

## eventtest1.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # eventtest1.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10: import PyQt5.QtCore
11:
12: window = None
13:
14: def red_button_slot():
15:     palette = window.palette()
16:     palette.setColor(window.backgroundRole(), PyQt5.QtCore.Qt.red)
17:     window.setPalette(palette)
18:     window.repaint()
19:
20: def green_button_slot():
21:     palette = window.palette()
22:     palette.setColor(window.backgroundRole(), PyQt5.QtCore.Qt.green)
23:     window.setPalette(palette)
24:     window.repaint()
25:
26: def blue_button_slot():
27:     palette = window.palette()
28:     palette.setColor(window.backgroundRole(), PyQt5.QtCore.Qt.blue)
29:     window.setPalette(palette)
30:     window.repaint()
31:
32: def main():
33:
34:     global window
35:
36:     app = PyQt5.QtWidgets.QApplication(sys.argv)
37:
38:     red_button = PyQt5.QtWidgets.QPushButton('red')
39:     green_button = PyQt5.QtWidgets.QPushButton('green')
40:     blue_button = PyQt5.QtWidgets.QPushButton('blue')
41:
42:     red_button.clicked.connect(red_button_slot)
43:     green_button.clicked.connect(green_button_slot)
44:     blue_button.clicked.connect(blue_button_slot)
45:
46:     layout = PyQt5.QtWidgets.QGridLayout()
47:     layout.addWidget(red_button, 0, 0)
48:     layout.addWidget(green_button, 0, 1)
49:     layout.addWidget(blue_button, 0, 2)
50:
51:     frame = PyQt5.QtWidgets.QFrame()
52:     frame.setLayout(layout)
53:
54:     window = PyQt5.QtWidgets.QMainWindow()
55:     window.setCentralWidget(frame)
56:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
57:     window.resize(screen_size.width()//2, screen_size.height()//2)
58:     window.setWindowTitle('Event Test 1')
59:     window.show()
60:
61:     sys.exit(app.exec_())
62:
63: if __name__ == '__main__':
64:     main()

```

## eventtest2.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # eventtest2.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10: import PyQt5.QtCore
11:
12: def main():
13:
14:     app = PyQt5.QtWidgets.QApplication(sys.argv)
15:
16:     red_button = PyQt5.QtWidgets.QPushButton('red')
17:     green_button = PyQt5.QtWidgets.QPushButton('green')
18:     blue_button = PyQt5.QtWidgets.QPushButton('blue')
19:
20:     def red_button_slot():
21:         palette = window.palette()
22:         palette.setColor(window.backgroundRole(), PyQt5.QtCore.Qt.red)
23:         window.setPalette(palette)
24:         window.repaint()
25:
26:     def green_button_slot():
27:         palette = window.palette()
28:         palette.setColor(window.backgroundRole(), PyQt5.QtCore.Qt.green)
29:         window.setPalette(palette)
30:         window.repaint()
31:
32:     def blue_button_slot():
33:         palette = window.palette()
34:         palette.setColor(window.backgroundRole(), PyQt5.QtCore.Qt.blue)
35:         window.setPalette(palette)
36:         window.repaint()
37:
38:     red_button.clicked.connect(red_button_slot)
39:     green_button.clicked.connect(green_button_slot)
40:     blue_button.clicked.connect(blue_button_slot)
41:
42:     layout = PyQt5.QtWidgets.QGridLayout()
43:     layout.addWidget(red_button, 0, 0)
44:     layout.addWidget(green_button, 0, 1)
45:     layout.addWidget(blue_button, 0, 2)
46:
47:     frame = PyQt5.QtWidgets.QFrame()
48:     frame.setLayout(layout)
49:
50:     window = PyQt5.QtWidgets.QMainWindow()
51:     window.setCentralWidget(frame)
52:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
53:     window.resize(screen_size.width()//2, screen_size.height()//2)
54:     window.setWindowTitle('Event Test 2')
55:     window.show()
56:
57:     sys.exit(app.exec_())
58:
59: if __name__ == '__main__':
60:     main()

