Summer Training Report
On
Fantasy Cricket Game with Python

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

(COMPUTER SCIENCE & ENGINEERING)

June-July 2022

Submitted By:

Kunal Sharma(20BCS5686)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Chandigarh University, Gharuan, Mohali

DECLARATION

I, student of Bachelor of Engineering in Computer Science & Engineering, 3rd Semester, session: June-July 2021, Chandigarh University, hereby declare that the work presented in this Project Report entitled “E Commerce Website” is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patent work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Kunal Sharma

Signature of the Student

ACKNOWLEDGEMENT

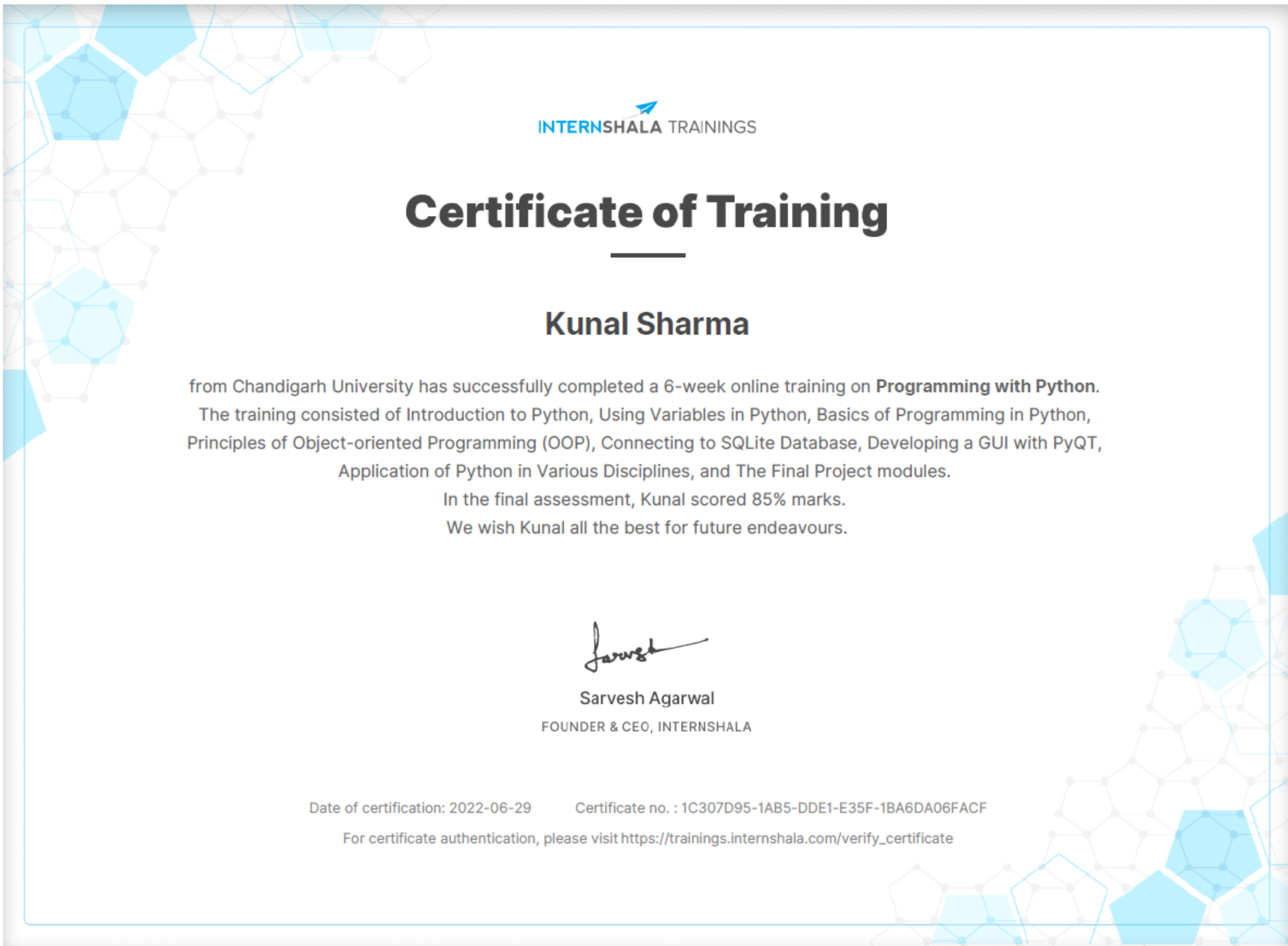
It is my pleasure to be indebted to various people, who directly or indirectly helped me to learn this course and who influenced my thinking, behaviour and acts during study.

I am thankful to Internshala for this course and for providing me the platform to learn and apply the skills learned through a project.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best viable way, and I will continue to work on their improvement, to attain desired career objectives. Hope to continue cooperation with all of you in the future.

Kunal Sharma

TRAINING CERTIFICATE



INTERNSHALA TRAININGS

Certificate of Training

Kunal Sharma

from Chandigarh University has successfully completed a 6-week online training on **Programming with Python**. The training consisted of Introduction to Python, Using Variables in Python, Basics of Programming in Python, Principles of Object-oriented Programming (OOP), Connecting to SQLite Database, Developing a GUI with PyQT, Application of Python in Various Disciplines, and The Final Project modules.

In the final assessment, Kunal scored 85% marks.

We wish Kunal all the best for future endeavours.

Sarvesh Agarwal
FOUNDER & CEO, INTERNSHALA

Date of certification: 2022-06-29 Certificate no. : 1C307D95-1AB5-DDE1-E35F-1BA6DA06FACF

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

TABLE OF CONTENTS

INTRODUCTION

Python Language Introduction

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted -** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive -** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented -** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language -** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

PYTHON FEATURES

Python's features include −

- **Easy-to-learn -** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read -** Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain -** Python's source code is fairly easy-to-maintain.

