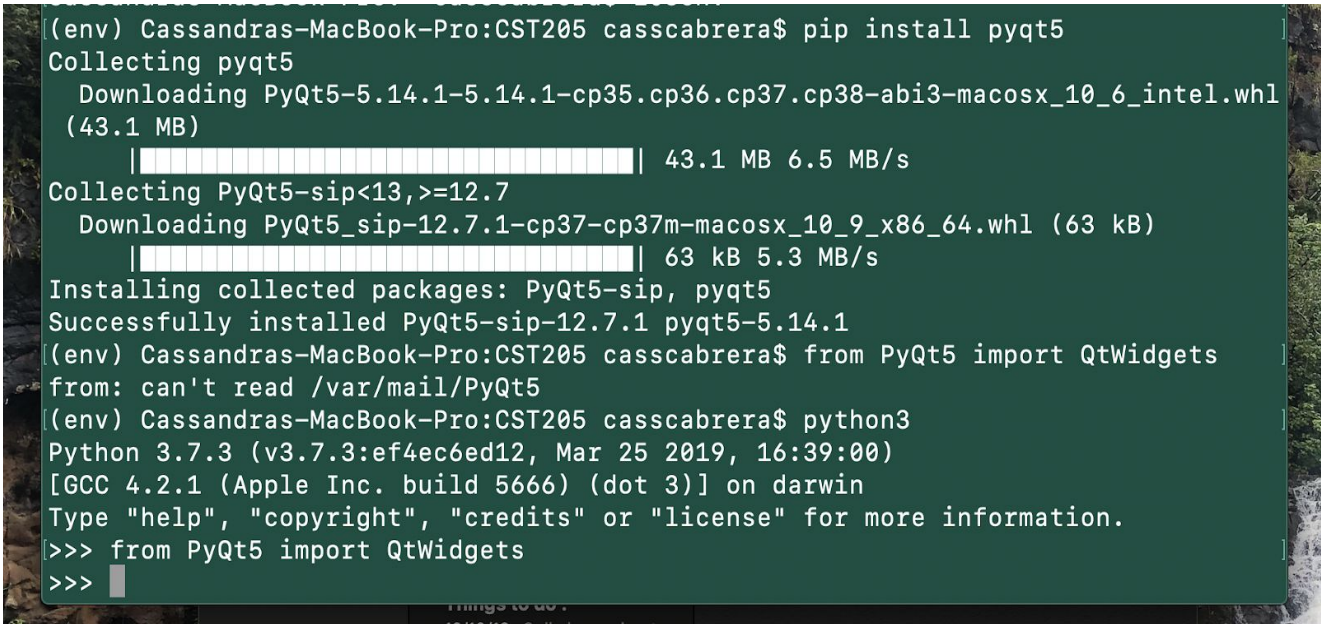
LAB 9: Cassandra Cabrera and Mike Menendez **Due:** Feb.26, 2020

Summary: For this lab, understanding exactly what a GUI is was kind of weird to wrap my head around. To better understand it, I watched some explanatory videos on YouTube and looked at different examples of how PyQt5 works and the different use cases for it. For Task 2, I used PyQt5 to first create my layout, then to create labels (which were our names), and then used the math settings (geometry) like we did in class. For task 3, this one was kind of confusing. Originally I was trying to make a program that would have a class and define itself, but for some reason I was not able to get that version to work. By doing it the way shown below, it does work, but I think it could be improved for reusability. For task 4, the on click was confusing, because at first I thought that the on click was a built in function and I didn't realize how to make it work with multiple buttons. After reading more documentation, I understood how it worked and got my button clicks to work for button 1 and 2.