```

## eventtestbad.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # eventtestbad.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10: import PyQt5.QtCore
11:
12: def main():
13:
14:     app = PyQt5.QtWidgets.QApplication(sys.argv)
15:
16:     red_button = PyQt5.QtWidgets.QPushButton('red')
17:     green_button = PyQt5.QtWidgets.QPushButton('green')
18:     blue_button = PyQt5.QtWidgets.QPushButton('blue')
19:
20:     def button_slot(color):
21:         palette = window.palette()
22:         palette.setColor(window.backgroundRole(), color)
23:         window.setPalette(palette)
24:         window.repaint()
25:
26:     red_button.clicked.connect(button_slot(PyQt5.QtCore.Qt.red))
27:     green_button.clicked.connect(button_slot(PyQt5.QtCore.Qt.green))
28:     blue_button.clicked.connect(button_slot(PyQt5.QtCore.Qt.blue))
29:
30:     layout = PyQt5.QtWidgets.QGridLayout()
31:     layout.addWidget(red_button, 0, 0)
32:     layout.addWidget(green_button, 0, 1)
33:     layout.addWidget(blue_button, 0, 2)
34:
35:     frame = PyQt5.QtWidgets.QFrame()
36:     frame.setLayout(layout)
37:
38:     window = PyQt5.QtWidgets.QMainWindow()
39:     window.setCentralWidget(frame)
40:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
41:     window.resize(screen_size.width()//2, screen_size.height()//2)
42:     window.setWindowTitle('Event Test 2')
43:     window.show()
44:
45:     sys.exit(app.exec_())
46:
47: if __name__ == '__main__':
48:     main()

```

## dialogwritemessage.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # dialogwritemessage.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10:
11: def main():
12:
13:     app = PyQt5.QtWidgets.QApplication(sys.argv)
14:
15:     button = PyQt5.QtWidgets.QPushButton('Show Dialog')
16:
17:     textedit = PyQt5.QtWidgets.QTextEdit()
18:
19:     layout = PyQt5.QtWidgets.QGridLayout()
20:     layout.addWidget(button, 0, 0)
21:     layout.addWidget(textedit, 1, 0)
22:
23:     frame = PyQt5.QtWidgets.QFrame()
24:     frame.setLayout(layout)
25:
26:     window = PyQt5.QtWidgets.QMainWindow()
27:     window.setWindowTitle('Write a message')
28:     window.setCentralWidget(frame)
29:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
30:     window.resize(screen_size.width()//2, screen_size.height()//2)
31:
32:     def button_slot():
33:         PyQt5.QtWidgets.QMessageBox.information(
34:             window, 'My title', 'My message')
35:         textedit.append('Dialog dismissed.')
36:
37:     button.clicked.connect(button_slot)
38:
39:     window.show()
40:     sys.exit(app.exec_())
41:
42: if __name__ == '__main__':
43:     main()

