same lines of code, we faced several different results during testing. In none of those, was the second challenge completed, but it got very close to be.