COMPSYS302: PyQt5

4. Advanced skills

Ho Seok AHN

hs.ahn@auckland.ac.nz

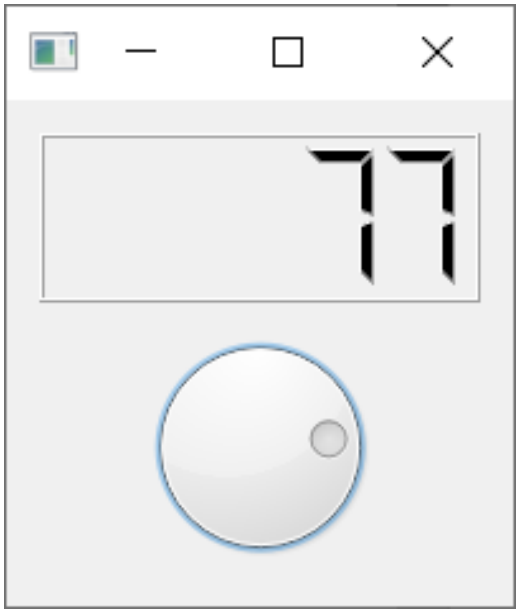
THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

4.1 Connect

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

Connect

- PyQt uses "**Signal & Slot**" mechanism for event handling.



  - Ex) 4_01_Connect.py

```
lcd = QLCDNumber(self)
dial = QDial(self)
```

➔ **QLCDNumber**(): Add a LCD widget.

➔ **QDial**(): Add a dial widget.

```
dial.valueChanged.connect(lcd.display)
```

➔ When the value of dial is changed, connect it to display of lcd.

➔ Here, "valueChanged" is a **signal**, and "display" is a **slot**.
Also "dial" is a **sender**, and "lcd" is a **receiver**.

4.2 Event Handler

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

Create Event Handler

- Add events with push buttons.

- Ex) 4_02_EventHandler.py

btn1.**clicked**.connect(self.resizeBig)

btn2.**clicked**.connect(self.resizeSmall)

➔ When the button is clicked, connect it to each button.

self.**resize**(width, height)

➔ Resize the window.

4.2 Event Handler

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

Existing Event Handlers

- PyQt supports different event handlers.
  - keyPressEvent: any keys on the keyboard are pressed.
  - keyReleaseEvent: any keys on the keyboard are released.
  - mousePressEvent: when mouse button is pressed.
  - mouseDoubleClickEvent: when double-click the mouse.
  - mouseMoveEvent: when mouse cursor is moved.
  - mouseReleaseEvent: when mouse button is released.
  - moveEvent: when widget is moved.
  - resizeEvent: when the size of widget is changed.

4.2 Event Handler

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

☐ Keyboard Event Handlers

- Ex) 4_03_Keyboard.py
  - When 'esc' is pressed, close the window.
  - When 'F' is pressed, make the window full screen.
  - When 'N' is pressed, make the window size as normal.
  - Use keyPressEvent() event handler.

def **keyPressEvent**(self, **e**):

➔ No need to connect it between signal and slot.

➔ Just use the existing event handler.

if **e**.key() == Qt.**Key_Escape**:

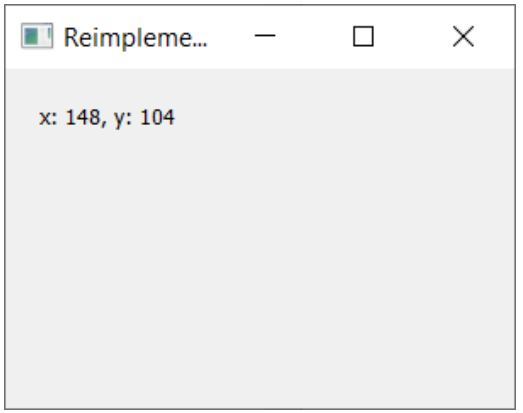
➔ Check whether the event is 'esc' pressed.

➔ Same for all other keys.

4.2 Event Handler

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

Mouse Event Handlers



- Ex) 4_04_Mouse.py
  - Tracking mouse positions.
  - Use mouseMoveEvent() event handler.

self.**setMouseTracking**(True)

➔ Set ON tracking mouse position.

def **mouseMoveEvent**(self, **e**):

➔ No need to connect it between signal and slot.

➔ Just use the existing event handler.

x = **e**.x()

y = **e**.y()

➔ Get the x and y position on the window.