```

## dialogchooseoption.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # dialogchooseoption.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10:
11: def main():
12:
13:     app = PyQt5.QtWidgets.QApplication(sys.argv)
14:
15:     button = PyQt5.QtWidgets.QPushButton('Show Dialog')
16:
17:     textedit = PyQt5.QtWidgets.QTextEdit()
18:
19:     layout = PyQt5.QtWidgets.QGridLayout()
20:     layout.addWidget(button, 0, 0)
21:     layout.addWidget(textedit, 1, 0)
22:
23:     frame = PyQt5.QtWidgets.QFrame()
24:     frame.setLayout(layout)
25:
26:     window = PyQt5.QtWidgets.QMainWindow()
27:     window.setWindowTitle('Choose an option')
28:     window.setCentralWidget(frame)
29:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
30:     window.resize(screen_size.width()//2, screen_size.height()//2)
31:
32:     def button_slot():
33:         reply = PyQt5.QtWidgets.QMessageBox.question(
34:             window,
35:             'My title',
36:             'My message',
37:             buttons=(
38:                 PyQt5.QtWidgets.QMessageBox.Yes
39:                 | PyQt5.QtWidgets.QMessageBox.No))
40:         # Others: Ok, Open, Save, Cancel, Close, Discard,
41:         # Apply, Reset, RestoreDefaults, Help, SaveAll,
42:         # YesToAll, NoToAll, Abort, Retry, Ignore, NoButton
43:         if reply == PyQt5.QtWidgets.QMessageBox.Yes:
44:             text = 'Yes'
45:         elif reply == PyQt5.QtWidgets.QMessageBox.No:
46:             text = 'No'
47:         else:
48:             text = '(No option chosen)' # Unused!!!
49:         textedit.append(text)
50:
51:     button.clicked.connect(button_slot)
52:
53:     window.show()
54:     sys.exit(app.exec_())
55:
56: if __name__ == '__main__':
57:     main()

```

## dialogreadvalue.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # dialogreadvalue.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10:
11: def main():
12:
13:     app = PyQt5.QtWidgets.QApplication(sys.argv)
14:
15:     button = PyQt5.QtWidgets.QPushButton('Show Dialog')
16:
17:     textedit = PyQt5.QtWidgets.QTextEdit()
18:
19:     layout = PyQt5.QtWidgets.QGridLayout()
20:     layout.addWidget(button, 0, 0)
21:     layout.addWidget(textedit, 1, 0)
22:
23:     frame = PyQt5.QtWidgets.QFrame()
24:     frame.setLayout(layout)
25:
26:     window = PyQt5.QtWidgets.QMainWindow()
27:     window.setWindowTitle('Read a value')
28:     window.setCentralWidget(frame)
29:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
30:     window.resize(screen_size.width()//2, screen_size.height()//2)
31:
32:     def button_slot():
33:         reply, successful = PyQt5.QtWidgets.QInputDialog.getText(
34:             window, 'My title', 'My prompt')
35:         if successful:
36:             textedit.append(reply)
37:         else:
38:             textedit.append('(no reply)')
39:
40:     button.clicked.connect(button_slot)
41:
42:     window.show()
43:     sys.exit(app.exec_())
44:
45: if __name__ == '__main__':
46:     main()

```

## dialogchoosefile.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # dialogchoosefile.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10:
11: def main():
12:
13:     app = PyQt5.QtWidgets.QApplication(sys.argv)
14:
15:     button = PyQt5.QtWidgets.QPushButton('Show Dialog')
16:
17:     textedit = PyQt5.QtWidgets.QTextEdit()
18:
19:     layout = PyQt5.QtWidgets.QGridLayout()
20:     layout.addWidget(button, 0, 0)
21:     layout.addWidget(textedit, 1, 0)
22:
23:     frame = PyQt5.QtWidgets.QFrame()
24:     frame.setLayout(layout)
25:
26:     window = PyQt5.QtWidgets.QMainWindow()
27:     window.setWindowTitle('Choose a file')
28:     window.setCentralWidget(frame)
29:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
30:     window.resize(screen_size.width()//2, screen_size.height()//2)
31:
32:     def button_slot():
33:         # Second value returned is selected_filter.
34:         file_name, _ = PyQt5.QtWidgets.QFileDialog.getOpenFileName()
35:         if file_name == '':
36:             textedit.append('(empty string)')
37:         else:
38:             textedit.append(file_name)
39:
40:     button.clicked.connect(button_slot)
41:
42:     window.show()
43:     sys.exit(app.exec_())
44:
45: if __name__ == '__main__':
46:     main()

```

## dialogchoosecolor.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # dialogchoosecolor.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10:
11: def main():
12:
13:     app = PyQt5.QtWidgets.QApplication(sys.argv)
14:
15:     button = PyQt5.QtWidgets.QPushButton('Show Dialog')
16:
17:     textedit = PyQt5.QtWidgets.QTextEdit()
18:
19:     layout = PyQt5.QtWidgets.QGridLayout()
20:     layout.addWidget(button, 0, 0)
21:     layout.addWidget(textedit, 1, 0)
22:
23:     frame = PyQt5.QtWidgets.QFrame()
24:     frame.setLayout(layout)
25:
26:     window = PyQt5.QtWidgets.QMainWindow()
27:     window.setWindowTitle('Choose a color')
28:     window.setCentralWidget(frame)
29:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
30:     window.resize(screen_size.width()//2, screen_size.height()//2)
31:
32:     def button_slot():
33:         color = PyQt5.QtWidgets.QColorDialog.getColor()
34:         if color.isValid():
35:             textedit.append('Red: %s Green: %s Blue: %s' %
36:                             (color.red(), color.green(), color.blue()))
37:         else:
38:             textedit.append('(No color chosen)')
39:
40:     button.clicked.connect(button_slot)
41:
42:     window.show()
43:     sys.exit(app.exec_())
44:
45: if __name__ == '__main__':
46:     main()

