File Name: Lab10.py
Contributors: Michael Avalos-Garcia , Jesus A. Bernal Lopez, & Paul Whipp
Date: 2/28/19
Description: In this lab we are digging further into the PyQt5 module by implementing a simple color picking GUI.

Task 1 & 2: For these tasks we focused our attention on making very clear, readable code. By splitting the code into short, targeted functions, we believe we have accomplished just that. Because these tasks are so closely linked, we produced them both at the same time.

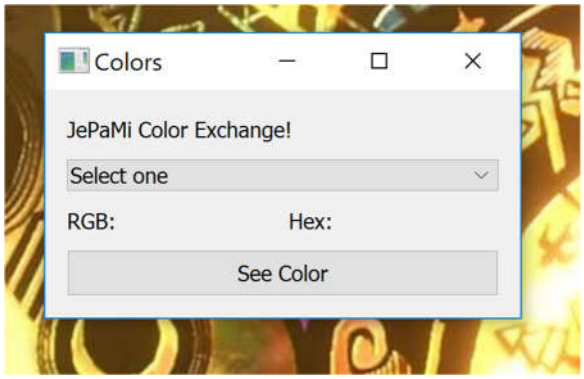
Code:

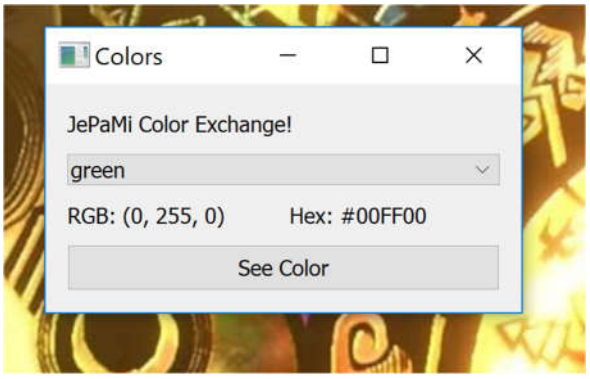
```
from PyQt5.QtWidgets import (QApplication, QVBoxLayout, QHBoxLayout, QWidget,
                                        QLabel, QComboBox, QPushButton)
from PyQt5.QtCore import pyqtSlot
import sys


class ShowColor(QWidget):

    def __init__(self):
        super().__init__()
        self.setWindowTitle("Color")

    def open_window(self, color):
        self.resize(250, 250)
        self.setWindowTitle(color)
        self.setStyleSheet(f"background-color: {color};")
        self.show()

    def no_color_error(self):
        self.resize(500, 1)
        self.setWindowTitle("Error: A color must be selected")
        self.show()


class HexConverter(QWidget):

    def __init__(self, color):
        super().__init__()
        self.colors = color

        self.setWindowTitle("Colors")
        color_names = list(self.colors.keys())
        color_names.insert(0, "Select one")

        # layouts
        self.vbox = QVBoxLayout()
        self.hbox = QHBoxLayout()

        # creating window items
```