- **A broad standard library -** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode -** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable -** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable -** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases -** Python provides interfaces to all major commercial databases.

- **GUI Programming -** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable -** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

TRAINING CONTENTS

1. **Introduction to Python**

   Learn how to install Python, distinguish between important data types and use basic features of the Python interpreter, IDLE.

2. **Using Variables in Python**

   Learn about numeric, string, sequence and dictionary data types and relevant operations while practicing Python syntax.

3. **Basics of Programming in Python**

   Learn how to write programs using conditionals, loops, iterators and generators, functions and modules and packages.

4. **Principles of Object-oriented Programming (OOP)**

   Learn about the important features of Object-oriented Programming while using Classes and Objects, two main aspects of the OOP paradigm.

5. **Connecting to SQLite Database**

   Learn about relational databases while learning how to store and retrieve data from an SQLite database through Python.

6. **Developing a GUI with PyQT**

   Learn how to install PyQt5 toolkit, Qt Designer and create a graphical user interface using common widgets and menu systems.

7. **Application of Python in Various Disciplines**

   Learn about various resources to extend your learning for the Python programming language.

Developing a GUI with PyQT

Developing a GUI with PyQT GUI and Event Driven Programming-

GUI:- Graphical user interface (GUI), a computer program that enables a person to communicate with a computer through the use of symbols, visual metaphors, and pointing devices. Best known for its implementation in Apple Inc.'s Macintosh and Microsoft Corporation's Windows operating system, the GUI has replaced the arcane and difficult textual interfaces of earlier computing with a relatively intuitive system that has made computer operation not only easier to learn but more pleasant and natural. The GUI is now the standard computer interface, and its components have themselves become unmistakable cultural artifacts.

Event driven Programming:- An event-driven program is one that largely responds to user events or other similar input. The concept of event-driven programming is an important one in application development and other kinds of programming, and has spawned the emergence of event handlers and other resources. The idea in event-driven programming is that the program is designed to react. It reacts to specific kinds of input from users, whether it's a click on a command button, a choice from a drop-down list, an entry into a text box, or other kinds of user events.

PyQT :- PyQt is a GUI widgets toolkit. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. PyQt was developed by RiverBank Computing Ltd. The latest version of PyQt can be downloaded from its official website. PyQt is a blend of Python programming language and the Qt library. PyQT API is a collection of more than 400 classes.

Major classes in PyQT :

QObject is at the top of the class hierarchy. It is the base class of all the Qt objects.
QApplication class manages the main settings and control flow.
QWidget is the base class for all user interface objects.

QDialogue and Qframe classes are derived from the QWidget class.
Using Common Widgets.

Qt Designer

Qt Designer is the Qt tool for designing and building graphical user interfaces (GUIs) with Qt Widgets. You can compose and customize your windows or dialogs in a what-you-see-is -what.

you-get (WYSIWYG) manner, and test them using different styles and resolutions.

Widgets and forms created with Qt Designer integrate seamlessly with programmed code, using Qt's signals and slots mechanism, so that you can easily assign behavior to graphical elements.

All properties set in Qt Designer can be changed dynamically within the code. Furthermore, features like widget promotion and custom plugins allow you to use your own components with Qt Designer.

Qt Designer helps only in designing GUI file, then it needs to be converted into py file. By using pyuic 5 utility xml file can be converted into .py file and is installed along with PyQT5 package.

We use pyuic5-x myui.ui-o myui.py command to convert ui file to python file. For the sake of simplicity, we place myuic.ui file in the same folder where pyuic5.exe file is placed.

PROFILE OF THE PROBLEM

Create a Fantasy Cricket game in Python. The game should have all the features displayed in the mock-up screens in the scenario. To calculate the points for each player, we can use rules similar to the sample rules displayed below.

Sample of Rules

Batting

- 1 point for 2 runs scored
- Additional 5 points for half century
- Additional 10 points for century
- 2 points for strike rate (runs/balls faced) of 80-100
- Additional 4 points for strike rate>100
- 1 point for hitting a boundary (four) and 2 points for over boundary (six)

