

This program was made with PYQT5. It loads several countries, their population, their total area, their population density, and percentage of the world population. It will also display a flag of the country being viewed.


Countries of the World

File

Choose a country from the list:

- Afghanistan
- Albania
- Algeria
- Andorra
- Angola
- Anguilla
- Antigua and Barbuda
- Argentina
- Armenia
- Aruba
- Australia
- Austria
- Azerbaijan
- Bahamas
- Bahrain
- Bangladesh
- Barbados
- Belarus
- Belgium
- Belize
- Benin
- Bermuda
- Bhutan

Afghanistan



Population:

Total Area in **252,072.0**

Population Density

Per Square Mile

Per Square KM

Percentage of World Population: **0.0516%**

Once loaded you have the option to change the data between square miles and square kilometers.


Countries of the World

File

Choose a country from the list:

- Bermuda
- Bhutan
- Bolivia
- Bosnia and Herzegovina
- Botswana
- Brazil
- Brunei
- Bulgaria
- Burkina Faso
- Burundi
- Cambodia
- Cameroon
- Canada**
- Cape Verde
- Cayman Islands
- Central African Republic
- Chad
- Chile
- China
- Christmas Island
- Colombia
- Comoros
- Cook Islands

Canada



Population:

Total Area in **3,855,081.0**

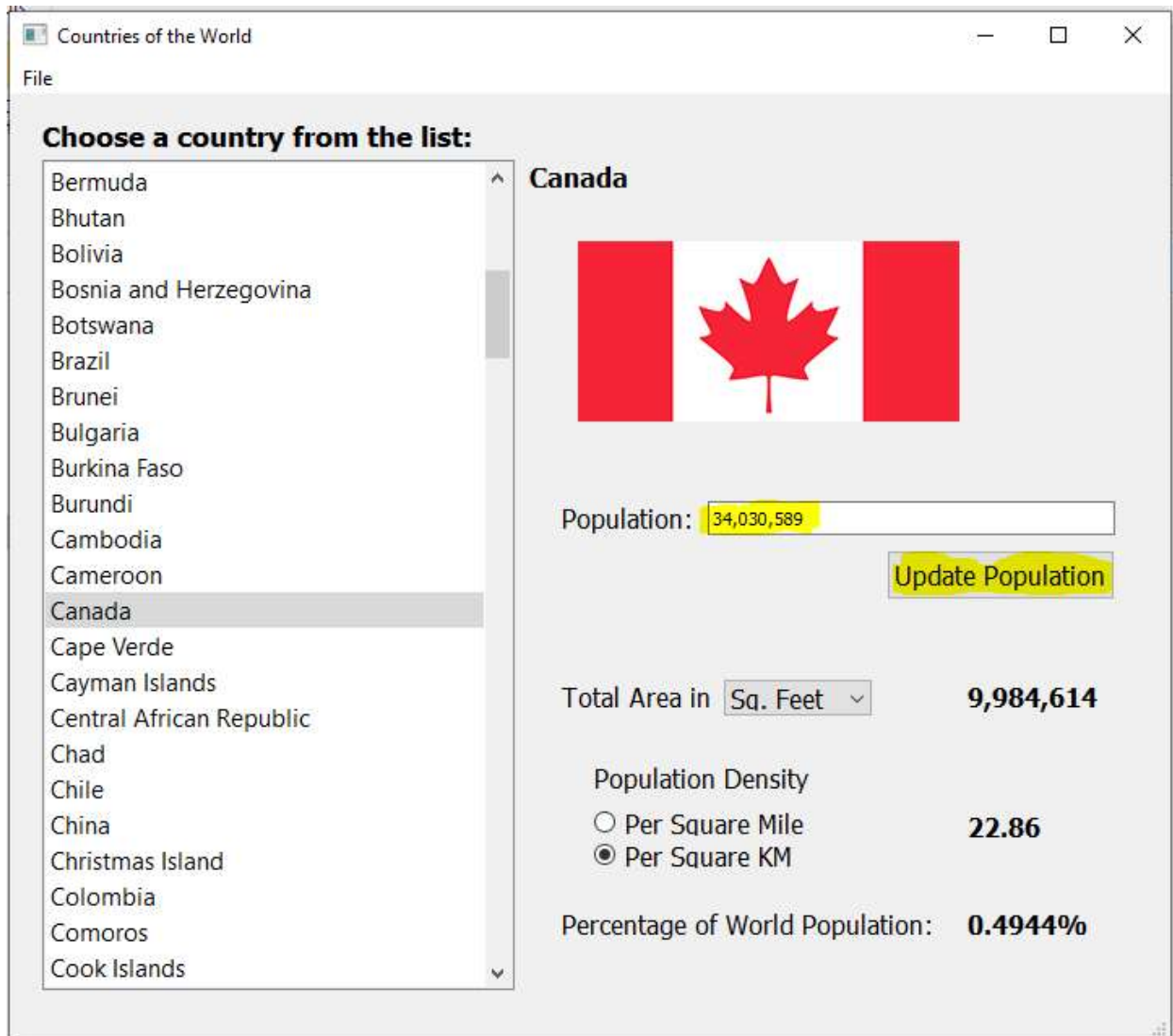
Population Density

Per Square Mile

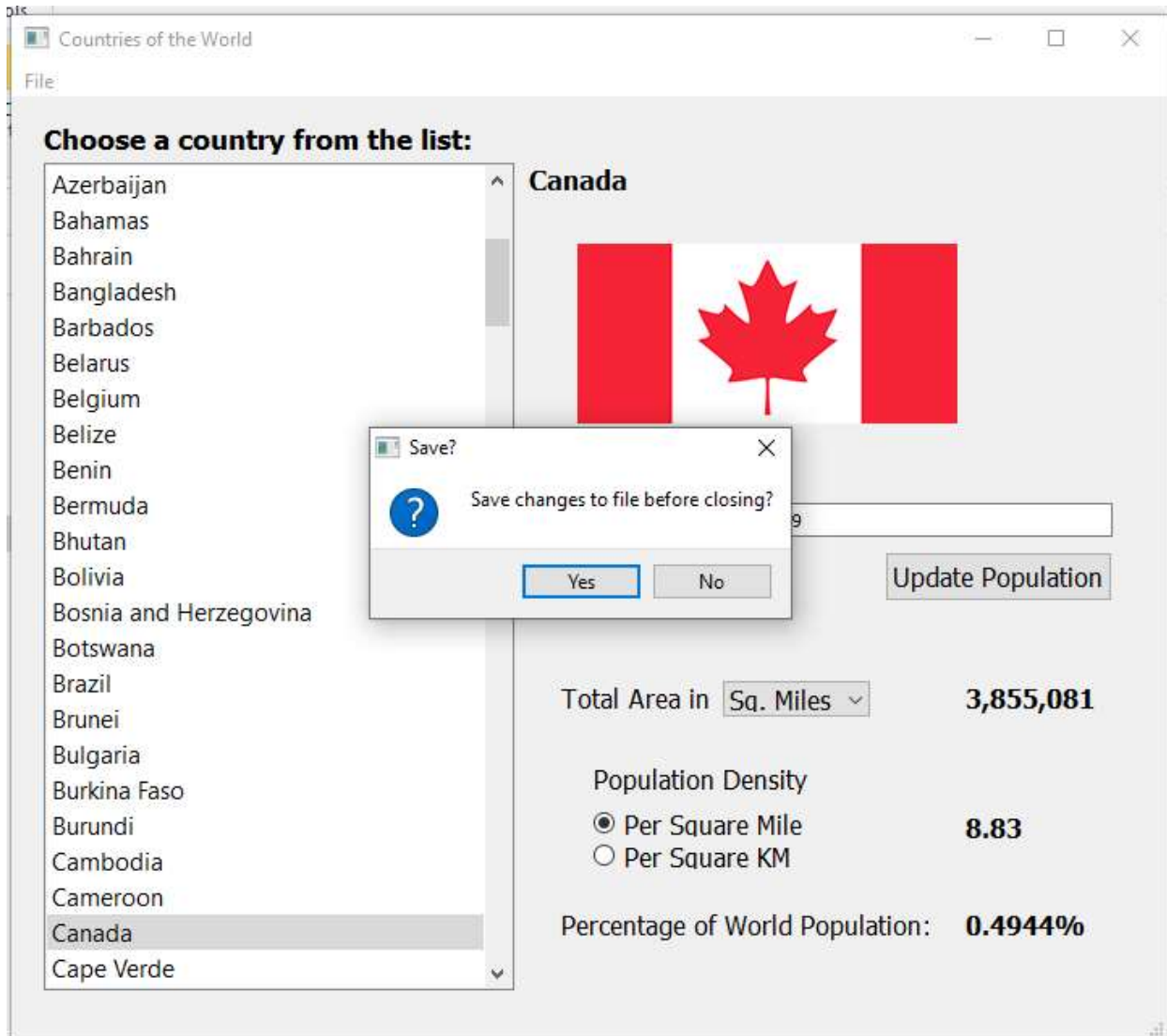
Per Square KM

Percentage of World Population: **0.4944%**

This screen is showing the data loading with a second country.



This program also gives you the option to update the program data. It will take the information you enter and add commas to the number as required. You can click Update Population to save the data to memory.



When you are ready, you can commit the changes to the program permanently, either by going to File > Save or by exiting the program. You can also choose to cancel.

PYW Code:

This program was from the very first programming class I ever took, there for some of this code was provided in the form of a template by my instructor which I then had to modify to make work. The sections my instructor added are commented in the code.

```

1  """
2  1. Create UI
3  2. Compile UI as MainWindow, import it.
4  3. Read data from countries.txt, save it as countries.csv
5  4. Create a 2dlist out of the CSV values
6  5. Column 0 is country name, column 1 is country population and column 2 is country
   square miles.
7  6. Create a menu that reads the CSV file the values from Column 0 in the different rows