➔ Use e.globalX() and e.globalY() for getting the position on the whole screen.

4.2 Event Handler

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

☐ User-defined Event Handlers

- We can define the event handlers.
  - Ex) 4_05_UserdefinedEventHandler.py
    - Close the window when mouse button is pressed by using **pyqtSignal**() event handler.

```
class Communicate(QObject):
    closeApp = pyqtSignal()
```

➔ Create a signal 'closeApp' under Communicate() class.

➔ Should import **pyqtSignal**().

```
self.c.closeApp.connect(self.close)
```

➔ When 'closeApp' signal is emitted, connect it to close().

```
def mousePressEvent(self, e):
    self.c.closeApp.emit()
```

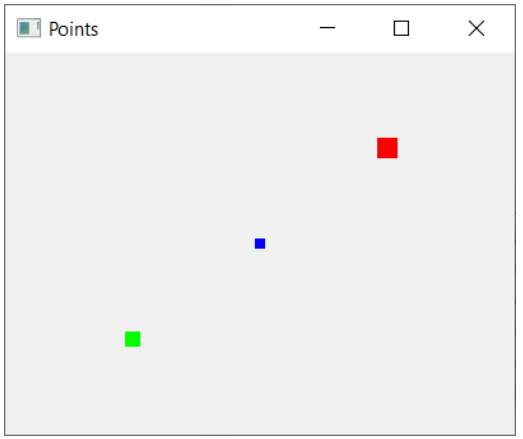
➔ When mouse button is pressed, to emit 'closeApp' signal.

4.3 Painting

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

drawPoint()

- Qpainter supports drawing related functions for GUI.
  - Ex) 4_06_drawPoint1.py



qp.**setPen**(QPen(Qt.blue, 8))

➔ Set the colour and size of the pen.

qp.**drawPoint**(x, y)

➔ Draw a point to the (x, y).

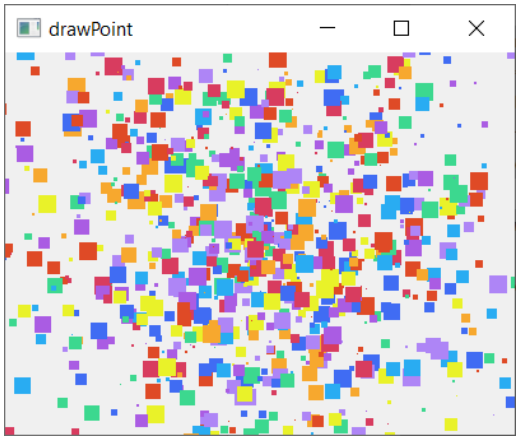
➔ Can get the width and height of the window using self.**width**() and self.**height**().

4.3 Painting

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

drawPoint()

- Draw 1000 x random points.



  - Ex) 4_06_drawPoint2.py

```
colors = ['#D83C5F', '#3CD88F', '#AA5CE3',
          '#DF4A26', '#AE85F6', '#F7A82E',
          '#406CF3', '#E9F229', '#29ACF2']
```

➔ Set random colours of the pen.

```
pen.setWidth(np.random.randint(1, 15))
pen.setColor(QColor(np.random.choice(colors)))
```

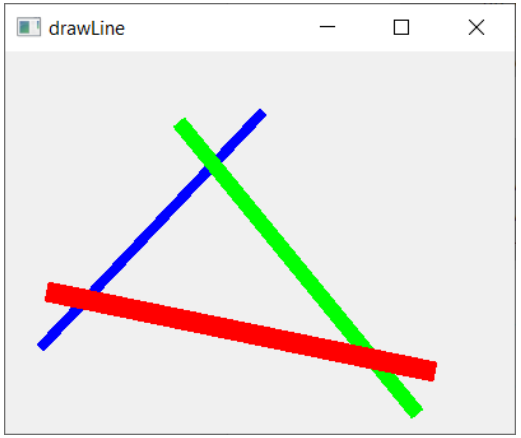
➔ Set the width and colour of the pen randomly.

4.3 Painting

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

drawLine()

- Draw different lines.



  - Ex) 4_07_drawLine1.py

    qp.**setPen**(QPen(Qt.blue, 8))

    ➔ Set the colour and size of the pen.

