8059D

VR Skills Challenge Over Under

by: Ian, Kai Ray, Marcus & Gabriel

```
#region VEXcode Generated Robot Configuration
import math
import random
from vexcode_vrc import *
from vexcode_vrc.events import get_Task_func

# Brain should be defined by default
brain=Brain()

drivetrain = Drivetrain("drivetrain", 0)
arm_motor = Motor("ArmMotor", 3)
rotation = Rotation("Rotation", 7)
intake_motor = Motor("IntakeMotor", 8)
optical = Optical("Optical", 11)
gps = GPS("GPS", 20)

import math

X = 0
Y = 0
Bearing = 0
armdown = 1440;
toDeg = 180/3.14159

def baseMove(targX , targY, rev = False):
    brain.screen.clear_screen()
    flag = True

    while flag:
        drivetrain.set_drive_velocity(100, PERCENT)
        drivetrain.set_turn_velocity(100, PERCENT)

        # getting robot details
        Bearing = drivetrain.rotation(DEGREES)
        X = gps.x_position(INCHES);
        Y = gps.y_position(INCHES);
```

Getting the position of the robot and storing it in variables

```
        # printing robot details
        brain.screen.print("X Location:" + str(X), precision=3)
        brain.screen.next_row()
        brain.screen.print("Y Location:" + str(Y), precision=3)
        brain.screen.next_row()
        brain.screen.print("Bearing:" + str(Bearing), precision=3)
        brain.screen.next_row()

        # finding error in X and Y
        errorX = targX - X;
        errorY = targY - Y;
```

Finding the error between the target and current position

```
        # printing error details
        brain.screen.print("errorX:" + str(errorX), precision=3)
        brain.screen.next_row()
        brain.screen.print("errorY:" + str(errorY), precision=3)
        brain.screen.next_row()

        #Finding Turn Angle
        arcTan = math.atan2(errorX, errorY)
        BearingChange = arcTan * toDeg
        targBearing = BearingChange + 180 if rev else BearingChange
        brain.screen.print("TargBearing:")
        brain.screen.print(targBearing,precision=3)
        brain.screen.next_row()
        drivetrain.turn_to_rotation(targBearing, DEGREES)
```

Using trigonometry to calculate the turn bearing

```
        #Finding Distance Using Pythagoras
        distance = -pow((pow (errorX, 2) + pow(errorY, 2)), 0.5) if rev else pow((pow (errorX, 2) +
pow(errorY, 2)), 0.5)
        brain.screen.print(distance)
        brain.screen.next_row
        drivetrain.drive_for(FORWARD, distance, INCHES)
        wait(5,MSEC)
        flag = False

def baseTurn(bearing):
    drivetrain.turn_to_rotation(bearing, DEGREES)

def armMove():
    arm_motor.set_velocity(600, RPM)
    arm_motor.spin(FORWARD)

def intakeMove(dir):
    intake_motor.set_velocity(600, RPM)
    intake_motor.spin(dir)

def delay(dT):
    wait(dT, MSEC)

def main():
    #starting position is -35.2, -59.05

    #First Ball
    armMove()
    baseMove(-35.2,0)
    baseTurn(-90)
    intakeMove(REVERSE)
    delay(650)

    # Second Ball
    arm_motor.stop()
    intakeMove(FORWARD)
    baseTurn(0)
    delay(100)
    baseTurn(-90)
    intakeMove(REVERSE)
    delay(650)

    #Third Ball
    intakeMove(FORWARD)
    baseTurn(90)
    baseMove(-15,0)
    baseMove(-15,-10)
    baseTurn(90)
    drivetrain.drive(FORWARD)
    intakeMove(REVERSE)
    delay(700)
    drivetrain.stop()

    # Fourth Ball
    drivetrain.drive(REVERSE)
    delay(690);
    intakeMove(FORWARD)
    drivetrain.stop()
    baseTurn(30)
    delay(100)
    baseTurn(90)
    drivetrain.drive(FORWARD)
    intakeMove(REVERSE)
    delay(800)
    drivetrain.stop()

    # Fifth Ball
```

Using Pythagoras to calculate the distance moved

Function To turn

Function To move intake and arm

```
drivetrain.drive(REVERSE)
delay(600);
intakeMove(FORWARD)
drivetrain.stop()
baseTurn(170)
delay(100)
baseTurn(90)
drivetrain.drive(FORWARD)
intakeMove(REVERSE)
delay(800)
drivetrain.stop()

# Sixth Ball
drivetrain.drive(REVERSE)
delay(900);
intakeMove(FORWARD)
drivetrain.stop()
baseTurn(180)
baseMove(-6,-30)
baseMove(-8,12, True)
baseTurn(90)
drivetrain.drive(FORWARD)
intakeMove(REVERSE)
delay(700);
drivetrain.stop()


# Seventh Ball
drivetrain.drive(REVERSE)
delay(700);
intakeMove(FORWARD)
drivetrain.stop()
baseTurn(10)
baseMove(-7,15)
baseTurn(90)
drivetrain.drive(FORWARD)
intakeMove(REVERSE)
delay(700)
drivetrain.stop()

# Eighth Ball
drivetrain.drive(REVERSE)
delay(800)
intakeMove(FORWARD)
drivetrain.stop()
baseMove(-6,30)
baseMove(-5,15)
baseTurn(90);
drivetrain.drive(FORWARD)
intakeMove(REVERSE)
delay(800)
drivetrain.stop()

# Ninth Ball
baseMove(55,55);
intakeMove(FORWARD)
delay(200)
baseTurn(180)
intakeMove(REVERSE)
baseMove(55,40)
delay(600)
baseMove(30,57)
intakeMove(FORWARD)
baseMove(0,57)

# Tenth Ball
```

```
    baseMove(50, 38)
    intakeMove(REVERSE)
    baseTurn(180)
    drivetrain.drive(REVERSE)
    delay(690);
    intakeMove(FORWARD)
    drivetrain.stop()
    baseTurn(30)
    delay(100)
    baseTurn(90)
    drivetrain.drive(FORWARD)
    intakeMove(REVERSE)
    delay(800)
    drivetrain.stop()

    # 11th - 15th Ball
    drivetrain.drive(REVERSE)
    delay(690);
    intakeMove(FORWARD)
    drivetrain.stop()
    baseTurn(30)
    delay(100)
    baseTurn(90)
    drivetrain.drive(FORWARD)
    intakeMove(REVERSE)
    delay(800)
    drivetrain.stop()




#VR threads TEST – Do not delete
vr_thread(main)
```

8059blank. Robotics
Republic of Singapore