```

## colordisplayer.py (Page 1 of 4)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # colordisplayer.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10: import PyQt5.QtCore
11: import PyQt5.QtGui
12:
13: #-----
14:
15: MAX_INTENSITY = 255
16:
17: #-----
18:
19: R = 0
20: G = 1
21: B = 2
22:
23: #-----
24:
25: def create_labels():
26:
27:     red_label = PyQt5.QtWidgets.QLabel('Red:')
28:     red_label.setAlignment(
29:         PyQt5.QtCore.Qt.AlignRight | PyQt5.QtCore.Qt.AlignVCenter)
30:     red_label.setAutoFillBackground(True)
31:     palette = red_label.palette()
32:     palette.setColor(
33:         red_label.backgroundRole(), PyQt5.QtCore.Qt.red)
34:     red_label.setPalette(palette)
35:
36:     green_label = PyQt5.QtWidgets.QLabel('Green:')
37:     green_label.setAlignment(
38:         PyQt5.QtCore.Qt.AlignRight | PyQt5.QtCore.Qt.AlignVCenter)
39:     green_label.setAutoFillBackground(True)
40:     palette = green_label.palette()
41:     palette.setColor(
42:         green_label.backgroundRole(), PyQt5.QtCore.Qt.green)
43:     green_label.setPalette(palette)
44:
45:     blue_label = PyQt5.QtWidgets.QLabel('Blue:')
46:     blue_label.setAlignment(
47:         PyQt5.QtCore.Qt.AlignRight | PyQt5.QtCore.Qt.AlignVCenter)
48:     blue_label.setAutoFillBackground(True)
49:     palette = blue_label.palette()
50:     palette.setColor(
51:         blue_label.backgroundRole(), PyQt5.QtCore.Qt.blue)
52:     blue_label.setPalette(palette)
53:
54:     return (red_label, green_label, blue_label)
55:
56: #-----
57:
58: def create_sliders():
59:
60:     red_slider = PyQt5.QtWidgets.QSlider(PyQt5.QtCore.Qt.Horizontal)
61:     red_slider.setMinimum(0)
62:     red_slider.setMaximum(MAX_INTENSITY)
63:
64:     green_slider = PyQt5.QtWidgets.QSlider(PyQt5.QtCore.Qt.Horizontal)
65:     green_slider.setMinimum(0)

```

## colordisplayer.py (Page 2 of 4)

