import RPi.GPIO as GPIO

import time # import RPi.GPIO module

from time import sleep

from collections import deque

import numpy as np

import argparse

import imutils

import cv2

GPIO.setmode(GPIO.BOARD) # choose BCM or BOARD

GPIO.setup(21, GPIO.OUT)

GPIO.setup(15, GPIO.OUT)

GPIO.setup(23, GPIO.OUT)

GPIO.setup(13, GPIO.OUT)# set GPIO24 as an output

GPIO.setwarnings(False)

def lft(t):

GPIO.output(13 ,1)

GPIO.output(21, 1)

sleep(t)

GPIO.output(21, 0)

GPIO.output(13, 0)

def rght(t):

GPIO.output(15, 1)

GPIO.output(23, 1)

sleep(t)

GPIO.output(23, 0)

GPIO.output(15, 0)

def fwrd(t):

GPIO.output(21, 1)

GPIO.output(23, 1)

sleep(t)

GPIO.output(21, 0)

GPIO.output(23, 0)

def bkwrd(t):

GPIO.output(13 ,0)

GPIO.output(15, 0)

sleep(t)

GPIO.output(13, 0)

GPIO.output(15, 0)

ball\_found = False

camera = 0

def find\_ball(ball\_found):

greenLower = (0,84,150)

greenUpper = (31, 255, 255)

camera = cv2.VideoCapture(0)

while (ball\_found == False):

(grabbed,frame) = camera.read()

frame = imutils.resize(frame, width=600)

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, greenLower, greenUpper)

mask = cv2.erode(mask, None, iterations=2)

mask = cv2.dilate(mask, None, iterations=2)

cnts = cv2.findContours(mask.copy(), cv2.RETR\_EXTERNAL,

cv2.CHAIN\_APPROX\_SIMPLE)[-2]

center = None

if len(cnts) > 0:

c = max(cnts, key=cv2.contourArea)

((x, y), radius) = cv2.minEnclosingCircle(c)

M = cv2.moments(c)

center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))

if radius > 10:

cv2.circle(frame, (int(x), int(y)), int(radius),

(0, 255, 255), 2)

cv2.circle(frame, center, 5, (0, 0, 255), -1)

a,b,r = int(x),int(y),int(radius)

ball\_found = True

camera.release()

print(a,b,r)

turn\_to\_ball(a,b,r)

def turn\_to\_ball(x,y,r):

if( x > 360 and x < 500):

rght(0.04)

print("right")

elif( x < 240 and x > 100):

lft(0.08)

print("left")

elif( x > 450 ):

rght(0.04)

print("mega right")

elif( x < 150 ):

lft(0.08)

print("mega left")

else:

pass

ball\_found = False

goto\_ball(y,r)

def goto\_ball(y,g):

if(g < 40 and g > 15):

fwrd(0.5)

elif(g > 31 and g < 25):

fwrd(0.5)

elif(g > 25 and g < 19):

fwrd(0.5)

elif(g < 15):

fwrd(0.5)

elif(g > 40):

fwrd(1)

find\_ball(False)

ball\_found = False

print(g)

def find\_goal(goal\_found,ball\_found):

goal\_lower = (135,30,150)

goal\_upper = (185,245,245)

time.sleep(0.05)

# camera = cv2.VideoCapture(0)

while goal\_found == False and ball\_found == True :

grabbed,frame = camera.read(0)

frame = imutils.resize(frame, width=600)

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, goal\_lower, goal\_upper)

mask = cv2.erode(mask, None, iterations=2)

mask = cv2.dilate(mask, None, iterations=2)

cnts = cv2.findContours(mask.copy(), cv2.RETR\_EXTERNAL,

cv2.CHAIN\_APPROX\_SIMPLE)[-2]

center = None

if len(cnts) > 0:

c = max(cnts, key=cv2.contourArea)

((x, y), radius) = cv2.minEnclosingCircle(c)

M = cv2.moments(c)

center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m01"]/ M["m00"]))

if radius > 10:

cv2.circle(frame, (int(x), int(y)), int(radius),

(0, 255, 255), 2)

cv2.circle(frame, center, 5, (0, 0, 255), -1)

ga,gb,gr = int(x),int(y),int(radius)

goal\_found = True

turn\_to\_goal(ga,gb,gr)

print(ga,gb,gr)

def turn\_to\_goal(x,y,r):

if( x > 360 and x < 500):

rght(0.03)

print("right")

elif( x < 240 and x > 100):

lft(0.03)

print("left")

elif( x > 450 ):

rght(0.07)

print("mega right")

elif( x < 150 ):

lft(0.07)

print("mega left")

else:

pass

ball\_found = True

goto\_ball(x,y,r)

def goto\_goal(p,q,g):

if(g < 200 and g > 65):

fwrd(0.1)

elif(g > 55 and g < 65):

fwrd(0.2)

elif(g > 30 and g < 55):

fwrd(0.8)

elif(g < 30):

fwrd(1)

else:

pass

def man():

ball\_found = False

while(1):

find\_ball(ball\_found)

man()