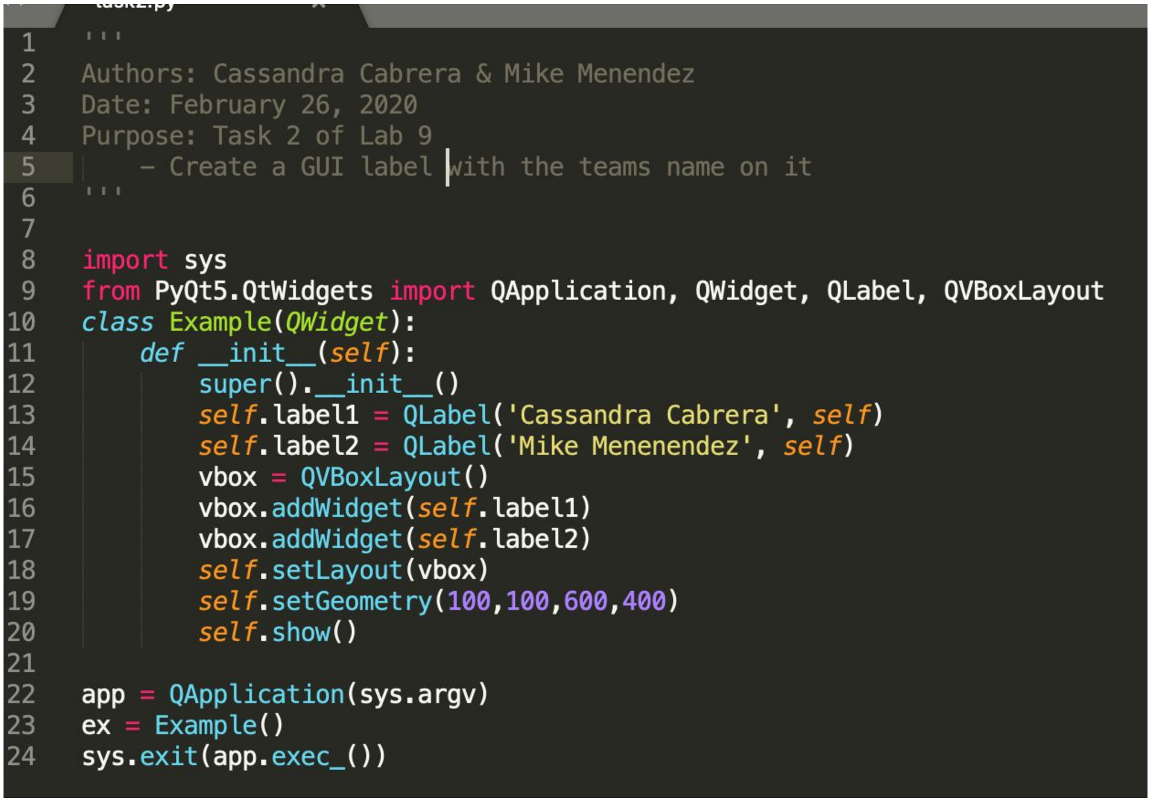
Code:

Task1



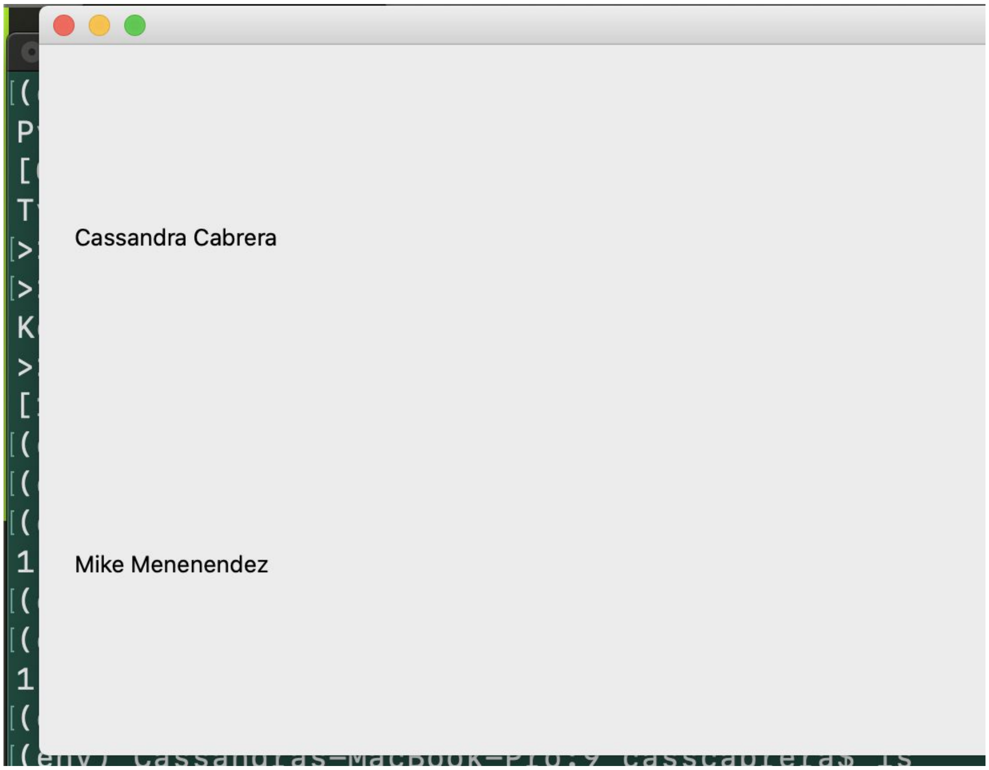
```
(env) Cassandras-MacBook-Pro:CST205 casscabrera$ pip install pyqt5
Collecting pyqt5
  Downloading PyQt5-5.14.1-5.14.1-cp35.cp36.cp37.cp38-abi3-macosx_10_6_intel.whl
 (43.1 MB)
     |████████████████████████████████| 43.1 MB 6.5 MB/s
Collecting PyQt5-sip<13,>=12.7
  Downloading PyQt5_sip-12.7.1-cp37-cp37m-macosx_10_9_x86_64.whl (63 kB)
     |████████████████████████████████| 63 kB 5.3 MB/s
Installing collected packages: PyQt5-sip, pyqt5
Successfully installed PyQt5-sip-12.7.1 pyqt5-5.14.1
(env) Cassandras-MacBook-Pro:CST205 casscabrera$ from PyQt5 import QtWidgets
from: can't read /var/mail/PyQt5
(env) Cassandras-MacBook-Pro:CST205 casscabrera$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:39:00)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from PyQt5 import QtWidgets
>>> 
```