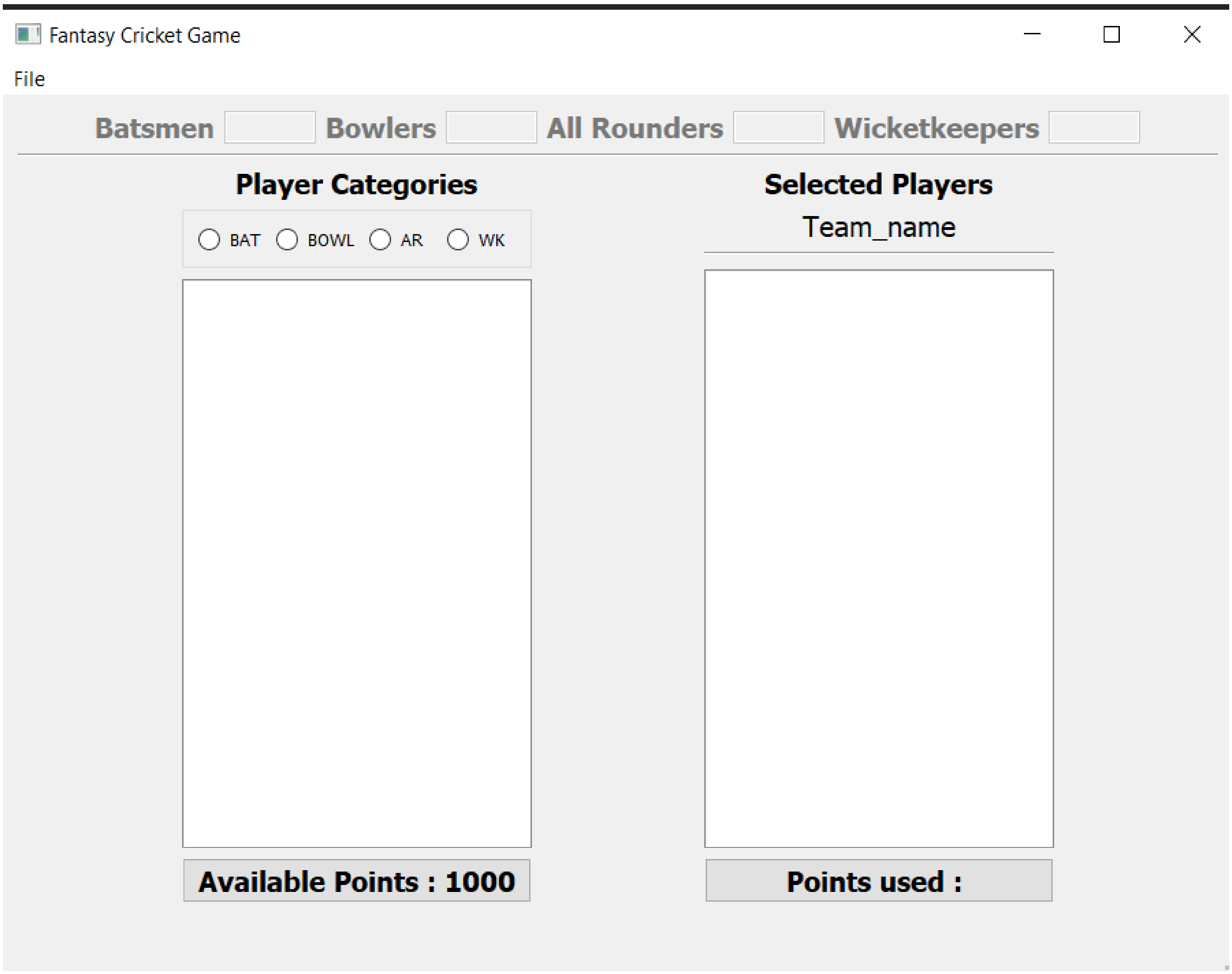
Bowling

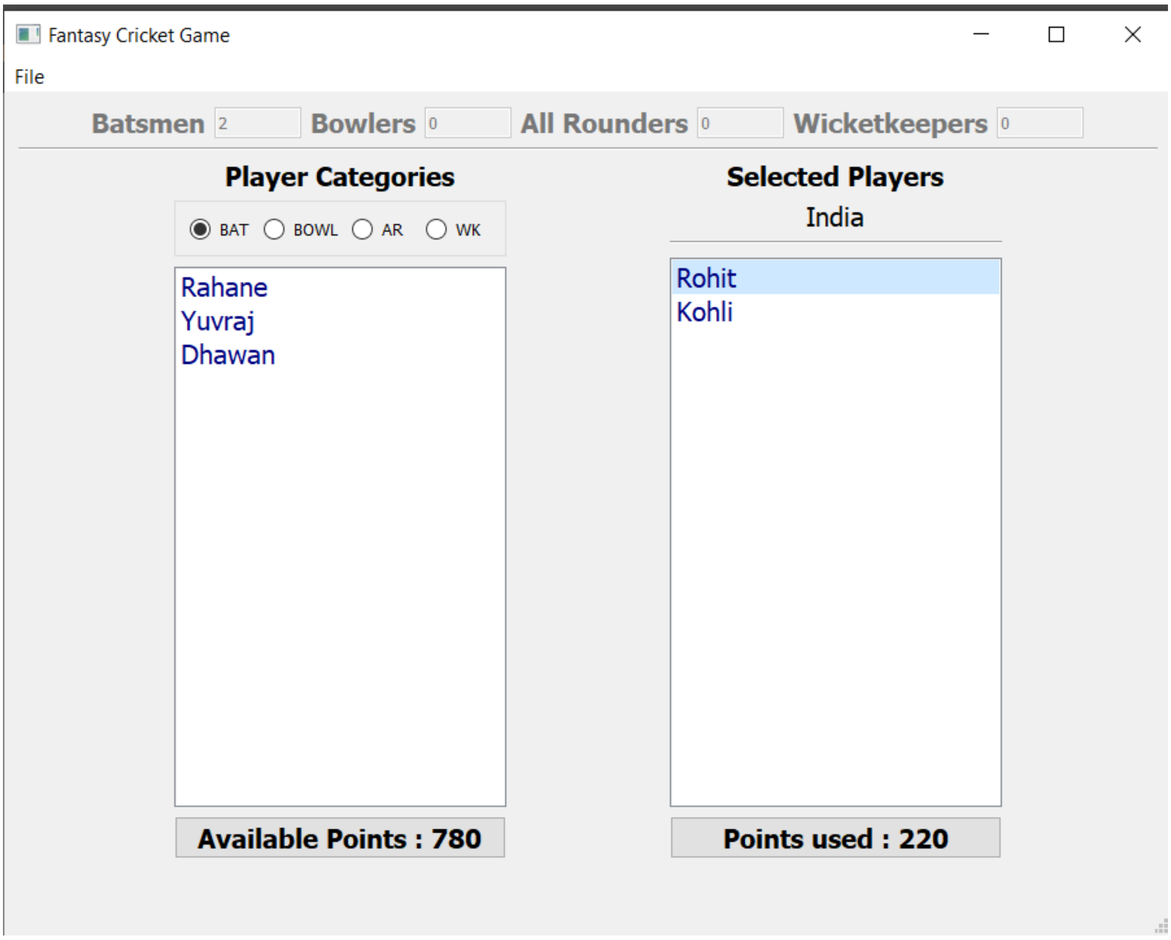
- 10 points for each wicket
- Additional 5 points for three wickets per innings
- Additional 10 points for 5 wickets or more in innings
- 4 points for economy rate (runs given per over) between 3.5 and 4.5
- 7 points for economy rate between 2 and 3.5
- 10 points for economy rate less than 2

