Indian Institute of Technology Jodhpur
Autonomous System
Assignment-2: Reactive Programming

Submitted To:
Dr. Hiranmay Ghosh

Submitted By:
Dipinti Manandhar (M20CS020)

Question:

Write a code to detect mouse clicks and distinguish between single and double clicks.

Description:

Asynchronous reactive programming in python using Rx library is written for distinguishing the single and double clicks. The source code is given below. A time-out period of 1 second is defined such that after 1 second, the click event is considered as a new click, and the process of determining whether it is single or double occurs. After a single click, if no click even is done within 1 sec then it is considered to be a single click. When there is more than one click in one second, then it is considered to be a double click as multiple clicks (more than two) are ignored and considered to be a double click event. We used PyQt5 library for building GUI in python. Some of the modules used in this project are QtCore, QLabel, QApplication

Libraries to be installed before running the code:

pip install rx
pip install PyQt5

The process to run the code:

1. Go to command prompt
2. Write command " python M20CS020_AS1.py "
3. A Window will open then we need to hit single and double click events and the detected result is printed in the command prompt

Source code (written in Python):

```
#---------------------------------importing
libraries-------------------------------------------------------------------------
------------

import os
import sys
import rx
from rx.scheduler.mainloop import QtScheduler
from rx.subject import Subject
import rx.operators as ops
from PyQt5 import QtCore
from PyQt5.QtCore import QSize
# from PyQt5.QtWidgets import QLabel
from PyQt5.QtWidgets import QApplication, QLabel,QWidget

# --------------------------------Defining length of buffer (timeout) to distinguish
between single and double click-----------------------------

TIME = 1 #in seconds
#-------------------------Creating                  class                   MouseEvent
-------------------------------------------------------------------------------------
----------------

class MouseEvent(QWidget):

    def __init__(self,parent = None):
        QWidget.__init__(self)

#-----Providing
Title-----------------------------------------------------------------------------
--------------------------------------

        self.setWindowTitle("Distinguish Single and Double Clicks M20CS020")
        self.setFixedSize(QSize(800, 800))

#-----Display                    text                    inside                    the
window----------------------------------------------------------------------------
--------------------------

         self.label = QLabel('Click anywhere inside this window to distinguish whether
it is single or double click and check the result in command prompt.', self)
```

```
#-------Setting                                                                    some
styles-------------------------------------------------------------------------------
---------------------------------

                self.label.setStyleSheet("background-color:  pink;  border:  1px  solid
red;padding :5px; color:red")
        self.label.move(30, 350)

#---------Creating                                                                reactive
environment--------------------------------------------------------------------------
---------------------------

        self.checkmouse = Subject()

#----------Overriding                          mousePressEvent                          and
mouseReleaseEvent--------------------------------------------------------------------
-------------

    def mousePressEvent(self, event):
        self.checkmouse.on_next((event.pos(), 1))
    def mouseReleaseEvent(self, event):
       self.checkmouse.on_next((event.pos(), 2))

#---------Funstion   to   detect   whether   the   click   is   single   click   or   double
click--------------------------------------------------------------

def determineClick(data):
    length_data=len(data)
    if (length_data==2):
        print('Single click detected')
    elif (length_data>2):
        print('Double click detected')




#----------------Use                  main                  for                  calling
function-----------------------------------------------------------------------------
--------------------

if __name__ == '__main__':
# start_event()
    application = QApplication(sys.argv)
#-----------creating            a            scheduler            for            event
loop---------------------------------------------------------------------------------
```

```
----------
    scheduling_ = QtScheduler(QtCore)
#----------measuring                              time                              between
events-------------------------------------------------------------------------------------
------------------
    time_calculation = ops.time_interval(scheduling_)
#----------determine                                                                 timing
information--------------------------------------------------------------------------------
-------------------------
    timing_info = ops.buffer_with_time(TIME)
    mapper = ops.map(determineClick)
    window_ = MouseEvent("800x800")
#----------show/display
window-------------------------------------------------------------------------------------
----------------------------
    window_.show()

    window_.checkmouse.pipe(time_calculation, timing_info, mapper).subscribe(
        on_next=(lambda x: x), scheduler=scheduling_
        )
#----------end           the           main           loop           of           the
application--------------------------------------------------------------------------------
--------
    sys.exit(application.exec_())
```

