6627S Virtual Skills

Santa Ana, CA
Presented by Cian

Autonomous Routine Code

```
#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)

#endregion VEXcode Generated Robot Configuration
timer = 0
randomAngle = 0
#First Function that works on a timer baseline and allows for the initial skills part
to be completed
#Less Adjustable more speed on outtake exit
def intakeJuggle():
   global timer
   #While loop with easy adjustment through variable
   while timer >= 1:
       intake_motor.spin(REVERSE)
       wait(500,MSEC)
       intake_motor.spin(FORWARD)
       timer= timer-1
#More Adjustable less speed on Outtake exit
def throw(ending, timeT):
   while timeT < ending:
       intake_motor.spin(FORWARD)
       wait(5,MSEC)
       intake_motor.spin(REVERSE)
       timeT = timeT-5
   intake_motor.spin(REVERSE)
#Allow all motors at maximum speed
def initialize():
   arm_motor.set_velocity(100, PERCENT)
   drivetrain.set_drive_velocity(100, PERCENT)
   drivetrain.set_turn_velocity(100, PERCENT)
   intake_motor.set_velocity(100,PERCENT)
#Prior Test With no Result
   #Second Function Consists of the Match Loading Part of the Skills Run
   #def bowl(usesO, clock):
       #Stores starting input of uses for later use
       #uses = usesO
       #Two while loops with limited runs
       #while clock >=1:
           #Match loading and Reset
           #while uses >=1:
           #    intake_motor.spin(FORWARD)
               #wait(300,MSEC)
           # randomAngle = random.randint(90,100)
           #  drivetrain.turn_to_heading(randomAngle,DEGREES)
           #   intake_motor.spin(REVERSE)
           #    wait(760,MSEC)
       #     intake_motor.spin(FORWARD)
       #      drivetrain.turn_to_heading(225,DEGREES)
       #       drivetrain.drive_for(FORWARD,2.3,INCHES)
       #        uses = uses-1
           #Pushing Triballs down the Alley then resetting to Match Loading
           #drivetrain.turn_to_heading(95,DEGREES)
       # intake_motor.spin(REVERSE)
       #  drivetrain.drive_for(FORWARD,20,INCHES)
       #   drivetrain.drive_for(REVERSE,21,INCHES)
       #    intake_motor.spin(FORWARD)
           #drivetrain.turn_to_heading(225,DEGREES)
       # drivetrain.drive_for(FORWARD,1,INCHES)
       #  #Reset Match Loading Runs
       #   uses=uses+usesO
       #    clock=clock-1
       #Finish Match Loading then Push triballs down Alley and Score Them
   # intake_motor.spin(REVERSE)
       #drivetrain.set_drive_velocity(60,PERCENT)
   #  drivetrain.turn_to_heading(135,DEGREES)
   #   drivetrain.drive_for(FORWARD,5,INCHES)
   #    drivetrain.turn_to_heading(90,DEGREES)
   # drivetrain.drive_for(FORWARD,55,INCHES)
   #  drivetrain.drive_for(REVERSE,5,INCHES)
   #   wait(1,SECONDS)
   #    drivetrain.drive_for(FORWARD,35,INCHES)
       #drivetrain.drive_for(REVERSE,2,INCHES)
   # drivetrain.turn_to_heading(45,DEGREES)
   #  drivetrain.drive_for(FORWARD,45,INCHES)
#Start by Initializing then Scoring Both Preloads and rotating the arm down
vr_thread(initialize)
arm_motor.spin(FORWARD)
```

```
drivetrain.drive_for(FORWARD,58,INCHES)
drivetrain.turn_to_heading(270,DEGREES)
intake_motor.spin(REVERSE)
wait(650,MSEC)
intake_motor.spin(FORWARD)
drivetrain.turn_to_heading(0,DEGREES)
wait(375,MSEC)

#Intake then Outake One over Barrier then Score One and Score Other Later
drivetrain.turn_to_heading(270,DEGREES)
intake_motor.spin(REVERSE)
wait(650,MSEC)
intake_motor.spin(FORWARD)
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,5,INCHES)