Fielding

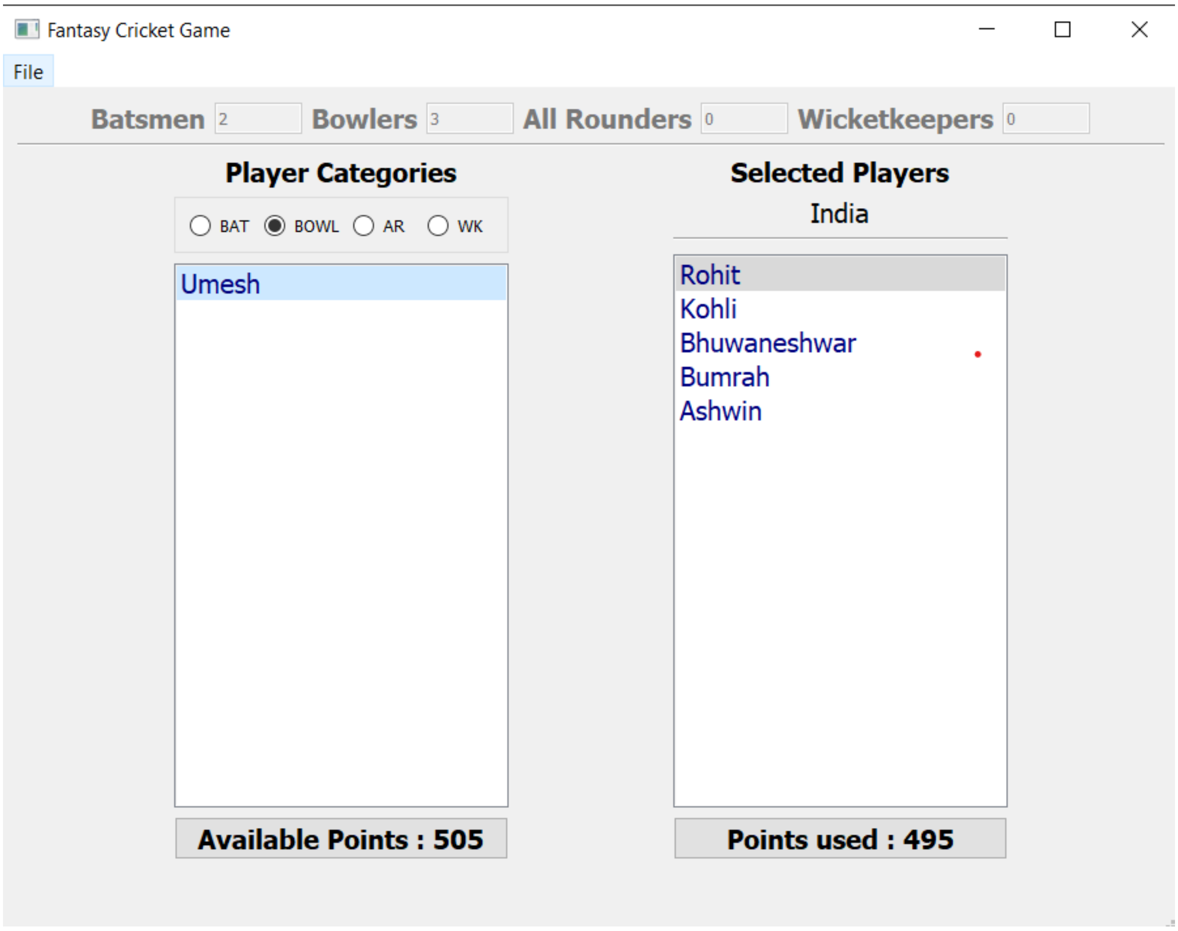
- 10 points each for catch/stumping/run out