8  7. Store data in memory.
9  8. Create labels/text boxes to display the country name, population and link it to the
   proper index.
10 9. Create a label for Flags and use the ID of the country name from the 2Dlist as a
   variable to get the
11    name of the country. Create a variable and use it when calling the file path.
   Replacing spaces with underscores
12 10. Calculate conversion from square miles to square KM and display on screen when
   selected via radio button.
13 11. Calculate population percentage by taking the sum of all column 1 values and doing
   (current pop X 100) / sum of col 1
14 12. Enable save option in menu to save from menu to countries.txt
15 13. Add confirmation messages and save/exit confirmation.
16 Student Name:   Steven Crosby
17 Program Title:  Countries of the World
18 Description:    Final Project: Phase 3
19 """
20
21 import sys
22
23 from PyQt5.QtWidgets import QApplication, QMainWindow, QMessageBox
24 from PyQt5.QtGui import QPixmap
25
26 #ADD IMPORT STATEMENT FOR YOUR GENERATED UI.PY FILE HERE
27 import Ui_FinalProject
28 #      ^^^^^^^^^^^ Change this!
29
30 #CHANGE THE SECOND PARAMETER (Ui ChangeMe) TO MATCH YOUR GENERATED UI.PY FILE
31 class MyForm(QMainWindow, Ui_FinalProject.Ui_MainWindow):
32     #      ^^^^^^^^^^^ Change this!
33
34     def LoadCountriesFromFile(newCSVFile):
35         try:load the file, convert to CSV and create a 2DLIST.
36             fileName = "Files/countries.txt"
37             accessMode = "r"
38
39             with open(fileName, accessMode) as myTXTFile:
40                 myTXT=myTXTFile.read()
41
42             fileName = "Files/countries.csv"
43             accessMode = "w"
44
45             with open(fileName, accessMode) as newCSVFile:
46                 newCSVFile.write(myTXT)
47         except FileNotFoundError:
48             fileName = "Files/countries.txt"
49             accessMode = "w"
50
51             with open(fileName, accessMode) as myTXTFile:
52                 data = "Error Reading from File Path,0,0"
53                 myTXTFile.write(data)
54                 myTXT = data
55
56             fileName = "Files/countries.csv"
57             accessMode = "w"
58
59             with open(fileName, accessMode) as newCSVFile:
60                 newCSVFile.write(myTXT)
61                 return myTXT
62
63     newCSVFile = ""
64     myTXT = LoadCountriesFromFile(newCSVFile)

