ДОДАТОК А

Код програми

```
import numpy as np
from PIL import ImageGrab
import cv2
from time import *
from test_key import PressKey, ReleaseKey, W, A, S, D

t = process_time()
print(t)

x1, y1 = 50, 500
x2, y2 = 500, 500

m = 0
m1 = 0

def dlines(screen, lines):
    global x1, y1
    try:
        for line in lines:
            cor = line[0]
            cv2.line(screen, (cor[0], cor[1]), (cor[2], cor[3]), [0, 255, 50], 5)
            # обчислення центру лінії
            x1 = (cor[0] + cor[2]) // 2
            y1 = (cor[1] + cor[3]) // 2
    except:
        pass

def dlines1(screen, lines1):
    global x2, y2
    try:
        for line in lines1:
            cor = line[0]
            cv2.line(screen, (cor[0], cor[1]), (cor[2], cor[3]), [0, 255, 50], 5)
            # обчислення центру лінії
            x2 = (cor[0] + cor[2]) // 2
            y2 = (cor[1] + cor[3]) // 2
    except:
        pass

def maskk(screen, vertex):
    mask = np.zeros_like(screen)
```

```
cv2.fillPoly(mask, vertex, 255)
    mask1 = cv2.bitwise_and(screen, mask)
    return mask1

def calculate_slope(x1, y1, x2, y2):
    if x2 - x1 == 0:
        return 0
    return (y2 - y1) / (x2 - x1)

while True:
    window = np.array(ImageGrab.grab(bbox=(10, 10, 810, 730)))
    t = process_time()
    im = cv2.cvtColor(window, cv2.COLOR_BGR2GRAY)
    window1 = cv2.cvtColor(window, cv2.COLOR_BGR2RGB)
    Can = cv2.Canny(im, 100, 300, 7)
    Can = cv2.GaussianBlur(Can, (5,5), 0)

    vertex1 = np.array([[10, 590], [100, 400], [250, 360], [350, 590]])
    vertex2 = np.array([[500, 590], [500, 400], [750, 360], [790, 590]])

    Can1 = maskk(Can, [vertex1])
    Can2 = maskk(Can, [vertex2])

    lines = cv2.HoughLinesP(Can1, 2, np.pi/180, 200, 50, 50)
    lines1 = cv2.HoughLinesP(Can2, 2, np.pi/180, 200, 70, 70)

    dlines(window1, lines)
    dlines1(window1, lines1)

    # Малюємо фіолетові лінії із використанням центру знайдених зелених ліній
    cv2.line(window1, (x1, y1), (200, 590), [100, 0, 100], 5)
    cv2.line(window1, (x2, y2), (500, 590), [100, 0, 100], 5)

    # обчислення відстаней
    d1 = int(np.sqrt((x1-200)**2+(y1-590)**2))
    d2 = int(np.sqrt((x2-500)**2+(y2-590)**2))

    if d1 != m and d2 == m1:
        PressKey(D)
        PressKey(W)
        sleep(0.5)
        ReleaseKey(D)
        ReleaseKey(W)
        ReleaseKey(S)
```

```
        ReleaseKey(A)
        sleep(0.3)
    elif d2 != m1 and d1 == m:
        PressKey(A)
        PressKey(W)
        sleep(0.5)
        ReleaseKey(D)
        ReleaseKey(W)
        ReleaseKey(S)
        ReleaseKey(A)
        sleep(0.3)
    elif d2 == m1 and d1 == m:
        PressKey(S)
        sleep(0.5)
        ReleaseKey(S)
        ReleaseKey(W)
        ReleaseKey(D)
        ReleaseKey(A)
        sleep(0.5)
    else:
        PressKey(W)
        sleep(1)
        ReleaseKey(W)
        sleep(0.6)
        PressKey(S)