Testing /Deployment

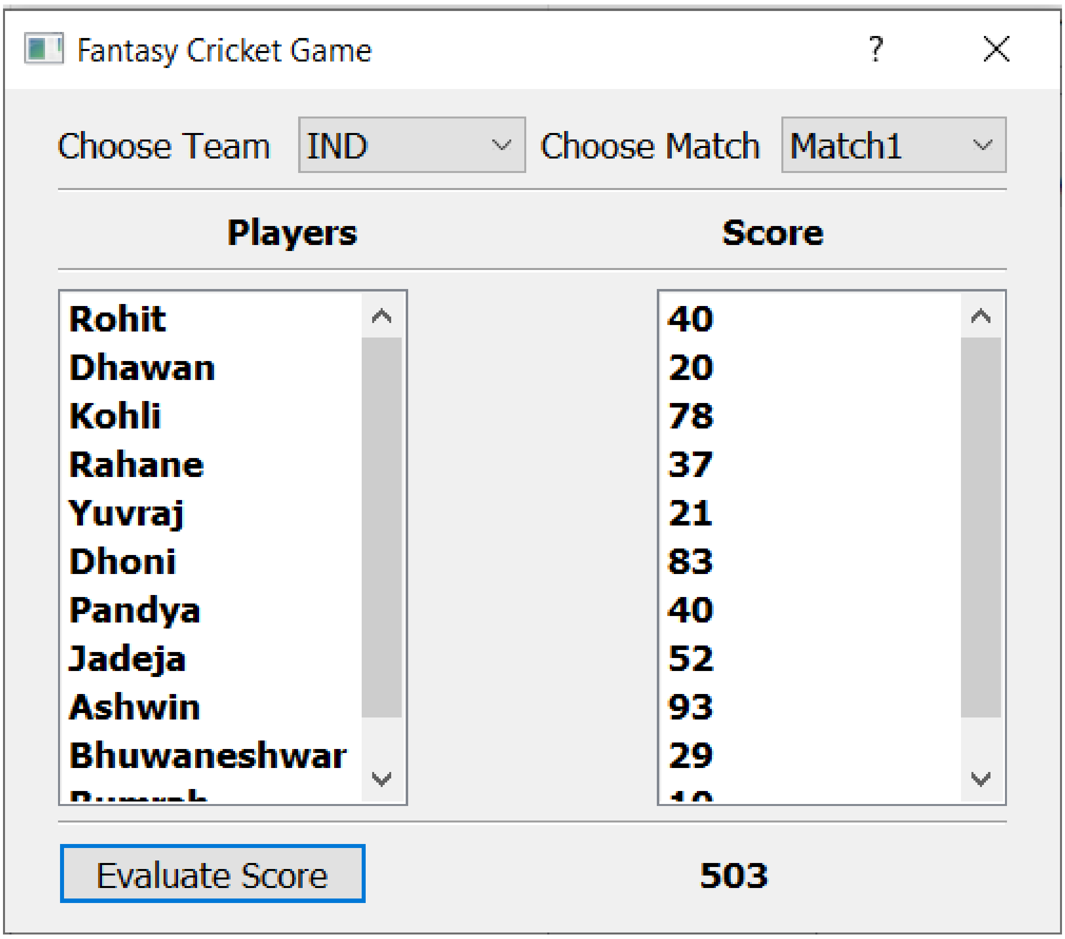




Fantasy Cricket Game
File
Batsmen 2
Bowlers 0
All Rounders 0
Wicketkeepers 0
Player Categories
● BAT ○ BOWL ○ AR ○ WK
Rahane
Yuvraj
Dhawan
Available Points : 780
Selected Players
India
Rohit
Kohli
Points used : 220



Fantasy Cricket Game
File
Batsmen 2
Bowlers 3
All Rounders 0
Wicketkeepers 0
Player Categories
○ BAT ● BOWL ○ AR ○ WK
Umesh
Available Points : 505
Selected Players
India
Rohit
Kohli
Bhuwaneshwar
Bumrah
Ashwin
Points used : 495



Fantasy Cricket Game

Choose Team IND

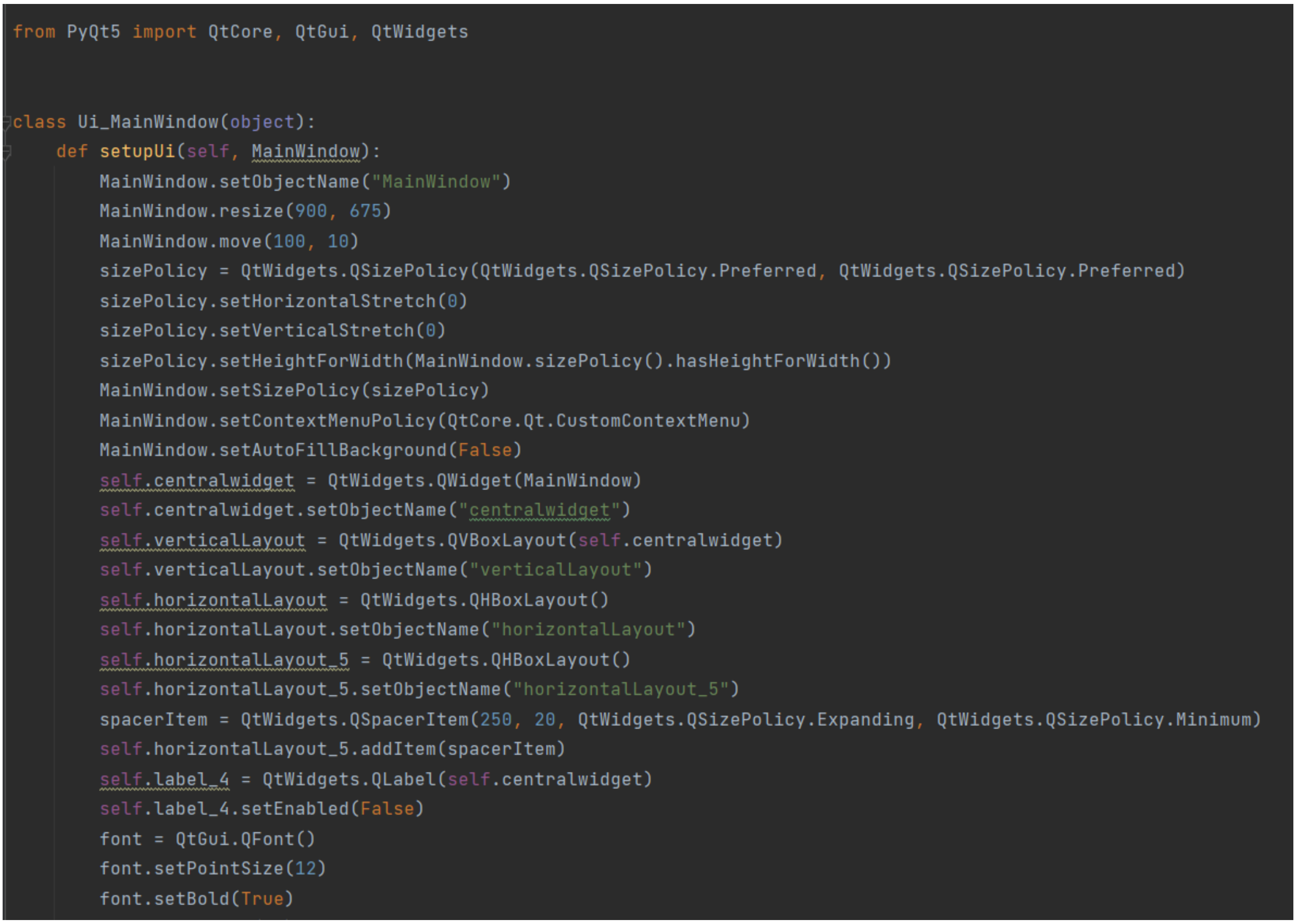
Choose Match Match1

Players	Score
Rohit	40
Dhawan	20
Kohli	78
Rahane	37
Yuvraj	21
Dhoni	83
Pandya	40
Jadeja	52
Ashwin	93
Bhuwaneshwar	29