Task2



```python
'''
Authors: Cassandra Cabrera & Mike Menendez
Date: February 26, 2020
Purpose: Task 2 of Lab 9
    - Create a GUI label with the teams name on it
'''

import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QVBoxLayout
class Example(QWidget):
    def __init__(self):
        super().__init__()
        self.label1 = QLabel('Cassandra Cabrera', self)
        self.label2 = QLabel('Mike Menenendez', self)
        vbox = QVBoxLayout()
        vbox.addWidget(self.label1)
        vbox.addWidget(self.label2)
        self.setLayout(vbox)
        self.setGeometry(100,100,600,400)
        self.show()

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

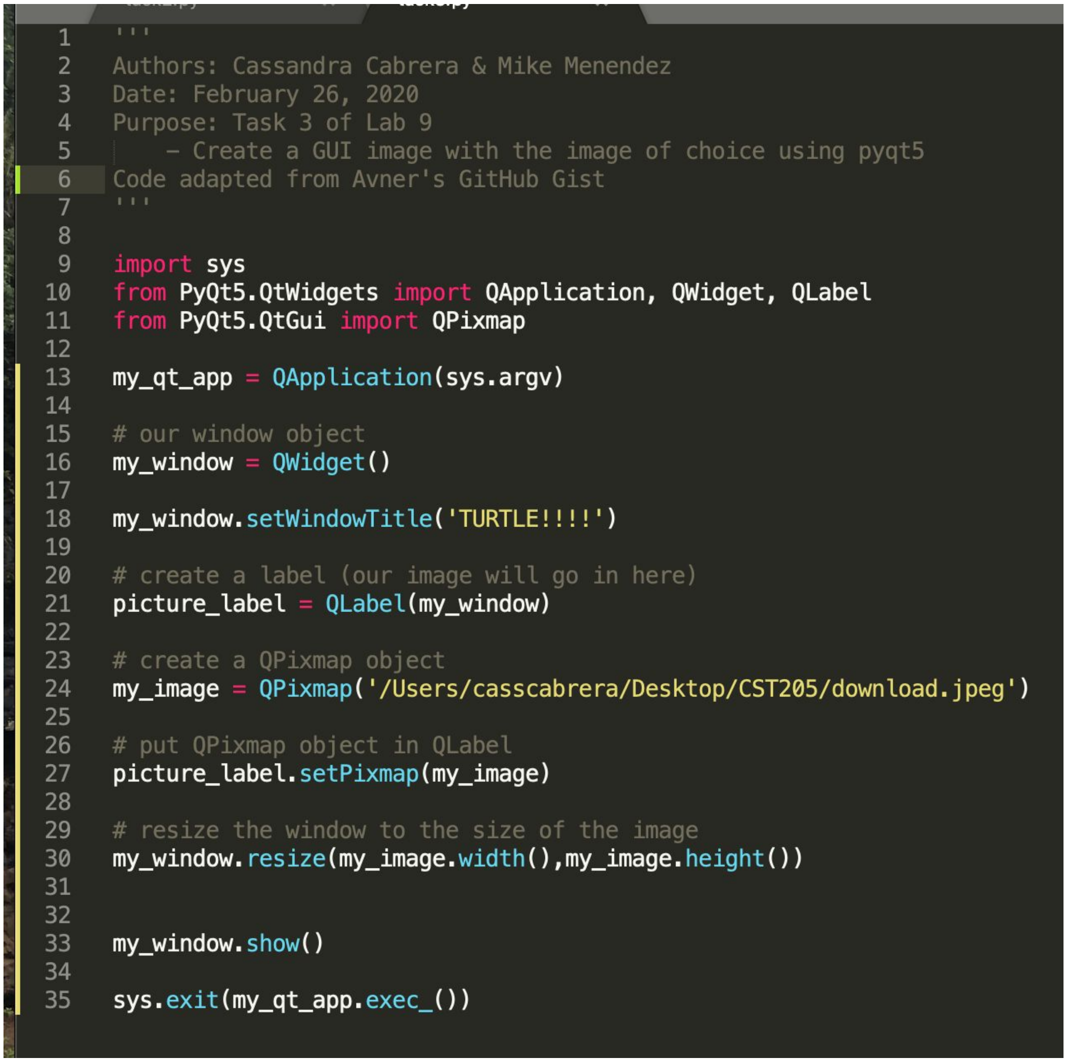
Results from Task 2



Cassandra Cabrera

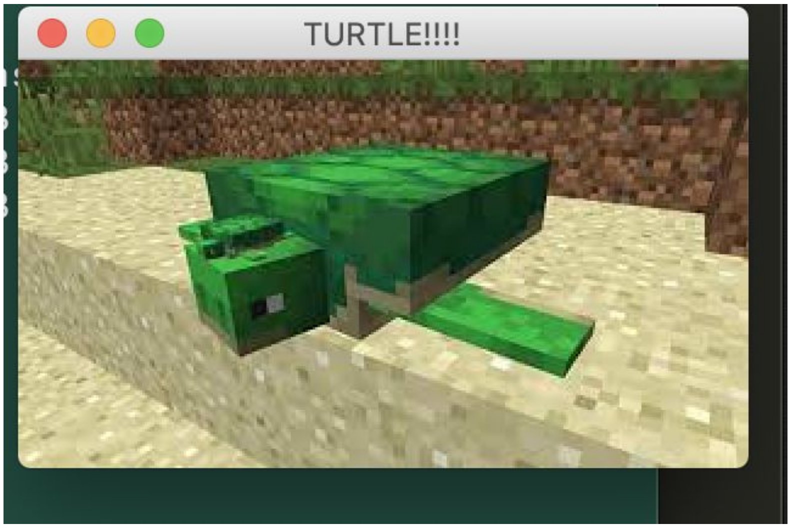
Mike Menenendez

Task 3

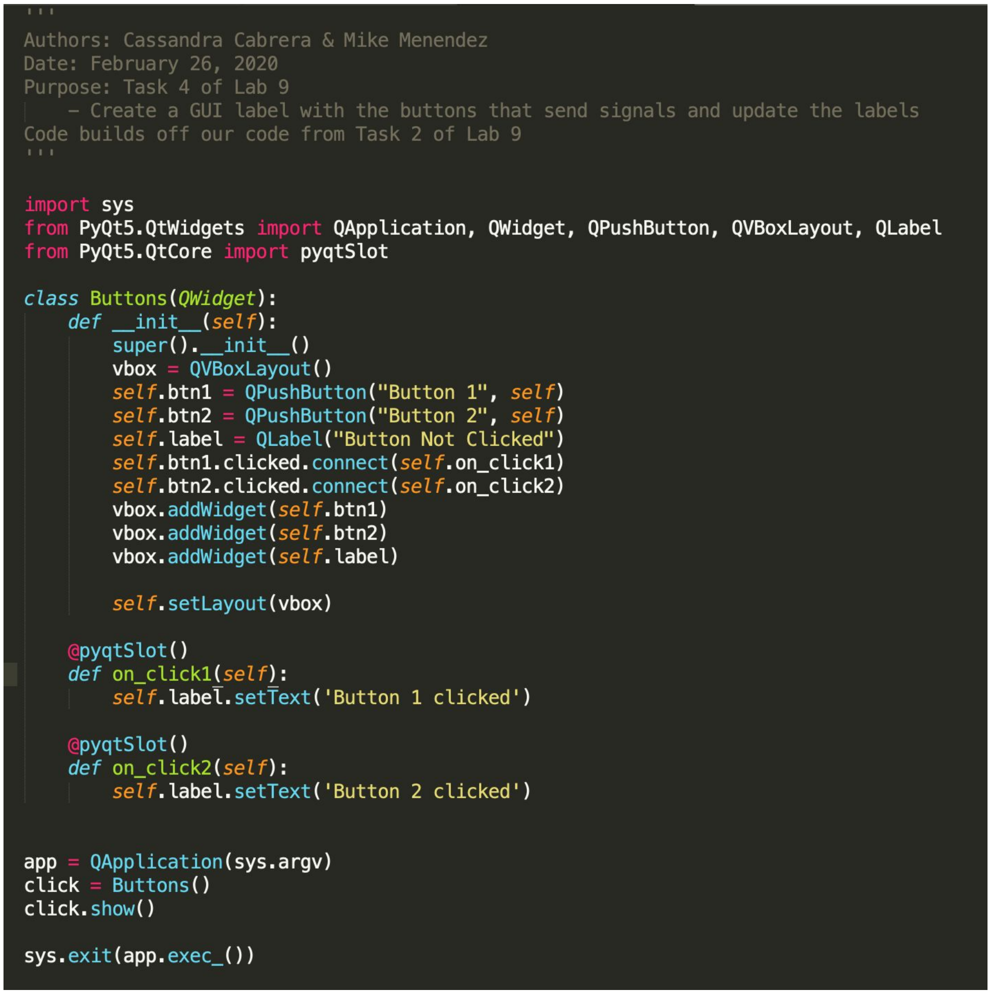