```

```

62 people = []
63 my2dList = []
64 unsaved changes = False
65 # DO NOT MODIFY THIS CODE
66 def __init__(self, parent=None): #Runs when form loads, before user sees anything
67     super(MyForm, self).__init__(parent)
68     self.setupUi(self)
69 # END DO NOT MODIFY
70
71
72     # ADD SLOTS HERE, indented to this level (ie. inside def __init__ )
73     self.frameCountries.setVisible(False)
74     self.listWidgetPeople.currentRowChanged.connect(self.DisplayCountryData)
75     self.listWidgetPeople.currentRowChanged.connect(self.ShowImage)
76     self.actionLoadCountries.triggered.connect(self.LoadCountriesListBox)
77     self.actionSaveToFile.triggered.connect(self.saveNoExit)
78     self.actionExit.triggered.connect(self.exitApplication)
79     self.pushButtonUpdate.clicked.connect(self.updatePeople)
80     self.comboBoxMiFt.currentIndexChanged.connect(self.CalculateTotalWorldPopulation)
81     self.comboBoxMiFt.addItem("Sq. Miles")
82     self.comboBoxMiFt.addItem("Sq. Feet")
83     self.radioButtonMile.clicked.connect(self.radioButtonMileClicked)
84     self.radioButtonKM.clicked.connect(self.radioButtonKMClicked)
85
86 # ADD SLOT FUNCTIONS HERE
87 # These are the functions your slots will point to
88 # Indent to this level (ie. inside the class, at same level as def __init__)
89 def LoadCountriesListBox(self):
90     self.frameCountries.setVisible(True)
91     if len(self.my2dList) == 0:
92         self.LoadCountriesFromFile()
93
94     self.listWidgetPeople.clear()
95     for row in self.my2dList:
96         self.listWidgetPeople.addItem(row[0])
97         #Add country names in 2Dlist to list widget
98     self.sumList = []
99     self.sumofSumList = []
100     global compileList
101     compileList = self.my2dList
102     selectedCountryIndex = self.listWidgetPeople.currentRow()
103     updatePop = (str(compileList[selectedCountryIndex][1]))
104     try:#Check for errors in the list/if file is non existant
105         updatePop = int(updatePop)
106     except ValueError:
107         QMessageBox.information(self, "Error", "Error reading from the file path.
108             Closing program.", QMessageBox.Ok)
109         exit()
110     except:
111         QMessageBox.information(self, "Error", "Error reading from the file path.
112             Closing program.", QMessageBox.Ok)
113         exit()
114
115     self.lineEditPop.setText("{:,.0F}".format(updatePop))
116     total = 0
117     indexLoop = 0
118     for row in compileList:
119         self.sumList.append(compileList[indexLoop][1])
120         indexLoop = indexLoop + 1
121     # appending the values of column 1 to its own list to sum
122     for counter in range(0, len(self.sumList)):
123         total = total + int(str(self.sumList[counter]))
124         self.popTotal=str(total)
125
126 def CalculateTotalWorldPopulation(self):
127     fileName = "Files/countries.csv"
128     accessMode = "r"