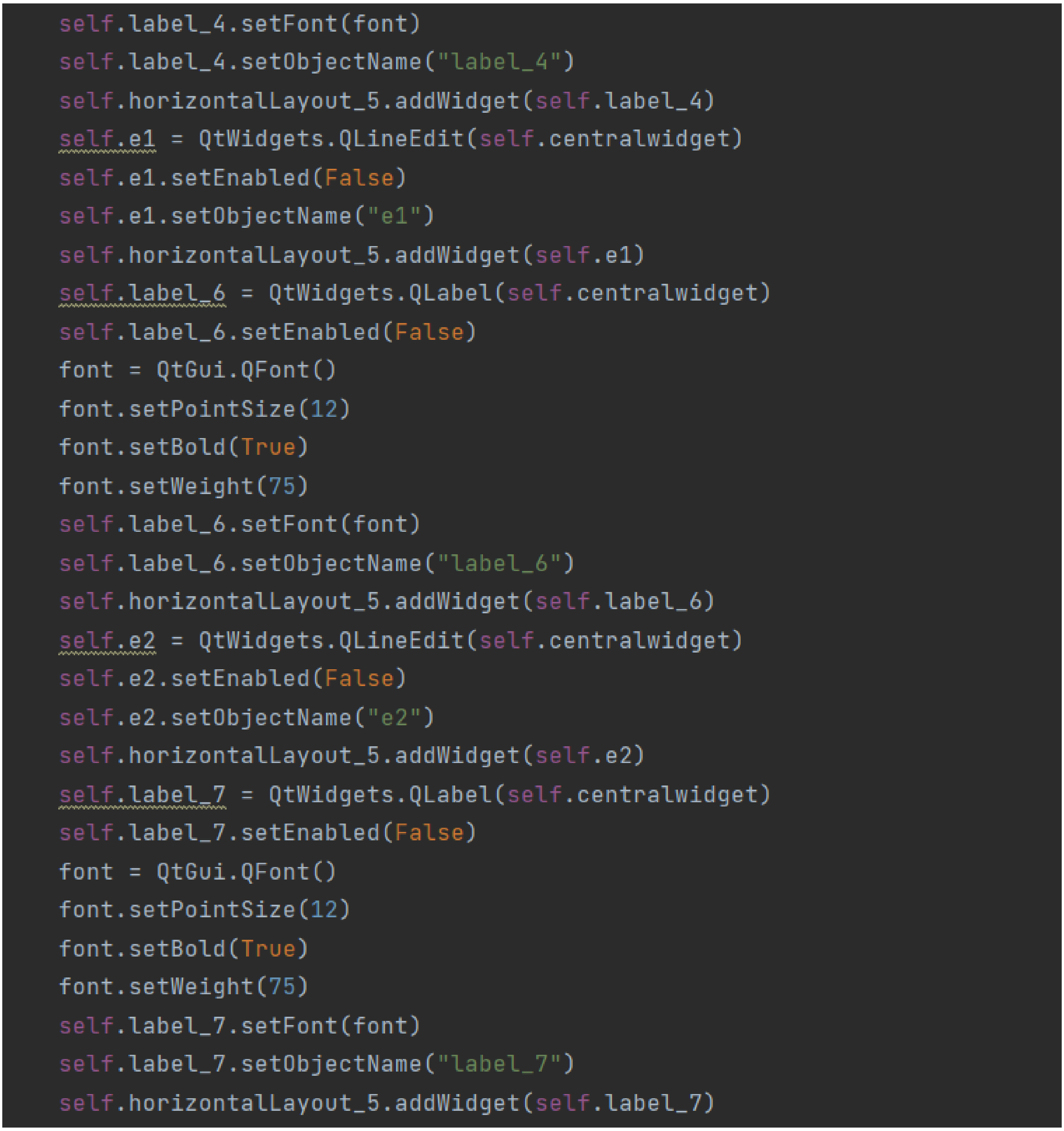
503

Project Code

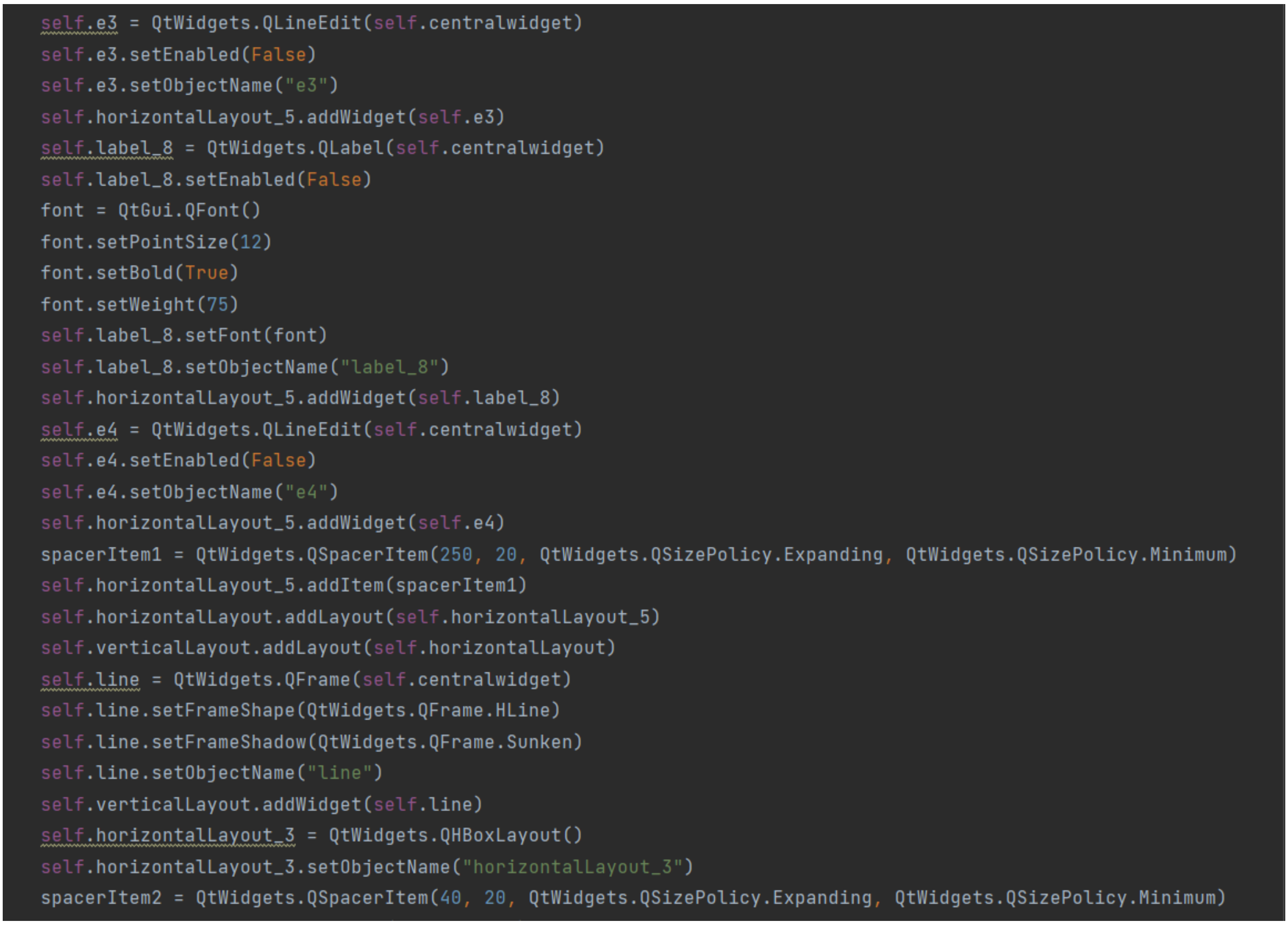
Main.py



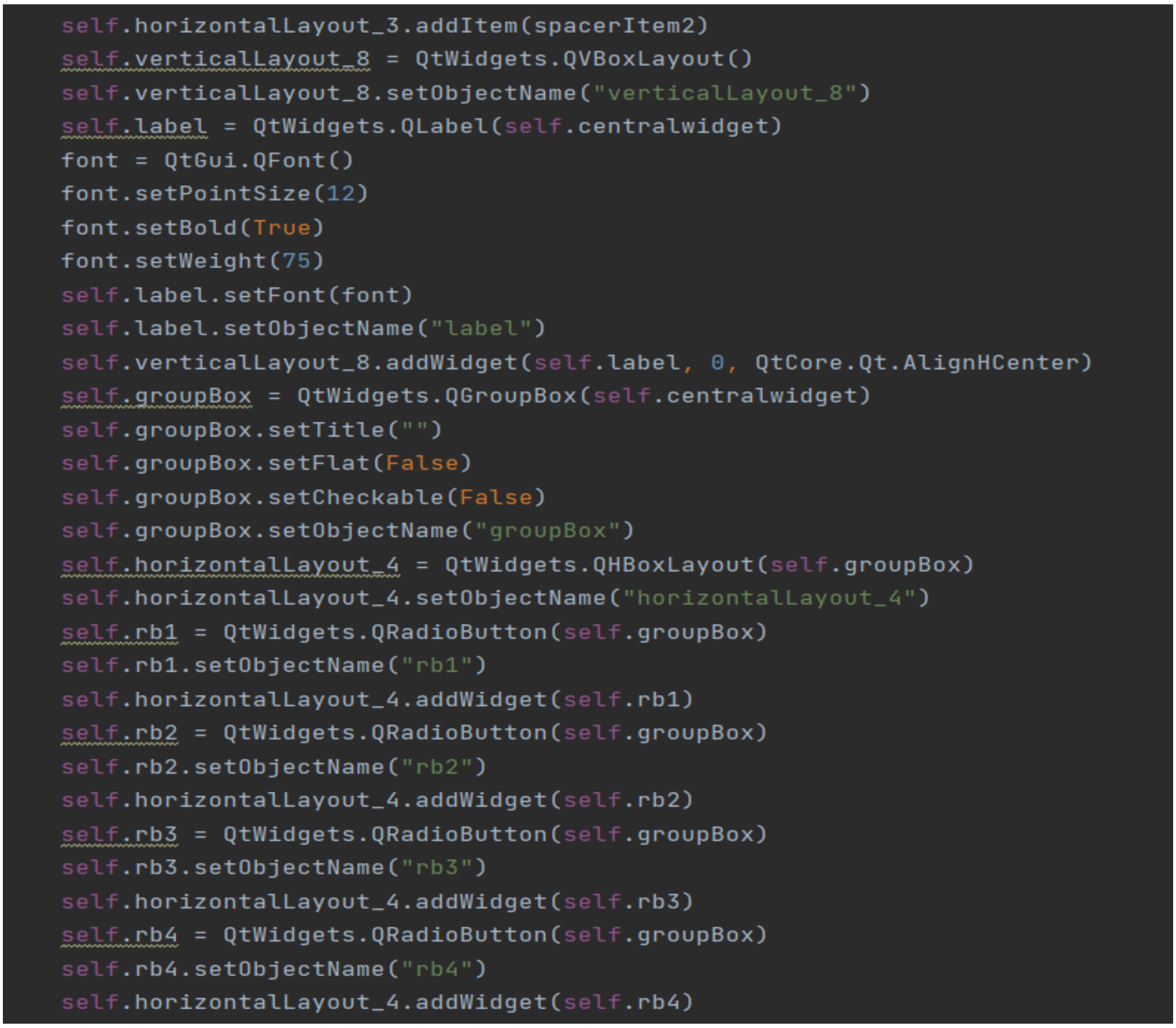
```python
from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(900, 675)
        MainWindow.move(100, 10)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred, QtWidgets.QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())
        MainWindow.setSizePolicy(sizePolicy)
        MainWindow.setContextMenuPolicy(QtCore.Qt.CustomContextMenu)
        MainWindow.setAutoFillBackground(False)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.centralwidget)
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.horizontalLayout_5 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_5.setObjectName("horizontalLayout_5")
        spacerItem = QtWidgets.QSpacerItem(250, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
        self.horizontalLayout_5.addItem(spacerItem)
        self.label_4 = QtWidgets.QLabel(self.centralwidget)
        self.label_4.setEnabled(False)
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
```