```

66:     green_slider.setMaximum(MAX_INTENSITY)
67:
68:     blue_slider = PyQt5.QtWidgets.QSlider(PyQt5.QtCore.Qt.Horizontal)
69:     blue_slider.setMinimum(0)
70:     blue_slider.setMaximum(MAX_INTENSITY)
71:
72:     return (red_slider, green_slider, blue_slider)
73:
74: #-----
75:
76: def create_lineedit():
77:
78:     red_lineedit = PyQt5.QtWidgets.QLineEdit('0')
79:     green_lineedit = PyQt5.QtWidgets.QLineEdit('0')
80:     blue_lineedit = PyQt5.QtWidgets.QLineEdit('0')
81:
82:     return (red_lineedit, green_lineedit, blue_lineedit)
83:
84: #-----
85:
86: def create_control_frame(labels, sliders, lineedit):
87:
88:     control_frame_layout = PyQt5.QtWidgets.QGridLayout()
89:     control_frame_layout.setSpacing(0)
90:     control_frame_layout.setContentsMargins(0, 0, 0, 0)
91:     control_frame_layout.setRowStretch(0, 0)
92:     control_frame_layout.setRowStretch(1, 0)
93:     control_frame_layout.setRowStretch(2, 0)
94:     control_frame_layout.setColumnStretch(0, 0)
95:     control_frame_layout.setColumnStretch(1, 1)
96:     control_frame_layout.setColumnStretch(2, 0)
97:     control_frame_layout.addWidget(labels[R], 0, 0)
98:     control_frame_layout.addWidget(labels[G], 1, 0)
99:     control_frame_layout.addWidget(labels[B], 2, 0)
100:    control_frame_layout.addWidget(sliders[R], 0, 1)
101:    control_frame_layout.addWidget(sliders[G], 1, 1)
102:    control_frame_layout.addWidget(sliders[B], 2, 1)
103:    control_frame_layout.addWidget(lineedit[R], 0, 2)
104:    control_frame_layout.addWidget(lineedit[G], 1, 2)
105:    control_frame_layout.addWidget(lineedit[B], 2, 2)
106:    control_frame = PyQt5.QtWidgets.QFrame()
107:    control_frame.setLayout(control_frame_layout)
108:    return control_frame
109:
110: #-----
111:
112: def create_color_frame():
113:
114:     color_frame = PyQt5.QtWidgets.QFrame()
115:     color_frame.setAutoFillBackground(True)
116:     palette = color_frame.palette()
117:     palette.setColor(
118:         color_frame.backgroundRole(), PyQt5.QtCore.Qt.black)
119:     color_frame.setPalette(palette)
120:     return color_frame
121:
122: #-----
123:
124: def create_central_frame(color_frame, control_frame):
125:
126:     central_frame_layout = PyQt5.QtWidgets.QGridLayout()
127:     central_frame_layout.setSpacing(0)
128:     central_frame_layout.setContentsMargins(0, 0, 0, 0)
129:     central_frame_layout.setRowStretch(0, 1)
130:     central_frame_layout.setRowStretch(1, 0)

```

## colordisplayer.py (Page 3 of 4)

```

131:     central_frame_layout.setColumnStretch(0, 1)
132:     central_frame_layout.addWidget(color_frame, 0, 0)
133:     central_frame_layout.addWidget(control_frame, 1, 0)
134:     central_frame = PyQt5.QtWidgets.QFrame()
135:     central_frame.setLayout(central_frame_layout)
136:     return central_frame
137:
138: #-----
139:
140: def create_window(central_frame):
141:
142:     window = PyQt5.QtWidgets.QMainWindow()
143:     window.setWindowTitle('Color Displayer')
144:     window.setCentralWidget(central_frame)
145:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
146:     window.resize(screen_size.width()/2, screen_size.height()/2)
147:     return window
148:
149: #-----
150:
151: def slider_slot_helper(sliders, lineedit, color_frame):
152:
153:     red = sliders[R].value()
154:     green = sliders[G].value()
155:     blue = sliders[B].value()
156:     lineedit[R].setText(str(red))
157:     lineedit[G].setText(str(green))
158:     lineedit[B].setText(str(blue))
159:     palette = color_frame.palette()
160:     palette.setColor(
161:         color_frame.backgroundRole(),
162:         PyQt5.QtGui.QColor(red, green, blue))
163:     color_frame.setPalette(palette)
164:
165: #-----
166:
167: def lineedit_slot_helper(lineedit, sliders, color_frame):
168:
169:     try:
170:         red = int(lineedit[R].text())
171:         green = int(lineedit[G].text())
172:         blue = int(lineedit[B].text())
173:         if (red < 0) or (red > MAX_INTENSITY):
174:             raise Exception()
175:         if (green < 0) or (green > MAX_INTENSITY):
176:             raise Exception()
177:         if (blue < 0) or (blue > MAX_INTENSITY):
178:             raise Exception()
179:         sliders[R].setValue(red)
180:         sliders[G].setValue(green)
181:         sliders[B].setValue(blue)
182:         palette = color_frame.palette()
183:         palette.setColor(
184:             color_frame.backgroundRole(),
185:             PyQt5.QtGui.QColor(red, green, blue))
186:         color_frame.setPalette(palette)
187:     except Exception:
188:         # Use the Slider objects to restore the QLineEdit objects.
189:         lineedit[R].setText(str(sliders[R].value()))
190:         lineedit[G].setText(str(sliders[G].value()))
191:         lineedit[B].setText(str(sliders[B].value()))
192:
193: #-----
194:
195: def main():

```

## colordisplayer.py (Page 4 of 4)