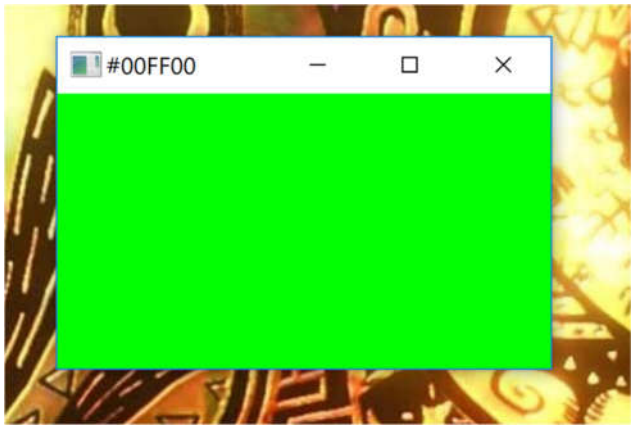
```
        self.title_label = QLabel("JePaMi Color Exchange!")
        self.color_dropdown = QComboBox()
        self.color_dropdown.addItems(color_names)
        self.see_color_button = QPushButton("See Color")
        self.see_color_button.clicked.connect(self.show_color)
        self.show_color = ShowColor()

        self.rgb_label = QLabel("RGB: ")
        self.hex_label = QLabel("Hex: ")

        # adding items to window
        self.vbox.addWidget(self.title_label)
        self.vbox.addWidget(self.color_dropdown)
        self.hbox.addWidget(self.rgb_label)
        self.hbox.addWidget(self.hex_label)
        self.vbox.addLayout(self.hbox)
        self.vbox.addWidget(self.see_color_button)

        self.setLayout(self.vbox)
        self.color_dropdown.currentIndexChanged.connect(self.update_labels)
        self.show()

    @pyqtSlot()
    def update_labels(self):
        try:
            self.rgb_label.setText(f"RGB:
{self.colors[self.color_dropdown.currentText()][0]}")
            self.hex_label.setText(f"Hex:
{self.colors[self.color_dropdown.currentText()][1]}")
        except KeyError:
            self.rgb_label.setText('')
            self.hex_label.setText('')

    @pyqtSlot()
    def show_color(self):
        try:

self.show_color.open_window(self.colors[self.color_dropdown.currentText()][1])
        except KeyError:
            self.show_color.no_color_error()

if __name__ == "__main__":

    color_dictionary = {
        "blue": [(0, 0, 255), "#0000FF"],
        "red": [(255, 0, 0), "#FF0000"],
        "green": [(0, 255, 0), "#00FF00"],
        "cyan": [(0, 255, 255), "#00FFFF"],
        "turquoise": [(64, 224, 208), "#40E0D0"],
        "teal": [(0, 128, 128), "#008080"],
        "pink": [(255, 192, 203), "#FFC0CB"],
        "lavender": [(230, 230, 250), "#E6E6FA"],
        "purple": [(85, 37, 130), "#552582"],
        "gold": [(253, 185, 39), "#FDB927"],
        "black": [(0, 0, 0), "#000000"],
```

```
    "white": [(255, 255, 255), "#FFFFFF"],
    "silver": [(192, 192, 192), "#C0C0C0"],
    "snow": [(255, 250, 250), "#FFFAFA"],
}

app = QApplication(sys.argv)
win = HexConverter(color_dictionary)
sys.exit(app.exec_())
```

Task 1 & 2 Results:







Task3: Please see the next page for our quiz notes.

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget
from PyQt5.QtWebEngineWidgets import QWebEngineView
from PyQt5.QtCore import QUrl

class Example(QWebEngineView):
    def __init__(self):
        super().__init__()
        self.load(QUrl('https://csumb.edu'))

app = QApplication(sys.argv)
ex = Example()
ex.show()
sys.exit(app.exec_())
```

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget
from PyQt5.QtGui import QColor

class Example(QWidget):
    def __init__(self):
        super().__init__()
        self.setAutoFillBackground(True)
        p = self.palette()
        p.setColor(self.backgroundRole(), QColor(227, 66, 52))
        self.setPalette(p)

app = QApplication(sys.argv)
ex = Example()
ex.show()
sys.exit(app.exec_())
```

```
@pyqtSlot()
def show_color(self):
    try:
        self.show_color.open_window(self.colors[self.color_dropdown.currentText()][1])
    except KeyError:
        self.show_color.no_color_error()
```

```
# adding items to window
self.vbox.addWidget(self.title_label)
self.vbox.addWidget(self.color_dropdown)
self.hbox.addWidget(self.rgb_label)
self.hbox.addWidget(self.hex_label)
self.vbox.addLayout(self.hbox)
self.vbox.addWidget(self.see_color_button)

self.setLayout(self.vbox)
self.color_dropdown.currentIndexChanged.connect(self.update_labels)
self.show()
```

```
class MyWindow(QWidget):
    def __init__(self):
        super().__init__()

        b1 = QPushButton('button 1')
        b2 = QPushButton('button 2')

        hbox = QHBoxLayout()
        hbox.addWidget(b1)
        hbox.addWidget(b2)

        b3 = QPushButton('button 3')
        b4 = QPushButton('button 4')

        vbox = QVBoxLayout()
        vbox.addWidget(b3)
        vbox.addWidget(b4)

        mbox = QHBoxLayout()
        mbox.addLayout(hbox)
        mbox.addLayout(vbox)
        self.setLayout(mbox)
```

```
# creating window items
self.title_label = QLabel("JePaMi Color Exchange!")
self.color_dropdown = QComboBox()
self.color_dropdown.addItems(color_names)
self.see_color_button = QPushButton("See Color")
self.see_color_button.clicked.connect(self.show_color)
self.show_color = ShowColor()
```