

Program introduction for VIQC

Virtual Skills Challenge

- Full Volume

Score: 96 pts, 1 second remaining

Author: DIANXIAO ZHAO

Teacher: XIN WANG

Team Number: 15159C

Team Name: Brave the waves

Participants: DIANXIAO ZHAO, PENGXIANG WANG, KANGZE SHI

Youth Science & Technology Center of Beijing Xicheng China

January 19th, 2024

1、 Our team

We are 15159C team from Youth Science & Technology Center of Beijing Xicheng China. There are 3 boys in our team. My Teacher is XIN WANG.

I have been learning python for 3 years, so our program is written in python.

In this challenge I have learned how to write a thread and how to debug a program more efficiently.

2、 Strategy

a) It's the easiest way to score red blocks.

b) Fill level is important than scored block number.

c) We will get a higher score if the robot can full park.

3、 Goal

a) Put 2 red blocks into goal II to reach Fill Level 3.

b) Put 4 purple blocks into goal III to reach Fill Level 3.

c) Put 5 purple blocks into goal I to reach Fill Level 3.

d) Full park

4、 How to do?

a)The robot adjusts the direction after running for a while, so that it can run straighter.

Example:

```
for i in range(3):  
    drivetrain.drive_for(FORWARD, 100 * 4, MM, wait=True)  
    drivetrain.turn_to_heading(-45.5, DEGREES)
```

It is a loop program, the robot will run for a distance and adjust the direction in each loop.

b)The robot starts releasing blocks controlling by threads before it reaches the goal. It is faster than releasing the block after it reached the goal.

Example:

```
def outIntake2():  
    # thread function  
    # 1.robot sleeps for a while before reaching the goal.
```

```

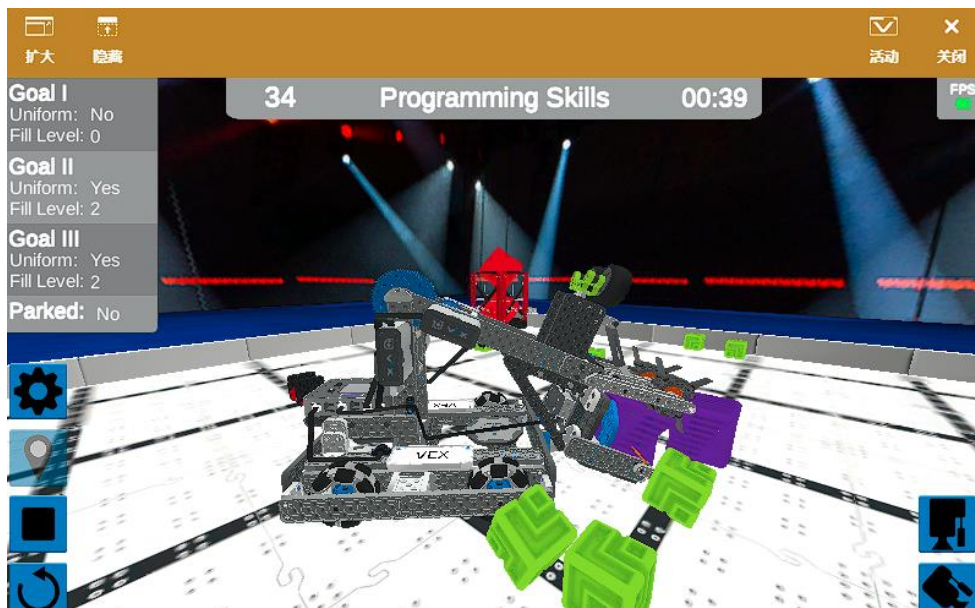
# 2.release object
intakeSleepTime = 1 # sleep time after the thread created
intakeRunningTime = 300 # angle the intake_motor_group will turn
wait(intakeSleepTime,SECONDS)
intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

drivetrain.turn_to_heading(0,DEGREES)
vr_thread(outIntake2)
drivetrain.drive_for(FORWARD, 600, MM)

```

We create a thread after the robot begin to run to the goal. In the thread robot sleeps for a while before reaching the goal and starts releasing the block when it is near the goal.

c) The robot can push one block to the goal while holding another one.



d) Design the shortest running route.

