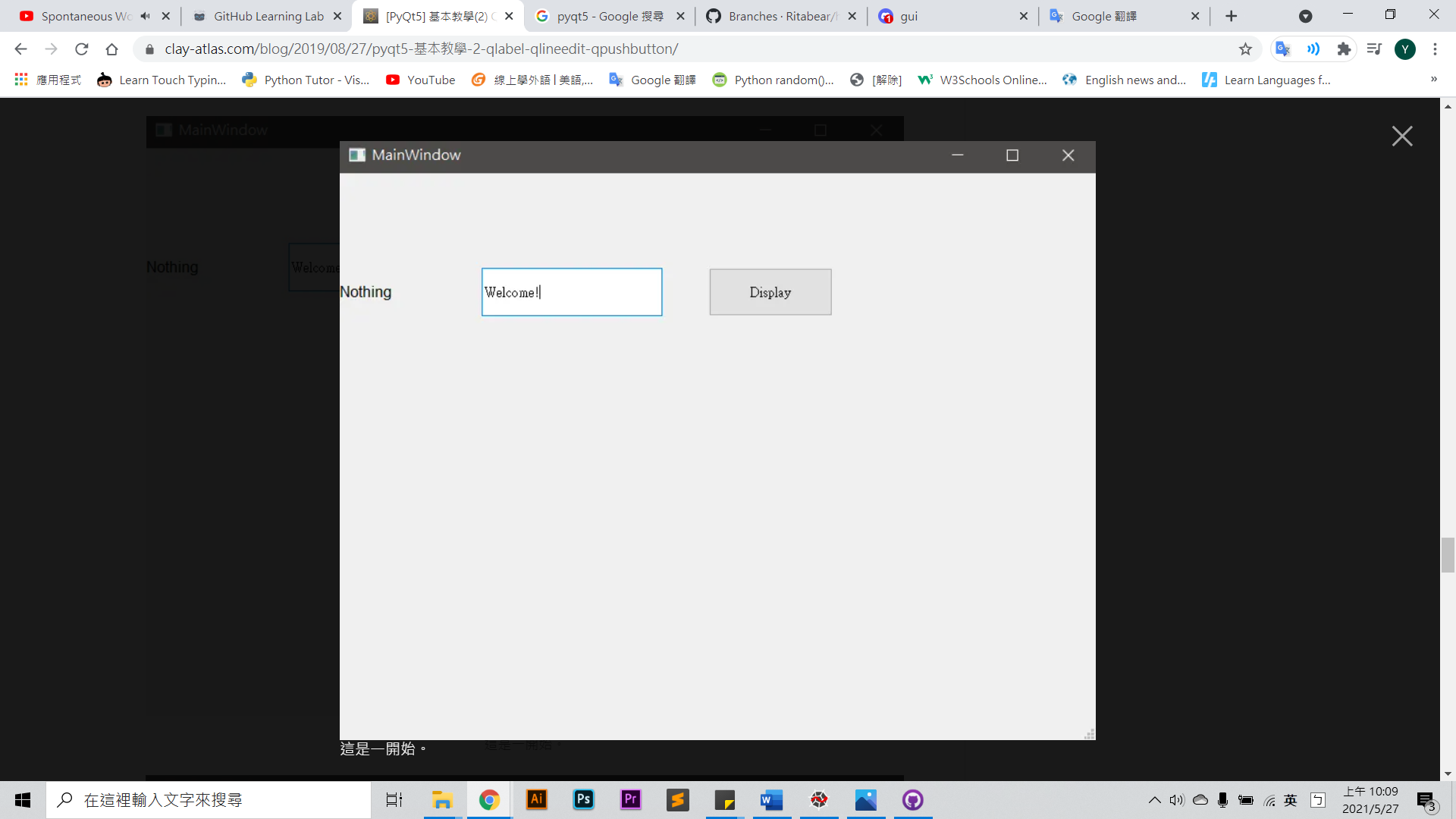
from PyQt5.QtCore import Qt

from PyQt5.QtGui import (QPainter, QColor, QPen)

import sys

from PyQt5.QtWidgets import (QApplication,QWidget,QLabel)



class Example(QWidget):

def \_\_init\_\_(self):

super(Example, self).\_\_init\_\_()

self.initUi()