wait(20,MSEC)
timer=5
vr_thread(intakeJuggle)
drivetrain.drive_for(FORWARD,49,INCHES)
drivetrain.drive_for(REVERSE,21.5,INCHES)
intake_motor.spin(FORWARD)
drivetrain.turn_to_heading(0,DEGREES)

drivetrain.drive_for(FORWARD,15,INCHES)
drivetrain.turn_to_heading(90,DEGREES)

#Launch 3 Triballs into Goal
timer = 3.5
vr_thread(intakeJuggle)
drivetrain.drive_for(FORWARD,19,INCHES)
wait(100,MSEC)
drivetrain.drive_for(REVERSE,20,INCHES)
intake_motor.spin(FORWARD)
drivetrain.turn_to_heading(180,DEGREES)
drivetrain.drive_for(FORWARD,30,INCHES)
wait(100,MSEC)
drivetrain.turn_to_heading(90,DEGREES)
vr_thread(throw(0,25))
drivetrain.drive_for(FORWARD,20,INCHES)
wait(100,MSEC)
drivetrain.drive_for(REVERSE,21,INCHES)
drivetrain.turn_to_heading(175,DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,25,INCHES)
drivetrain.drive_for(REVERSE,30,INCHES)
drivetrain.turn_to_heading(90,DEGREES)
vr_thread(throw(0,25))
```

```
#Score triball outaked over Previously
drivetrain.drive_for(FORWARD,22,INCHES)
drivetrain.turn_to_heading(0,DEGREES)
drivetrain.drive_for(FORWARD,14,INCHES)
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,10,INCHES)
drivetrain.turn_to_heading(20,DEGREES)

#Take Triball from corner and score
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,41,INCHES)
drivetrain.turn_to_heading(45,DEGREES)
drivetrain.drive_for(FORWARD,18,INCHES)
drivetrain.turn_to_heading(150,DEGREES)
intake_motor.spin(REVERSE)
drivetrain.drive_for(FORWARD,15,INCHES)
drivetrain.turn_to_heading(270,DEGREES)

#Grab Last Triball on field and Score
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,50,INCHES)
drivetrain.drive_for(REVERSE,50,INCHES)
drivetrain.turn_to_heading(180,DEGREES)
intake_motor.spin(REVERSE)
drivetrain.drive_for(FORWARD,3,INCHES)
wait(650,MSEC)

#Transition into Match Loading
drivetrain.turn_to_heading(270,DEGREES)
drivetrain.drive_for(FORWARD,90,INCHES)
intake_motor.spin(FORWARD)
drivetrain.turn_to_heading(315,DEGREES)
drivetrain.drive_for(FORWARD,33,INCHES)
drivetrain.drive_for(REVERSE,60,INCHES)
vr_thread(throw(0,155))
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,19,INCHES)
wait(100,MSEC)
drivetrain.drive_for(REVERSE,19,INCHES)
drivetrain.turn_to_heading(315,DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,60,INCHES)
drivetrain.drive_for(REVERSE,60,INCHES)
vr_thread(throw(0,155))
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,19,INCHES)
```

```
wait(100,MSEC)
drivetrain.drive_for(REVERSE,19,INCHES)
drivetrain.turn_to_heading(313,DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,60,INCHES)
drivetrain.drive_for(REVERSE,60,INCHES)
vr_thread(throw(0,155))
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,19,INCHES)
wait(100,MSEC)
drivetrain.drive_for(REVERSE,19,INCHES)
drivetrain.turn_to_heading(313,DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,60,INCHES)
drivetrain.drive_for(REVERSE,60,INCHES)
vr_thread(throw(0,155))
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,19,INCHES)
wait(100,MSEC)
drivetrain.drive_for(REVERSE,19,INCHES)
drivetrain.turn_to_heading(314,DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive_for(FORWARD,60,INCHES)
drivetrain.drive_for(REVERSE,60,INCHES)
vr_thread(throw(0,155))

#Push in and Score Match Loads
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,45,INCHES)
drivetrain.drive_for(REVERSE,5,INCHES)
drivetrain.turn_to_heading(0,DEGREES)
drivetrain.drive_for(FORWARD,5,INCHES)
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD,10,INCHES)
```