```

196:
197:     app = PyQt5.QtWidgets.QApplication(sys.argv)
198:
199:     # Create and lay out widgets.
200:
201:     labels = create_labels()
202:     sliders = create_sliders()
203:     lineedit = create_lineedit()
204:     control_frame = create_control_frame(labels, sliders, lineedit)
205:     color_frame = create_color_frame()
206:     central_frame = create_central_frame(color_frame, control_frame)
207:     window = create_window(central_frame)
208:
209:     # Handle events for the QSlider objects.
210:
211:     def slider_slot():
212:         slider_slot_helper(sliders, lineedit, color_frame)
213:     for slider in sliders:
214:         slider.valueChanged.connect(slider_slot)
215:
216:     # Handle events for the QLineEdit objects.
217:
218:     def lineedit_slot():
219:         lineedit_slot_helper(lineedit, sliders, color_frame)
220:     for lineedit in lineedit:
221:         lineedit.returnPressed.connect(lineedit_slot)
222:
223:     window.show()
224:     sys.exit(app.exec_())
225:
226: if __name__ == '__main__':
227:     main()

```

## eventtestlambda.py (Page 1 of 1)

```
1: #!/usr/bin/env python
2:
3: #-----
4: # eventtestlambda.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import PyQt5.QtWidgets
10: import PyQt5.QtCore
11:
12: def main():
13:
14:     app = PyQt5.QtWidgets.QApplication(sys.argv)
15:
16:     red_button = PyQt5.QtWidgets.QPushButton('red')
17:     green_button = PyQt5.QtWidgets.QPushButton('green')
18:     blue_button = PyQt5.QtWidgets.QPushButton('blue')
19:
20:     def set_window_color(color):
21:         palette = window.palette()
22:         palette.setColor(window.backgroundRole(), color)
23:         window.setPalette(palette)
24:         window.repaint()
25:
26:     red_button.clicked.connect(
27:         lambda: set_window_color(PyQt5.QtCore.Qt.red))
28:     green_button.clicked.connect(
29:         lambda: set_window_color(PyQt5.QtCore.Qt.green))
30:     blue_button.clicked.connect(
31:         lambda: set_window_color(PyQt5.QtCore.Qt.blue))
32:
33:     layout = PyQt5.QtWidgets.QGridLayout()
34:     layout.addWidget(red_button, 0, 0)
35:     layout.addWidget(green_button, 0, 1)
36:     layout.addWidget(blue_button, 0, 2)
37:
38:     frame = PyQt5.QtWidgets.QFrame()
39:     frame.setLayout(layout)
40:
41:     window = PyQt5.QtWidgets.QMainWindow()
42:     window.setCentralWidget(frame)
43:     screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
44:     window.resize(screen_size.width()//2, screen_size.height()//2)
45:     window.setWindowTitle('Event Test 4')
46:     window.show()
47:
48:     sys.exit(app.exec_())
49:
50: if __name__ == '__main__':
51:     main()
```