        sleep(0.2)
        ReleaseKey(S)
        ReleaseKey(D)
        ReleaseKey(A)


    cv2.putText(window1, " " + str(d1) + " ", (10,50), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 0, 255), 2)
    cv2.putText(window1, " " + str(d2) + " ", (700,50), cv2.FONT_HERSHEY_SIMPLEX,
1, (0, 0, 255), 2)
    cv2.imshow("window_new", window1)
    m = d1
    m1 = d2

    if cv2.waitKey(30) == ord("q"):
        cv2.destroyAllWindows()
        break
```

```
import ctypes
import time

SendInput = ctypes.windll.user32.SendInput
Q = 0x10
W = 0x11
A = 0x1E
S = 0x1F
D = 0x20
E = 0x12
R = 0x13
T = 0x14
Y = 0x15
C = 0x2E

RShift = 0x36
LControl = 0x1D
Space = 0x39
Left = 0xCB
Right = 0xCD
Up = 0xC8
Down = 0xD0
Escape = 0x01
Tab = 0x0F
Alt = 0x38
Enter = 0x1C

# C struct redefinitions
PUL = ctypes.POINTER(ctypes.c_ulong)
class KeyBdInput(ctypes.Structure):
    _fields_ = [("wVk", ctypes.c_ushort),
                ("wScan", ctypes.c_ushort),
                ("dwFlags", ctypes.c_ulong),
                ("time", ctypes.c_ulong),
                ("dwExtraInfo", PUL)]

class HardwareInput(ctypes.Structure):
    _fields_ = [("uMsg", ctypes.c_ulong),
                ("wParamL", ctypes.c_short),
                ("wParamH", ctypes.c_ushort)]

class MouseInput(ctypes.Structure):
    _fields_ = [("dx", ctypes.c_long),
                ("dy", ctypes.c_long),
                ("mouseData", ctypes.c_ulong),
```

```
                ("dwFlags", ctypes.c_ulong),
                ("time",ctypes.c_ulong),
                ("dwExtraInfo", PUL)]
class Input_I(ctypes.Union):
    _fields_ = [("ki", KeyBdInput),
                 ("mi", MouseInput),
                 ("hi", HardwareInput)]

class Input(ctypes.Structure):
    _fields_ = [("type", ctypes.c_ulong),
                ("ii", Input_I)]

# Actuals Functions

def PressKey(hexKeyCode):

    extra = ctypes.c_ulong(0)
    ii_ = Input_I()
    ii_.ki = KeyBdInput(0, hexKeyCode, 0x0008, 0, ctypes.pointer(extra) )
    x = Input( ctypes.c_ulong(1), ii_ )
    ctypes.windll.user32.SendInput(1, ctypes.pointer(x), ctypes.sizeof(x))

def ReleaseKey(hexKeyCode):

    extra = ctypes.c_ulong(0)
    ii_ = Input_I()
    ii_.ki = KeyBdInput(0, hexKeyCode, 0x0008|0x0002, 0, ctypes.pointer(extra) )
    x = Input( ctypes.c_ulong(1), ii_ )
    ctypes.windll.user32.SendInput(1, ctypes.pointer(x), ctypes.sizeof(x))
```

ДОДАТОК Б
Демонстраційний матеріал

№	Позначення	Найменування	Дод. відомості
		Текстові документи	
1	ГЮИК. ХХХХХХ.ХХ ПЗ	Пояснювальна записка	77 с., ф. А4
2		Код програми	5 с., ф. А4
3		Демонстраційний матеріал	12 с., у форматі *.ppt

Змін.	Арк	№ документа	Підп.	Дата	ГЮИК. ХХХХХХ.ХХ ВД			
Розроб.		Столяренко Н.В			Розроблення інтерфейсів для керування роботами з використанням штучного інтелекту	Літ.	Аркуш	Аркушів
Перевір.		Хрустальова С.В						
Н. контр.		Демська Н.П				ХНУРЕ Кафедра КІТАР		
Затвер.		Невлюдов І.Ш						