```

```

127
128     with open(fileName, accessMode) as myCSVFile:
129         import csv
130         fileData = csv.reader(myCSVFile)
131
132         self.my2dList = []
133         for row in fileData:
134             self.my2dList.append(row)
135         comboIndex = self.comboBoxMiFt.currentIndex()
136         try:
137             if comboIndex == 1:
138                 selectedCountryIndex = self.listWidgetPeople.currentRow()
139                 convertKM1 = float(self.my2dList[selectedCountryIndex][2])
140                 convertKM2 = float(2.58998611)
141                 convertTotal = (convertKM1 * convertKM2)
142                 self.labelEditSqm.setText("{:,.0F}".format(convertTotal))
143             elif comboIndex == 0:
144                 selectedCountryIndex = self.listWidgetPeople.currentRow()
145                 convertTotal = float(self.my2dList[selectedCountryIndex][2])
146                 self.labelEditSqm.setText("{:,.0F}".format(convertTotal))
147         except ValueError:
148             convertTotal = 1
149             self.labelEditSqm.setText("{:,.0F}".format(convertTotal))
150             #Square Mile to KM conversion + error checking
151
152
153
154     def DisplayCountryData(self, selectedCountryIndex):
155         OneHundred = 100
156         convertPop = compileList[selectedCountryIndex][1]
157         try:
158             convertPop = float(convertPop)
159             self.lineEditPop.clear()
160             self.lineEditPop.setText("{:,.0F}".format(convertPop))
161             self.lineEditPop.update()
162         except ValueError:
163             value = self.listWidgetPeople.currentItem()
164             if value == "Error Reading from File Path":
165                 QMessageBox.information(self, "Invalid", "Data is invalid so not
166                 updated in memory", QMessageBox.Ok)
167                 exit()
168             else:
169                 QMessageBox.information(self, "Invalid", "Data is invalid so not
170                 updated in memory", QMessageBox.Ok)
171
172         try:
173             popLabel = (convertPop)/float(self.popTotal)*float(OneHundred)
174         except AttributeError:
175             exit()
176         except TypeError:
177             QMessageBox.information(self, "Invalid", "Data is invalid so not updated in
178             memory", QMessageBox.Ok)
179         except ZeroDivisionError:
180             QMessageBox.information(self, "Error", "Error reading from the file path.
181             Closing program.", QMessageBox.Ok)
182             exit()
183         self.labelCountry.clear()
184         self.labelCountry.setText(str(compileList[selectedCountryIndex][0]))
185         self.labelCountry.update()
186         self.labelEditSqm.clear()
187         MiFeetUpdate = compileList[selectedCountryIndex][2]
188         try:
189             MiFeetUpdate = float(MiFeetUpdate)
190         except ValueError:
191             MiFeetUpdate = 0.0
192             QMessageBox.information(self, "Error", "Error reading from the file path.
193             Closing program.", QMessageBox.Ok)
194             exit()

```



```

189     self.labelEditSqM.setText("{:,.1f}".format(MiFeetUpdate))
190     self.labelEditSqM.update()
191     MiFeetUpdate = str(MiFeetUpdate)
192     MiFeetUpdate = MiFeetUpdate.replace(",","")
193     MiFeetUpdate = MiFeetUpdate.replace("_","")
194     self.labelEditPopulation.clear()
195     try:
196         self.labelEditPopulation.setText("{:.4f}%".format(popLabel))
197         self.labelEditPopulation.update()
198     except UnboundLocalError:
199         QMessageBox.information(self, "Invalid", "Data is invalid so not updated in
200         memory", QMessageBox.Ok)
201         #population percentage calculations and program \
202         # refresh after value update
203
204     def radioButtonMileClicked(self, enabled):
205         selectedCountryIndex = self.listWidgetPeople.currentRow()
206         squareMil = float(compileList[selectedCountryIndex][1])
207         squareMi2 = float(compileList[selectedCountryIndex][2])
208         squareMiTotal= (squareMil) / (squareMi2)
209         if enabled:
210             self.comboBoxMiFt.currentIndexChanged.connect(self.populateListWithPeople)
211             self.labelEditDensity.clear()
212             self.labelEditDensity.setText("{:.2f}".format(squareMiTotal))
213             self.listWidgetPeople.currentRowChanged.connect(self.radioButtonMileClicked)
214             self.listWidgetPeople.currentRowChanged.connect(self.radioButtonKMClicked)
215         #Radio button conversion
216     def radioButtonKMClicked(self, enabled):
217         selectedCountryIndex = self.listWidgetPeople.currentRow()
218         squareKM1 = float(compileList[selectedCountryIndex][1])
219         squareKM2 = float(2.58998811)
220         squareKM3 = float(compileList[selectedCountryIndex][2])
221         squareKMTotal= (squareKM1 * squareKM2) / (squareKM3)
222         if enabled:
223             self.labelEditDensity.clear()
224             self.labelEditDensity.setText("{:.2f}".format(squareKMTotal))
225             self.listWidgetPeople.currentRowChanged.connect(self.radioButtonMileClicked)
226             self.listWidgetPeople.currentRowChanged.connect(self.radioButtonKMClicked)
227
228     #ADD HELPER FUNCTIONS HERE
229     # These are the functions the slot functions will call, to
230     # contain the custom code that you'll write to make your program work.
231     # Indent to this level (ie. inside the class, at same level as def __init__)
232
233     def ShowImage(self):
234         fileName = "Files/countries.csv"
235         accessMode = "r"
236         with open(fileName, accessMode) as imageCSVFile:
237             import csv
238             imageData = csv.reader(imageCSVFile) #KIND OF gives me a 2d list of the
             file's contents
239
240             self.imageList = []
241             for row in imageData:
242                 self.imageList.append(row)
243             index = self.listWidgetPeople.currentRow()
244             printIndex=(self.imageList[index][0])
245             printIndex = printIndex.replace(" ", "_")
246             image=QPixmap("Files/Flags/"+str(printIndex)+".png")
247             self.labelPicture.clear()
248             self.labelPicture.setPixmap(image)
249             #get proper flag image loaded
250
251     def closeEvent(self, event):
252         self.unsaved_changes == True
253         quit_msg = "Save changes to file before closing?"