#預設情況下禁用滑鼠跟蹤， 如果啟用滑鼠跟蹤，即使沒有按鈕被按下，小部件也會接收滑鼠移動事件。

#當然你也可以不寫，只需要在執行的過程中按照滑鼠左鍵也行

self.setMouseTracking(True)

def initUi(self):

self.setGeometry(400,300,400,300)

self.setWindowTitle("鍵盤響應事件")

self.lab1 = QLabel("方向",self)

self.lab1.setGeometry(200,150,100,100)

self.lab2 = QLabel("顯示滑鼠座標", self)

self.lab2.setGeometry(200, 80, 100, 100)

"""重定義鍵盤事件"""

def keyPressEvent(self,e ):

if e.key() == Qt.Key\_Up:

self.lab1.setText("↑")

elif e.key() == Qt.Key\_Down:

self.lab1.setText("↓")

elif e.key() == Qt.Key\_Left:

self.lab1.setText("←")

else:

self.lab1.setText("→")

"""重定義滑鼠單擊事件"""

def mousePressEvent(self, event):

if event.button() == Qt.LeftButton:

self.lab1.setText("滑鼠左鍵點選！")

# print(event.pos().x(),event.pos().y())

if event.button() == Qt.RightButton:

self.lab1.setText("滑鼠右鍵點選！")

"""當滑鼠左鍵點選拖動時觸發事件,有無if判斷條件效果都一樣"""

def mouseMoveEvent(self, event):

# if event.buttons() == Qt.LeftButton:

# # print(type(event.pos().x())) #<class 'int'>

# self.lab2.setText(str(event.pos().x())+","+str(event.pos().y()))

self.pos = event.pos()

print(self.pos)

self.lab2.setText(str(event.pos().x()) + "," + str(event.pos().y()))

self.update()

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = Example()

ex.show()

sys.exit(app.exec\_()

import time

import sys

from PyQt5.QtCore import \*

from PyQt5.QtGui import \*

from PyQt5.QtWidgets import \*

from cv2 import \*

from PyQt5.QtCore import QTimer

class VideoBox(QWidget):

VIDEO\_TYPE\_OFFLINE = 0

VIDEO\_TYPE\_REAL\_TIME = 1

STATUS\_INIT = 0

STATUS\_PLAYING = 1

STATUS\_PAUSE = 2

video\_url = ""

def \_\_init\_\_(self, video\_url="", video\_type=VIDEO\_TYPE\_OFFLINE, auto\_play=False):

QWidget.\_\_init\_\_(self)

self.video\_url = video\_url

self.video\_type = video\_type # 0: offline 1: realTime

self.auto\_play = auto\_play

self.status = self.STATUS\_INIT # 0: init 1:playing 2: pause

# 组件展示

self.pictureLabel = QLabel()

# init\_image = QPixmap("wzf.jpg").scaled(self.width(), self.height())

# self.pictureLabel.setPixmap(init\_image)

self.playButton = QPushButton()

self.playButton.setEnabled(True)

self.playButton.setIcon(self.style().standardIcon(QStyle.SP\_MediaPlay))

self.playButton.clicked.connect(self.entry\_info)

control\_box = QHBoxLayout()

control\_box.setContentsMargins(0, 0, 0, 0)

control\_box.addWidget(self.playButton)

layout = QVBoxLayout()

layout.addWidget(self.pictureLabel)

layout.addLayout(control\_box)

self.setLayout(layout)

self.playCapture = VideoCapture(0)

def entry\_info(self):

print('已完成信息录入')

def show\_video\_images(self):

if self.playCapture.isOpened():

success, frame = self.playCapture.read()

print('frame.shape:',frame.shape)

# frame = cvtColor(frame, COLOR\_BGR2RGB)

if success:

height, width = frame.shape[:2]

if frame.ndim == 3:

rgb = cvtColor(frame, COLOR\_BGR2RGB)

elif frame.ndim == 2:

rgb = cvtColor(frame, COLOR\_GRAY2BGR)

temp\_image = QImage(rgb.flatten(), width, height, QImage.Format\_RGB888)

temp\_pixmap =QPixmap.fromImage(temp\_image)

self.pictureLabel.setPixmap(temp\_pixmap)

else:

print("read failed, no frame data")

return

else:

print("open file or capturing device error, init again")

if \_\_name\_\_ == "\_\_main\_\_":

mapp = QApplication(sys.argv)

mw = VideoBox()

timer = QTimer()

timer.timeout.connect(mw.show\_video\_images) #计时结束调用operate()方法

timer.start(100) #设置计时间隔并启动

mw.show()

sys.exit(mapp.exec\_())