5、 About my code

The program runs step by step in main function.

There are 11 functions named like outIntake1. They are thread worker functions to release the blocks before the robot reaches the goal.

Our Code is shown below.

```
#region VEXcode Generated Robot Configuration
import math
import random
from vexcode_viqc import *

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

drivetrain = Drivetrain("drivetrain", 0)
intake_bumper = Bumper("IntakeBumper", 3)
front_optical = Optical("FrontOptical", 4)
intake_motor_group = Motor("IntakeMotorGroup", 5)
arm_motor_group = Motor("ArmMotorGroup", 6)
front_distance = Distance("FrontDistance", 9)
```

```

#endregion VEXcode Generated Robot Configuration
# -----
#
# Project:      VIQC Virtual Skills - Full Volume
# Author:      Dianxiao Zhao
# Created:     Dianxiao Zhao
# Description:  This is the virtual skills program for Full Volume game
# -----

def outIntake1():
    # thread function
    # 1.sleep
    # 2.release block
    intakeSleepTime = 0.2 # sleep time after the thread created
    intakeRunningTime = 300 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake2():
    # thread function
    # 1.sleep
    # 2.release block
    intakeSleepTime = 1 # sleep time after the thread created
    intakeRunningTime = 300 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake3():
    # thread function
    # 1.sleep
    # 2.release block
    intakeSleepTime = 0.1 # sleep time after the thread created
    intakeRunningTime = 200 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake4():
    # thread function
    # 1.sleep
    # 2.release block

```

```

intakeSleepTime = 1.9 # sleep time after the thread created
intakeRunningTime = 500 # angle the intake_motor_group will turn
wait(intakeSleepTime,SECONDS)
intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake5():
    # thread function
    # 1.sleep
    # 2.release block
    wait(2.8,SECONDS)
    arm_motor_group.spin_for(FORWARD,280,DEGREES,wait=False)
    intakeSleepTime = 0.1 # sleep time after the thread created
    intakeRunningTime = 500 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake6():
    # thread function
    # 1.sleep
    # 2.release block
    intakeSleepTime = 0.1 # sleep time after the thread created
    intakeRunningTime = 900 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake7():
    # thread function
    # 1.sleep
    # 2.release block
    wait(3,SECONDS)
    arm_motor_group.spin_for(FORWARD,280,DEGREES,wait=False)
    intakeSleepTime = 0.5 # sleep time after the thread created
    intakeRunningTime = 500 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake8():
    # thread function
    # 1.sleep
    # 2.release block
    arm_motor_group.spin_for(FORWARD,280,DEGREES,wait=True)
    intakeSleepTime = 0 # sleep time after the thread created
    intakeRunningTime = 800 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)

```



```

intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake9():
    # thread function
    # 1.sleep
    # 2.release block
    intakeSleepTime = 0 # sleep time after the thread created
    intakeRunningTime = 500 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake10():
    # thread function
    # 1.sleep
    # 2.release block
    intakeSleepTime = 0.1 # sleep time after the thread created
    intakeRunningTime = 1000 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

def outIntake11():
    # thread function
    # 1.robot sleeps for a while before reaching the goal.
    # 2.release block
    intakeSleepTime = 0.1 # sleep time after the thread created
    intakeRunningTime = 1000 # angle the intake_motor_group will turn
    wait(intakeSleepTime,SECONDS)
    intake_motor_group.spin_for(REVERSE,intakeRunningTime,DEGREES,wait=True)

# Add project code in "main"
def main():
    # default setting
    drivetrain.set_turn_velocity(100,PERCENT)
    drivetrain.set_drive_velocity(100,PERCENT)
    intake_motor_group.set_velocity(100,PERCENT)
    arm_motor_group.set_velocity(100,PERCENT)

    # score the 1st red block
    intake_motor_group.spin(FORWARD)
    drivetrain.turn_to_heading(43,DEGREES)
    drivetrain.drive_for(FORWARD, 110, MM)