```

```

254     reply = QMessageBox.question(self, 'Save?',
255         quit_msg, QMessageBox.Yes, QMessageBox.No)
256
257     if reply == QMessageBox.Yes:
258         self.unsaved_changes = False
259         self.actionSavetoFile.setEnabled(False)
260         myFile=""
261         self.SaveCountriesToFile(myFile)
262         QMessageBox.information(self, 'Saved', 'Changes were saved to the file',
263             QMessageBox.Ok)
264     else:
265         exit()
266         #Closing (click the X)
267
268 def saveNoExit(self, event):
269     self.unsaved_changes = True
270     quit_msg = "Save changes to file?"
271     reply = QMessageBox.question(self, 'Save?',
272         quit_msg, QMessageBox.Yes, QMessageBox.No)
273     if reply == QMessageBox.Yes:
274         myFile=""
275         self.SaveCountriesToFile(myFile)
276         self.actionSavetoFile.setEnabled(False)
277     if reply == QMessageBox.No:
278         self.unsaved_changes = False
279         pass
280     #Save menu)
281
282 def exitApplication(self, event):
283     self.unsaved_changes = True
284     quit_msg = "Save changes to file before closing?"
285     reply = QMessageBox.question(self, 'Save?',
286         quit_msg, QMessageBox.Yes, QMessageBox.No)
287     if reply == QMessageBox.Yes:
288         myFile=""
289         self.SaveCountriesToFile(myFile)
290         exit()
291     if reply == QMessageBox.No:
292         self.unsaved_changes = False
293         exit()
294         pass
295     # File > Exit
296
297 def SaveCountriesToFile(self, myFile):
298     with open("Files/countries.txt", "w") as myFile:
299         for person in compileList:
300             finalPrint = person[1].replace(" ", "")
301             selected_index = self.listWidgetPeople.currentRow()
302             finalPrint=compileList[selected_index][1]
303             finalPrint = self.lineEditPop.text()
304             myFile.write(",".join(person) + "\n")
305             self.unsaved_changes = False
306             #Saving function
307
308 def updatePeople(self):
309     selected_index = self.listWidgetPeople.currentRow()
310     replaceUnderscore = self.lineEditPop.text()
311     replaceUnderscore=replaceUnderscore.replace(" ", "_")
312     replaceUnderscore=replaceUnderscore.replace("_", " ")
313     if replaceUnderscore.isalpha() != True:
314         self.lineEditPop.setText(replaceUnderscore)
315         (compileList[selected_index][1]) = self.lineEditPop.text()
316         replaceUnderscore=int(replaceUnderscore)
317         self.lineEditPop.setText("{:,.0F}".format(replaceUnderscore))
318         QMessageBox.information(self, 'Updated', "Data has been updated in memory,
319             but hasn't been updated in the file yet.", QMessageBox.Ok)
320         self.actionSavetoFile.setEnabled(True)

```

```

319         self.unsaved_changes == True
320         self.populateListWithPeople()
321         replaceUnderscore=str(replaceUnderscore)
322     if replaceUnderscore.isalpha() == True:
323         try:
324             self.unsaved_changes == False
325             replaceUnderscore=str(replaceUnderscore)
326             self.lineEditPop.setText("{:,.0F}".format(replaceUnderscore))
327         except ValueError:
328             QMessageBox.information(self, "Invalid", "Data is invalid so not
updated in memory", QMessageBox.Ok)
329             self.lineEditPop.setText(self.my2dList[selected_index][1])
330             (compileList[selected_index][1]) = self.lineEditPop.text()
331             replaceUnderscore = (compileList[selected_index][1])
332             replaceUnderscore=int(replaceUnderscore)
333             self.lineEditPop.setText("{:,.0F}".format(replaceUnderscore))
334             #Error checking + removing underscore from user inputs
335
336
337     def populateListWithPeople(self):
338         self.unsaved_changes == True
339         self.lineEditPop.clear()
340         selected_index = self.listWidgetPeople.currentRow()
341         strCheck = (compileList[selected_index][1])
342         replaceUnderscore = strCheck
343         replaceUnderscore=replaceUnderscore.replace(",","_")
344         replaceUnderscore=replaceUnderscore.replace(" ","_")
345         intStrCheck = int(replaceUnderscore)
346         self.lineEditPop.clear()
347         self.lineEditPop.setText("{:,.0F}".format(intStrCheck))
348         #repopulate list, check there are no underscores
349
350     # DO NOT MODIFY THIS CODE
351     if __name__ == "__main__":
352         app = QApplication(sys.argv)
353         the_form = MyForm()
354         the_form.show()
355         sys.exit(app.exec_())
356     # END DO NOT MODIFY

```

PY code, created by: PyQt5 UI code generator 5.15.1

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file
4  'c:\PROG1700 SourceCode\w0218814 Crosby8\GUI\FinalProject\FinalProject.ui'
5  #
6  # Created by: PyQt5 UI code generator 5.15.1
7  #
8  # WARNING: Any manual changes made to this file will be lost when pyuic5 is
9  # run again. Do not edit this file unless you know what you are doing.
10
11 from PyQt5 import QtCore, QtGui, QtWidgets
12
13
14 class Ui_MainWindow(object):
15     def setupUi(self, MainWindow):
16         MainWindow.setObjectName("MainWindow")
17         MainWindow.resize(715, 602)
18         self.centralwidget = QtWidgets.QWidget(MainWindow)
19         self.centralwidget.setObjectName("centralwidget")
20         self.listWidgetPeople = QtWidgets.QListWidget(self.centralwidget)
21         self.listWidgetPeople.setGeometry(QtCore.QRect(20, 40, 291, 511))
22         font = QtGui.QFont()
23         font.setPointSize(11)
24         self.listWidgetPeople.setFont(font)
25         self.listWidgetPeople.setObjectName("listWidgetPeople")
26         self.labelPeople = QtWidgets.QLabel(self.centralwidget)
27         self.labelPeople.setGeometry(QtCore.QRect(20, 0, 291, 51))
28         font = QtGui.QFont()
29         font.setPointSize(13)
30         font.setBold(True)
31         font.setWeight(75)
32         self.labelPeople.setFont(font)
33         self.labelPeople.setObjectName("labelPeople")
34         self.frameCountries = QtWidgets.QFrame(self.centralwidget)
35         self.frameCountries.setGeometry(QtCore.QRect(320, -10, 391, 551))
36         self.frameCountries.setFrameShape(QtWidgets.QFrame.StyledPanel)
37         self.frameCountries.setFrameShadow(QtWidgets.QFrame.Raised)
38         self.frameCountries.setObjectName("frameCountries")
39         self.labelPop = QtWidgets.QLabel(self.frameCountries)
40         self.labelPop.setGeometry(QtCore.QRect(20, 250, 121, 41))
41         font = QtGui.QFont()
42         font.setPointSize(12)
43         self.labelPop.setFont(font)
44         self.labelPop.setObjectName("labelPop")
45         self.comboBoxMiFt = QtWidgets.QComboBox(self.frameCountries)
46         self.comboBoxMiFt.setGeometry(QtCore.QRect(120, 370, 91, 22))
47         font = QtGui.QFont()
48         font.setPointSize(12)
49         self.comboBoxMiFt.setFont(font)
50         self.comboBoxMiFt.setObjectName("comboBoxMiFt")
51         self.labelEditSqM = QtWidgets.QLabel(self.frameCountries)
52         self.labelEditSqM.setGeometry(QtCore.QRect(270, 360, 101, 41))
53         font = QtGui.QFont()
54         font.setPointSize(12)
55         font.setBold(True)
56         font.setWeight(75)
57         self.labelEditSqM.setFont(font)
58         self.labelEditSqM.setText("")
59         self.labelEditSqM.setObjectName("labelEditSqM")
60         self.labelDensity = QtWidgets.QLabel(self.frameCountries)
61         self.labelDensity.setGeometry(QtCore.QRect(40, 410, 141, 41))
62         font = QtGui.QFont()
63         font.setPointSize(12)
64         self.labelDensity.setFont(font)
65         self.labelDensity.setObjectName("labelDensity")
66         self.radioButtonKM = QtWidgets.QRadioButton(self.frameCountries)

```