Output

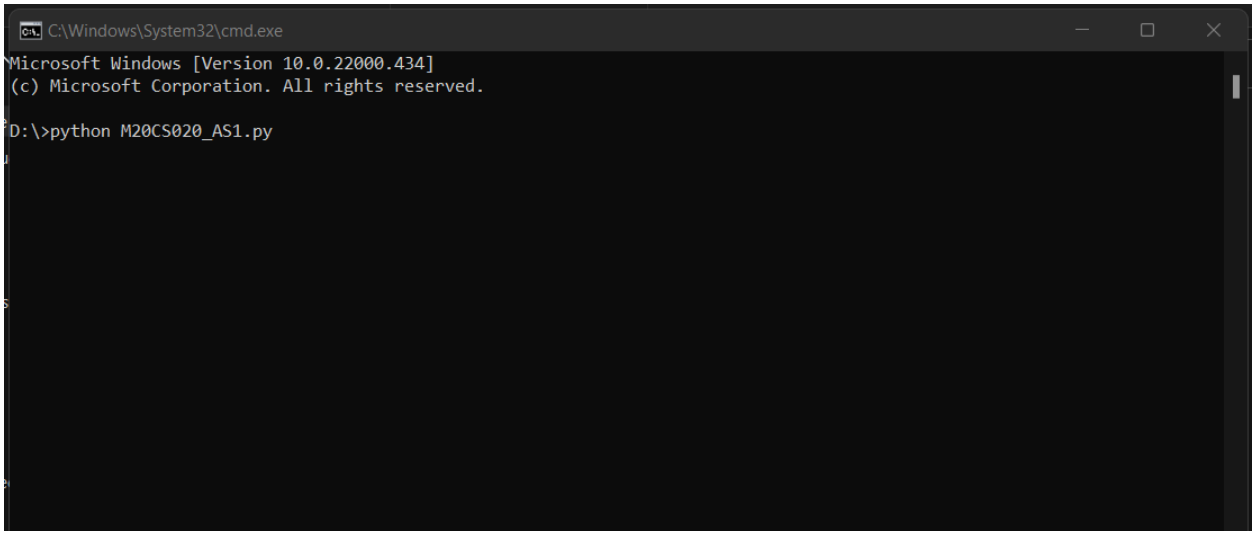

Fig: command to run the python file

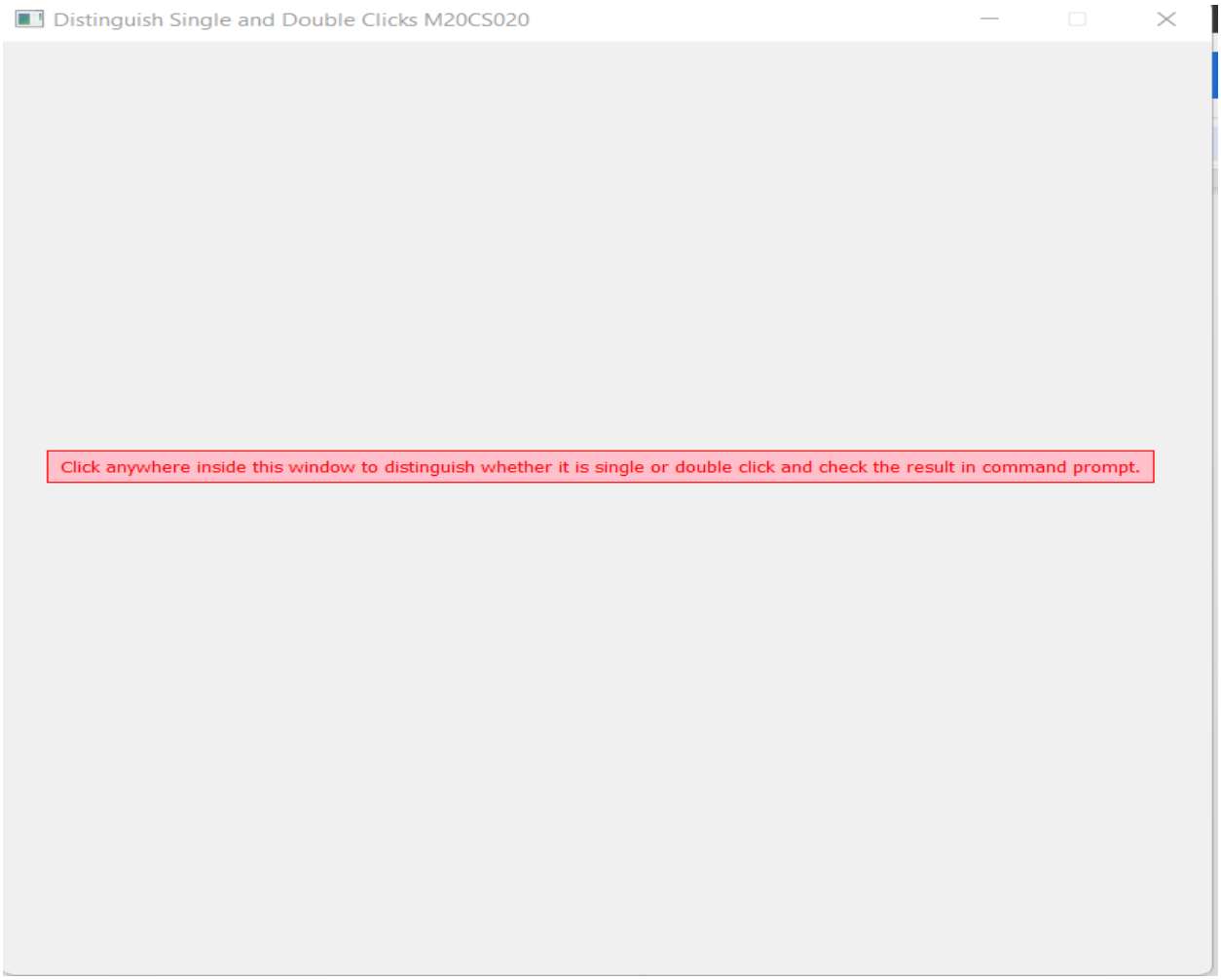

Fig: Window opened to perform mouse click events

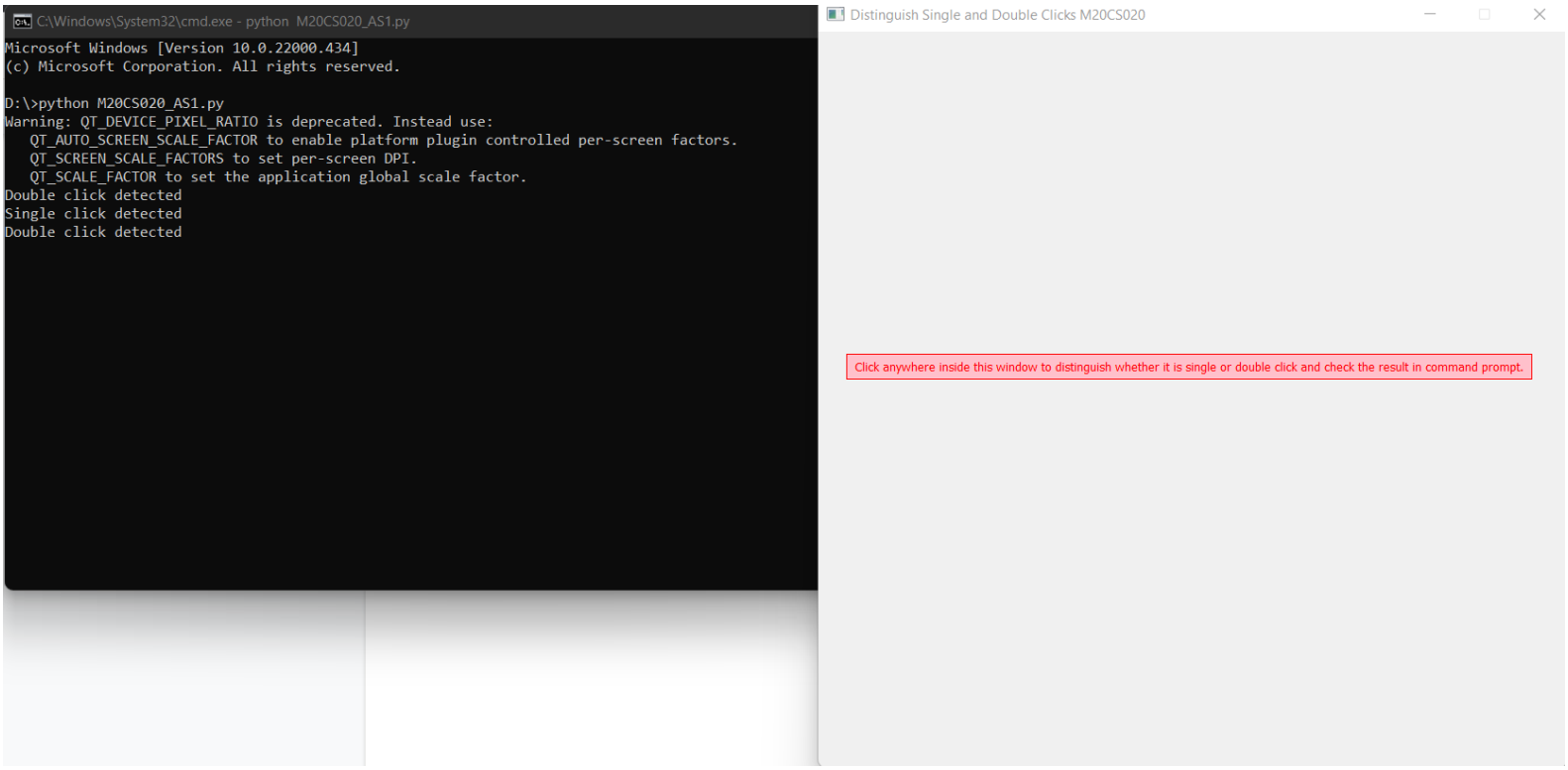

Figure: Output printed on command window after detecting single and double clicks

Conclusion:

Therefore, by performing this programming assignment, we became familiar with the principles of asynchronous programming and such a programming environment. We also became familiar with the Rx library reactive programming environment.