```

```

arm_motor_group.spin_for(FORWARD,320,DEGREES,wait=False)
drivetrain.turn_to_heading(-105,DEGREES)
# create a thread using outIntake1 thread function to release block while
running
vr_thread(outIntake1)
drivetrain.drive_for(FORWARD, 390, MM)

# score the 1st purple block
drivetrain.turn_to_heading(-30,DEGREES)
intake_motor_group.spin(FORWARD)
arm_motor_group.spin_for(REVERSE,260,DEGREES,wait=False)
drivetrain.drive_for(FORWARD, 600, MM)
arm_motor_group.spin_for(FORWARD,240,DEGREES,wait=False)
drivetrain.turn_to_heading(0,DEGREES)
vr_thread(outIntake2)
drivetrain.drive_for(FORWARD, 600, MM)

# score the 2nd purple block
drivetrain.turn_to_heading(60,DEGREES)
intake_motor_group.spin(FORWARD)
arm_motor_group.spin_for(REVERSE,240,DEGREES,wait=False)
drivetrain.drive_for(FORWARD, 400, MM)
arm_motor_group.spin_for(FORWARD,240,DEGREES,wait=False)
drivetrain.drive_for(REVERSE, 380, MM)
vr_thread(outIntake3)
drivetrain.turn_to_heading(-15,DEGREES)
wait(0.3,SECONDS)

# score the 2nd red block
arm_motor_group.spin_to_position(0,DEGREES,wait=False)
drivetrain.turn_to_heading(90,DEGREES)
intake_motor_group.spin(FORWARD)
drivetrain.drive_for(FORWARD, 280, MM)
arm_motor_group.spin_for(FORWARD,280,DEGREES,wait=False)
drivetrain.turn_to_heading(200,DEGREES)
vr_thread(outIntake4)
# adjust the direction once afte specified distance
for i in range(2):
    drivetrain.drive_for(FORWARD, 100*4, MM,wait=True)
    drivetrain.turn_to_heading(200,DEGREES)
drivetrain.drive_for(FORWARD, 150, MM,wait=True)

# score the 3rd and the 4th purple block
drivetrain.drive_for(REVERSE, 137, MM,wait=True)

```

```

drivetrain.turn_to_heading(-90,DEGREES)
arm_motor_group.spin_to_position(0,DEGREES,wait=False)
# adjust the direction once afte specified distance
for i in range(2):
    drivetrain.drive_for(REVERSE, 100*5, MM,wait=True)
    drivetrain.turn_to_heading(-90,DEGREES)
drivetrain.drive_for(REVERSE, 50, MM,wait=True)

drivetrain.turn_to_heading(-30,DEGREES)
drivetrain.drive_for(REVERSE, 65, MM)
intake_motor_group.spin(FORWARD)
drivetrain.turn_to_heading(-45.5,DEGREES)
vr_thread(outIntake5)
# adjust the direction once afte specified distance
for i in range(3):
    drivetrain.drive_for(FORWARD, 100*4, MM,wait=True)
    drivetrain.turn_to_heading(-45.5,DEGREES)
drivetrain.drive_for(FORWARD, 150, MM,wait=True)

drivetrain.drive_for(REVERSE, 115, MM)
intake_motor_group.spin(FORWARD)
arm_motor_group.spin_to_position(0,DEGREES,wait=True)
arm_motor_group.spin_for(FORWARD,280,DEGREES,wait=True)
vr_thread(outIntake6)
drivetrain.drive_for(FORWARD, 100, MM)
wait(0.4,SECONDS)

# score the 5th and the 6th purple block, push the 3rd red block
drivetrain.turn_to_heading(45,DEGREES)
arm_motor_group.spin_to_position(0,DEGREES,wait=False)
drivetrain.drive_for(FORWARD, 100, MM)
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.drive_for(FORWARD, 410, MM)
drivetrain.turn_to_heading(135,DEGREES)
intake_motor_group.spin(FORWARD)
drivetrain.drive_for(FORWARD, 360, MM)
drivetrain.turn_to_heading(80,DEGREES)
drivetrain.turn_to_heading(131.7,DEGREES)
vr_thread(outIntake7)
for i in range(2):
    drivetrain.drive_for(FORWARD, 132.5*5, MM,wait=True)
    drivetrain.turn_to_heading(131.7,DEGREES)
wait(0.3,SECONDS)
drivetrain.turn_to_heading(142,DEGREES)