```

67     self.radioButtonKM.setGeometry(QtCore.QRect(40, 470, 141, 16))
68     font = QtGui.QFont()
69     font.setPointSize(12)
70     self.radioButtonKM.setFont(font)
71     self.radioButtonKM.setObjectName("radioButtonKM")
72     self.labelEditDensity = QtWidgets.QLabel(self.frameCountries)
73     self.labelEditDensity.setGeometry(QtCore.QRect(270, 440, 91, 41))
74     font = QtGui.QFont()
75     font.setPointSize(12)
76     font.setBold(True)
77     font.setWeight(75)
78     self.labelEditDensity.setFont(font)
79     self.labelEditDensity.setText("")
80     self.labelEditDensity.setObjectName("labelEditDensity")
81     self.labelTotalArea = QtWidgets.QLabel(self.frameCountries)
82     self.labelTotalArea.setGeometry(QtCore.QRect(20, 360, 121, 41))
83     font = QtGui.QFont()
84     font.setPointSize(12)
85     self.labelTotalArea.setFont(font)
86     self.labelTotalArea.setObjectName("labelTotalArea")
87     self.radioButtonMile = QtWidgets.QRadioButton(self.frameCountries)
88     self.radioButtonMile.setGeometry(QtCore.QRect(40, 450, 141, 16))
89     font = QtGui.QFont()
90     font.setPointSize(12)
91     self.radioButtonMile.setFont(font)
92     self.radioButtonMile.setObjectName("radioButtonMile")
93     self.lineEditPop = QtWidgets.QLineEdit(self.frameCountries)
94     self.lineEditPop.setGeometry(QtCore.QRect(110, 260, 251, 21))
95     self.lineEditPop.setObjectName("lineEditPop")
96     self.labelPicture = QtWidgets.QLabel(self.frameCountries)
97     self.labelPicture.setGeometry(QtCore.QRect(30, 100, 241, 111))
98     self.labelPicture.setText("")
99     self.labelPicture.setObjectName("labelPicture")
100    self.labelEditPopulation = QtWidgets.QLabel(self.frameCountries)
101    self.labelEditPopulation.setGeometry(QtCore.QRect(270, 500, 91, 41))
102    font = QtGui.QFont()
103    font.setPointSize(12)
104    font.setBold(True)
105    font.setWeight(75)
106    self.labelEditPopulation.setFont(font)
107    self.labelEditPopulation.setText("")
108    self.labelEditPopulation.setObjectName("labelEditPopulation")
109    self.labelPercent = QtWidgets.QLabel(self.frameCountries)
110    self.labelPercent.setGeometry(QtCore.QRect(20, 500, 261, 41))
111    font = QtGui.QFont()
112    font.setPointSize(12)
113    self.labelPercent.setFont(font)
114    self.labelPercent.setObjectName("labelPercent")
115    self.labelCountry = QtWidgets.QLabel(self.frameCountries)
116    self.labelCountry.setGeometry(QtCore.QRect(0, 40, 441, 41))
117    font = QtGui.QFont()
118    font.setPointSize(12)
119    font.setBold(True)
120    font.setWeight(75)
121    self.labelCountry.setFont(font)
122    self.labelCountry.setText("")
123    self.labelCountry.setObjectName("labelCountry")
124    self.pushButtonUpdate = QtWidgets.QPushButton(self.frameCountries)
125    self.pushButtonUpdate.setGeometry(QtCore.QRect(220, 290, 141, 31))
126    font = QtGui.QFont()
127    font.setPointSize(12)
128    self.pushButtonUpdate.setFont(font)
129    self.pushButtonUpdate.setObjectName("pushButtonUpdate")
130    MainWindow.setCentralWidget(self.centralwidget)
131    self.menubar = QtWidgets.QMenuBar(MainWindow)
132    self.menubar.setGeometry(QtCore.QRect(0, 0, 715, 21))
133    self.menubar.setObjectName("menubar")

```