```
'''
Authors: Cassandra Cabrera & Mike Menendez
Date: February 26, 2020
Purpose: Task 3 of Lab 9
    - Create a GUI image with the image of choice using pyqt5
Code adapted from Avner's GitHub Gist
'''

import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel
from PyQt5.QtGui import QPixmap

my_qt_app = QApplication(sys.argv)

# our window object
my_window = QWidget()

my_window.setWindowTitle('TURTLE!!!!')

# create a label (our image will go in here)
picture_label = QLabel(my_window)

# create a QPixmap object
my_image = QPixmap('/Users/casscabrera/Desktop/CST205/download.jpeg')

# put QPixmap object in QLabel
picture_label.setPixmap(my_image)

# resize the window to the size of the image
my_window.resize(my_image.width(),my_image.height())


my_window.show()

sys.exit(my_qt_app.exec_())
```

Results from Task 3

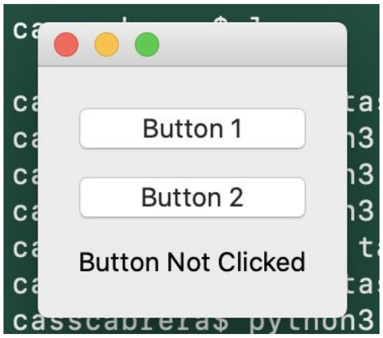


Task 4

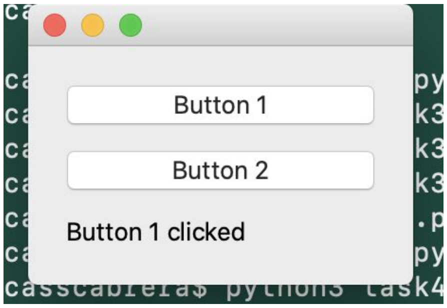


```
'''
Authors: Cassandra Cabrera & Mike Menendez
Date: February 26, 2020
Purpose: Task 4 of Lab 9
    - Create a GUI label with the buttons that send signals and update the labels
Code builds off our code from Task 2 of Lab 9
'''

import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QVBoxLayout, QLabel
from PyQt5.QtCore import pyqtSlot

class Buttons(QWidget):
    def __init__(self):
        super().__init__()
        vbox = QVBoxLayout()
        self.btn1 = QPushButton("Button 1", self)
        self.btn2 = QPushButton("Button 2", self)
        self.label = QLabel("Button Not Clicked")
        self.btn1.clicked.connect(self.on_click1)
        self.btn2.clicked.connect(self.on_click2)
        vbox.addWidget(self.btn1)
        vbox.addWidget(self.btn2)
        vbox.addWidget(self.label)

        self.setLayout(vbox)

    @pyqtSlot()
    def on_click1(self):
        self.label.setText('Button 1 clicked')

    @pyqtSlot()
    def on_click2(self):
        self.label.setText('Button 2 clicked')


app = QApplication(sys.argv)
click = Buttons()
click.show()

sys.exit(app.exec_())
```

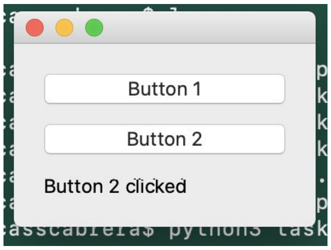
Results from Task4



Button Not Clicked



Button 1 clicked



Button 2 clicked