# ------------------------------------------

#

# Project: VEXcode Project

# Author: VEX

# Created:

# Description: VEXcode VR Python Project

#

# ------------------------------------------

# Library imports

from vexcode import \*

brain.clear()

def getLineBrightness(\*LineList):

global Csum,avg,Linebrg

avg=0

Csum=0

for Linebrg in LineList:

Csum+=float(Linebrg)

wait(5, MSEC)

length=len(LineList)

avg=Csum/length

brain.print(f"average brightness is {avg} for {length} lines")

# Add project code in "main"

def main():

global control,i,Turn\_Control

i=0

Turn\_Control=1

monitor\_sensor("left\_bumper.pressed")

drivetrain.drive(FORWARD)

colors=[]

while i<=4:

while not left\_bumper.pressed() :

control=1

#if brightness is less than 100,continue the loop

#and control the control variable

while down\_eye.brightness(PERCENT)<100:

#if control variable is not 0, add to colors list the brightness

if control!=0:

colors.append(down\_eye.brightness(PERCENT))

control=0

pass

wait(5, MSEC)

wait(5, MSEC)

#if bumper is true and i variable is 4, break the loop.

if left\_bumper.pressed() and i==4:

break

pass

#program is controlling the direction of robot

if Turn\_Control==1:

drivetrain.turn\_for(RIGHT, 90, DEGREES)

drivetrain.drive\_for(FORWARD, 400, MM)

drivetrain.turn\_for(RIGHT,90,DEGREES)

drivetrain.drive(FORWARD)

Turn\_Control=0

pass

else:

drivetrain.turn\_for(LEFT, 90, DEGREES)

drivetrain.drive\_for(FORWARD, 400, MM)

drivetrain.turn\_for(LEFT,90,DEGREES)

drivetrain.drive(FORWARD)

Turn\_Control=1

pass

i=i+1

wait(5, MSEC)

#run the function.

getLineBrightness(\*colors)

brain.new\_line()

brain.print(colors)

drivetrain.stop()

stop\_project()

# VR threads — Do not delete

vr\_thread(main())