```

134     self.menuFile = QtWidgets.QMenu(self.menubar)
135     self.menuFile.setObjectName("menuFile")
136     MainWindow.setMenuBar(self.menubar)
137     self.statusbar = QtWidgets.QStatusBar(MainWindow)
138     self.statusbar.setObjectName("statusbar")
139     MainWindow.setStatusBar(self.statusbar)
140     self.actionLoadCountries = QtWidgets.QAction(MainWindow)
141     self.actionLoadCountries.setObjectName("actionLoadCountries")
142     self.actionSavetoFile = QtWidgets.QAction(MainWindow)
143     self.actionSavetoFile.setEnabled(False)
144     self.actionSavetoFile.setObjectName("actionSavetoFile")
145     self.actionExit = QtWidgets.QAction(MainWindow)
146     self.actionExit.setObjectName("actionExit")
147     self.menuFile.addAction(self.actionLoadCountries)
148     self.menuFile.addAction(self.actionSavetoFile)
149     self.menuFile.addSeparator()
150     self.menuFile.addAction(self.actionExit)
151     self.menubar.addAction(self.menuFile.menuAction())
152
153     self.retranslateUi(MainWindow)
154     QtCore.QMetaObject.connectSlotsByName(MainWindow)
155
156 def retranslateUi(self, MainWindow):
157     translate = QtCore.QCoreApplication.translate
158     MainWindow.setWindowTitle(translate("MainWindow", "Countries of the World"))
159     self.labelPeople.setText(translate("MainWindow", "Choose a country from the
list:"))
160     self.labelPop.setText(translate("MainWindow", "Population:"))
161     self.labelDensity.setText(translate("MainWindow", "Population Density"))
162     self.radioButtonKM.setText(translate("MainWindow", "Per Square KM"))
163     self.labelTotalArea.setText(translate("MainWindow", "Total Area in"))
164     self.radioButtonMile.setText(translate("MainWindow", "Per Square Mile"))
165     self.labelPercent.setText(translate("MainWindow", "Percentage of World
Population:"))
166     self.pushButtonUpdate.setText(translate("MainWindow", "Update Population"))
167     self.menuFile.setTitle(translate("MainWindow", "File"))
168     self.actionLoadCountries.setText(translate("MainWindow", "Load Countries"))
169     self.actionSavetoFile.setText(translate("MainWindow", "Save to File"))
170     self.actionExit.setText(translate("MainWindow", "Exit"))

```

The is the text version of the code generated by QTDesigner from within PYQT5

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3 <class>MainWindow</class>
4 <widget class="QMainWindow" name="MainWindow">
5 <property name="geometry">
6 <rect>
7 <x>0</x>
8 <y>0</y>
9 <width>715</width>
10 <height>602</height>
11 </rect>
12 </property>
13 <property name="windowTitle">
14 <string>Countries of the World</string>
15 </property>
16 <widget class="QWidget" name="centralwidget">
17 <widget class="QListWidget" name="listWidgetPeople">
18 <property name="geometry">
19 <rect>
20 <x>20</x>
21 <y>40</y>
22 <width>291</width>
23 <height>511</height>
24 </rect>
25 </property>
26 <property name="font">
27 <font>
28 <pointsize>11</pointsize>
29 </font>
30 </property>
31 </widget>
32 <widget class="QLabel" name="labelPeople">
33 <property name="geometry">
34 <rect>
35 <x>20</x>
36 <y>0</y>
37 <width>291</width>
38 <height>51</height>
39 </rect>
40 </property>
41 <property name="font">
42 <font>
43 <pointsize>13</pointsize>
44 <weight>75</weight>
45 <bold>true</bold>
46 </font>
47 </property>
48 <property name="text">
49 <string>Choose a country from the list:</string>
50 </property>
51 </widget>
52 <widget class="QFrame" name="frameCountries">
53 <property name="geometry">
54 <rect>
55 <x>320</x>
56 <y>-10</y>
57 <width>381</width>
58 <height>551</height>
59 </rect>
60 </property>
61 <property name="frameShape">
62 <enum>QFrame::StyledPanel</enum>
63 </property>
64 <property name="frameShadow">
65 <enum>QFrame::Raised</enum>
66 </property>
67 <widget class="QLabel" name="labelPop">