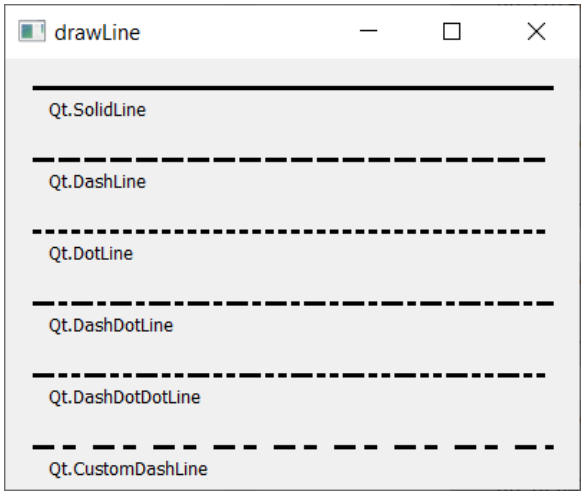
    qp.**drawLine**(x1, y1, x2, y2)

    ➔ Draw a line from the (x1, y1) to (x2, y2).

    ➔ Can get the width and height of the window using self.**width**() and self.**height**().

drawLine()

- Draw lines with different styles.
  - Ex) 4_07_drawLine2.py




```
qp.setPen(QPen(Qt.black, 3, Qt.SolidLine))
qp.setPen(QPen(Qt.black, 3, Qt.DashLine))
qp.setPen(QPen(Qt.black, 3, Qt.DotLine))
qp.setPen(QPen(Qt.black, 3, Qt.DashDotLine))
qp.setPen(QPen(Qt.black, 3, Qt.DashDotDotLine))
```

➔ Set different styles of lines.

```
pen = QPen(Qt.black, 3, Qt.CustomDashLine)
pen.setDashPattern([dash, space, dash, space])
```

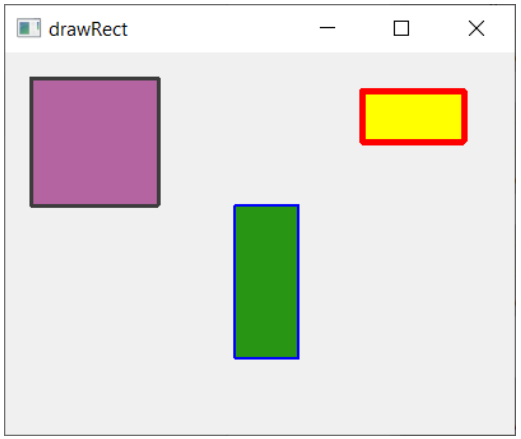
➔ Can set custom lines as well.

4.3 Painting

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

drawRect()

- Draw different rectangles.



- Ex) 4_08_drawRect1.py

qp.**setBrush**(QColor(180, 100, 160))

➔ Set the colour and size of the brush for area.

qp.**setPen**(QPen(QColor(60, 60, 60), 3))

➔ Set the colour and size of the pen.

qp.**drawRect**(x, y, width, height)

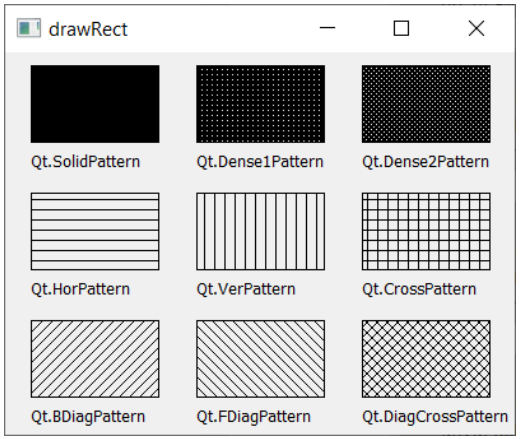
➔ Draw a rectangle.

4.3 Painting

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING

drawRect()

- Draw rectangles with different styles.



  - Ex) 4_08_drawRect2.py

```
brush = QBrush(Qt.SolidPattern)
brush = QBrush(Qt.Dense1Pattern)
brush = QBrush(Qt.Dense2Pattern)
brush = QBrush(Qt.HorPattern)
brush = QBrush(Qt.VerPattern)
brush = QBrush(Qt.CrossPattern)
brush = QBrush(Qt.BDiagPattern)
brush = QBrush(Qt.FDiagPattern)
brush = QBrush(Qt.DiagCrossPattern)
```

➔ Set different styles of rectangles.