```

```

drivetrain.drive_for(REVERSE, 100, MM,wait=True)
intake_motor_group.spin(FORWARD)
arm_motor_group.spin_to_position(0,DEGREES,wait=True)
vr_thread(outIntake8)
drivetrain.drive_for(FORWARD, 90, MM)
wait(1.2,SECONDS)

# score the 7th purple block
drivetrain.turn_to_heading(100,DEGREES,wait=True)
arm_motor_group.spin_to_position(40,DEGREES,wait=False)
drivetrain.turn_to_heading(-10,DEGREES)
intake_motor_group.spin(FORWARD)
drivetrain.drive_for(FORWARD, 240, MM)
arm_motor_group.spin_for(FORWARD,280,DEGREES,wait=False)
drivetrain.turn_to_heading(146,DEGREES)
vr_thread(outIntake9)
drivetrain.drive_for(FORWARD, 245, MM)
wait(0.3,SECONDS)

# score the 8th purple block
drivetrain.turn_to_heading(90,DEGREES)
intake_motor_group.spin(FORWARD)
arm_motor_group.spin_to_position(55,DEGREES,wait=False)
drivetrain.drive_for(FORWARD, 155, MM)
drivetrain.turn_to_heading(0,DEGREES)
drivetrain.drive_for(FORWARD, 550, MM)
arm_motor_group.spin_to_position(80,DEGREES,wait=True)
drivetrain.drive_for(REVERSE, 120, MM)
arm_motor_group.spin_to_position(280,DEGREES,wait=False)
drivetrain.turn_to_heading(90,DEGREES)
drivetrain.turn_to_heading(180,DEGREES)
vr_thread(outIntake10)
drivetrain.drive_for(FORWARD, 420, MM)
wait(0.3,SECONDS)

# score the 9th purple block
drivetrain.turn_to_heading(-10,DEGREES)
intake_motor_group.spin(FORWARD)
arm_motor_group.spin_to_position(55,DEGREES,wait=False)
drivetrain.drive_for(FORWARD, 155, MM)
drivetrain.turn_to_heading(0,DEGREES)
drivetrain.drive_for(FORWARD, 400, MM)
arm_motor_group.spin_to_position(80,DEGREES,wait=True)
drivetrain.drive_for(REVERSE, 150, MM)

```

```
wait(0.3,SECONDS)
arm_motor_group.spin_to_position(280,DEGREES,wait=False)

drivetrain.turn_to_heading(170,DEGREES)
vr_thread(outIntake11)
drivetrain.drive_for(FORWARD, 430, MM)
wait(0.6,SECONDS)

# full park

arm_motor_group.spin_to_position(100, DEGREES,wait=False)
drivetrain.turn_to_heading(0,DEGREES)
intake_motor_group.spin(REVERSE)
drivetrain.drive_for(FORWARD, 495, MM,wait=True)
arm_motor_group.spin_to_position(-100, DEGREES)
drivetrain.drive_for(FORWARD, 560, MM)
#wait(1,SECONDS)
arm_motor_group.spin_to_position(100, DEGREES)

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

Thank you!