```

```
68     <property name="geometry">
69         <rect>
70             <x>20</x>
71             <y>250</y>
72             <width>121</width>
73             <height>41</height>
74         </rect>
75     </property>
76     <property name="font">
77         <font>
78             <pointsize>12</pointsize>
79         </font>
80     </property>
81     <property name="text">
82         <string>Population:</string>
83     </property>
84 </widget>
85 <widget class="QComboBox" name="comboBoxMft">
86     <property name="geometry">
87         <rect>
88             <x>120</x>
89             <y>370</y>
90             <width>91</width>
91             <height>22</height>
92         </rect>
93     </property>
94     <property name="font">
95         <font>
96             <pointsize>12</pointsize>
97         </font>
98     </property>
99 </widget>
100 <widget class="QLabel" name="labelEditSqm">
101     <property name="geometry">
102         <rect>
103             <x>270</x>
104             <y>360</y>
105             <width>101</width>
106             <height>41</height>
107         </rect>
108     </property>
109     <property name="font">
110         <font>
111             <pointsize>12</pointsize>
112             <weight>75</weight>
113             <bold>true</bold>
114         </font>
115     </property>
116     <property name="text">
117         <string/>
118     </property>
119 </widget>
120 <widget class="QLabel" name="labelDensity">
121     <property name="geometry">
122         <rect>
123             <x>40</x>
124             <y>410</y>
125             <width>141</width>
126             <height>41</height>
127         </rect>
128     </property>
129     <property name="font">
130         <font>
131             <pointsize>12</pointsize>
132         </font>
133     </property>
134     <property name="text">
```



```
135     <string>Population Density</string>
136   </property>
137 </widget>
138 <widget class="QRadioButton" name="radioButtonKM">
139   <property name="geometry">
140     <rect>
141       <x>40</x>
142       <y>470</y>
143       <width>141</width>
144       <height>16</height>
145     </rect>
146   </property>
147   <property name="font">
148     <font>
149       <pointsize>12</pointsize>
150     </font>
151   </property>
152   <property name="text">
153     <string>Per Square KM</string>
154   </property>
155 </widget>
156 <widget class="QLabel" name="labelEditDensity">
157   <property name="geometry">
158     <rect>
159       <x>270</x>
160       <y>440</y>
161       <width>91</width>
162       <height>41</height>
163     </rect>
164   </property>
165   <property name="font">
166     <font>
167       <pointsize>12</pointsize>
168       <weight>75</weight>
169       <bold>true</bold>
170     </font>
171   </property>
172   <property name="text">
173     <string/>
174   </property>
175 </widget>
176 <widget class="QLabel" name="labelTotalArea">
177   <property name="geometry">
178     <rect>
179       <x>20</x>
180       <y>360</y>
181       <width>121</width>
182       <height>41</height>
183     </rect>
184   </property>
185   <property name="font">
186     <font>
187       <pointsize>12</pointsize>
188     </font>
189   </property>
190   <property name="text">
191     <string>Total Area in</string>
192   </property>
193 </widget>
194 <widget class="QRadioButton" name="radioButtonMile">
195   <property name="geometry">
196     <rect>
197       <x>40</x>
198       <y>450</y>
199       <width>141</width>
200       <height>16</height>
201     </rect>
```

```
202     </property>
203     <property name="font">
204         <font>
205             <pointsize>12</pointsize>
206         </font>
207     </property>
208     <property name="text">
209         <string>Per Square Mile</string>
210     </property>
211 </widget>
212 <widget class="QLineEdit" name="lineEditPop">
213     <property name="geometry">
214         <rect>
215             <x>110</x>
216             <y>260</y>
217             <width>251</width>
218             <height>21</height>
219         </rect>
220     </property>
221 </widget>
222 <widget class="QLabel" name="labelPicture">
223     <property name="geometry">
224         <rect>
225             <x>30</x>
226             <y>100</y>
227             <width>241</width>
228             <height>111</height>
229         </rect>
230     </property>
231     <property name="text">
232         <string/>
233     </property>
234 </widget>
235 <widget class="QLabel" name="labelEditPopulation">
236     <property name="geometry">
237         <rect>
238             <x>270</x>
239             <y>500</y>
240             <width>91</width>
241             <height>41</height>
242         </rect>
243     </property>
244     <property name="font">
245         <font>
246             <pointsize>12</pointsize>
247             <weight>75</weight>
248             <bold>true</bold>
249         </font>
250     </property>
251     <property name="text">
252         <string/>
253     </property>
254 </widget>
255 <widget class="QLabel" name="labelPercent">
256     <property name="geometry">
257         <rect>
258             <x>20</x>
259             <y>500</y>
260             <width>261</width>
261             <height>41</height>
262         </rect>
263     </property>
264     <property name="font">
265         <font>
266             <pointsize>12</pointsize>
267         </font>
268     </property>
```

```

269     <property name="text">
270         <string>Percentage of World Population:</string>
271     </property>
272 </widget>
273 <widget class="QLabel" name="labelCountry">
274     <property name="geometry">
275         <rect>
276             <x>0</x>
277             <y>40</y>
278             <width>441</width>
279             <height>41</height>
280         </rect>
281     </property>
282     <property name="font">
283         <font>
284             <pointsize>12</pointsize>
285             <weight>75</weight>
286             <bold>true</bold>
287         </font>
288     </property>
289     <property name="text">
290         <string/>
291     </property>
292 </widget>
293 <widget class="QPushButton" name="pushButtonUpdate">
294     <property name="geometry">
295         <rect>
296             <x>220</x>
297             <y>290</y>
298             <width>141</width>
299             <height>31</height>
300         </rect>
301     </property>
302     <property name="font">
303         <font>
304             <pointsize>12</pointsize>
305         </font>
306     </property>
307     <property name="text">
308         <string>Update Population</string>
309     </property>
310 </widget>
311 </widget>
312 </widget>
313 <widget class="QMenuBar" name="menubar">
314     <property name="geometry">
315         <rect>
316             <x>0</x>
317             <y>0</y>
318             <width>715</width>
319             <height>21</height>
320         </rect>
321     </property>
322     <widget class="QMenu" name="menuFile">
323         <property name="title">
324             <string>File</string>
325         </property>
326         <addaction name="actionLoadCountries"/>
327         <addaction name="actionSaveToFile"/>
328         <addaction name="separator"/>
329         <addaction name="actionExit"/>
330     </widget>
331     <addaction name="menuFile"/>
332 </widget>
333 <widget class="QStatusBar" name="statusbar"/>
334 <action name="actionLoadCountries">
335     <property name="text">

```

```
336     <string>Load Countries</string>
337   </property>
338 </action>
339 <action name="actionSaveToFile">
340   <property name="enabled">
341     <bool>false</bool>
342   </property>
343   <property name="text">
344     <string>Save to File</string>
345   </property>
346 </action>
347 <action name="actionExit">
348   <property name="text">
349     <string>Exit</string>
350   </property>
351 </action>
352 </widget>
353 <resources/>
354 <connections/>
355 </ui>
```