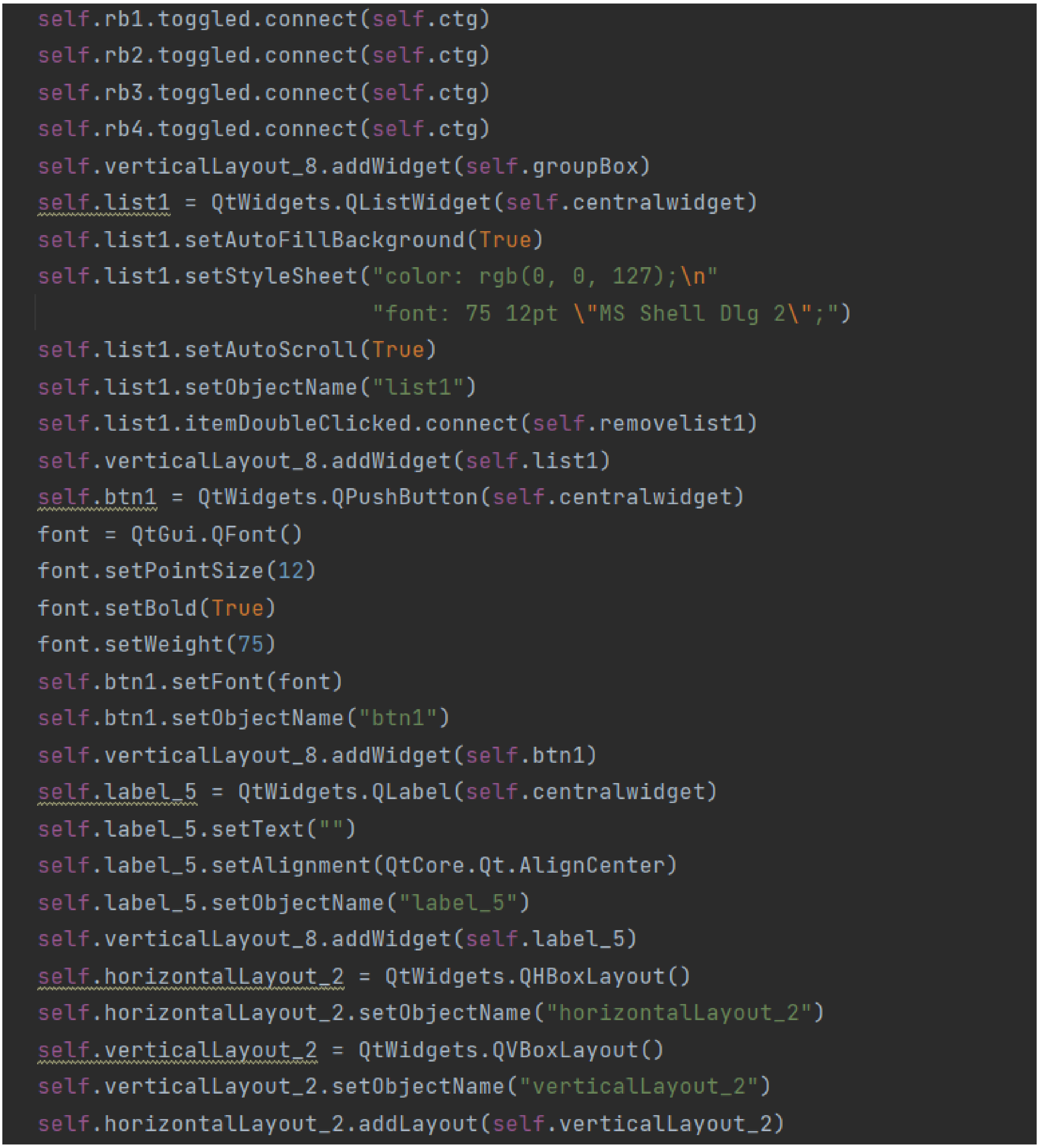
```
    self.label_4.setFont(font)
    self.label_4.setObjectName("label_4")
    self.horizontalLayout_5.addWidget(self.label_4)
    self.e1 = QtWidgets.QLineEdit(self.centralwidget)
    self.e1.setEnabled(False)
    self.e1.setObjectName("e1")
    self.horizontalLayout_5.addWidget(self.e1)
    self.label_6 = QtWidgets.QLabel(self.centralwidget)
    self.label_6.setEnabled(False)
    font = QtGui.QFont()
    font.setPointSize(12)
    font.setBold(True)
    font.setWeight(75)
    self.label_6.setFont(font)
    self.label_6.setObjectName("label_6")
    self.horizontalLayout_5.addWidget(self.label_6)
    self.e2 = QtWidgets.QLineEdit(self.centralwidget)
    self.e2.setEnabled(False)
    self.e2.setObjectName("e2")
    self.horizontalLayout_5.addWidget(self.e2)
    self.label_7 = QtWidgets.QLabel(self.centralwidget)
    self.label_7.setEnabled(False)
    font = QtGui.QFont()
    font.setPointSize(12)
    font.setBold(True)
    font.setWeight(75)
    self.label_7.setFont(font)
    self.label_7.setObjectName("label_7")
    self.horizontalLayout_5.addWidget(self.label_7)
```



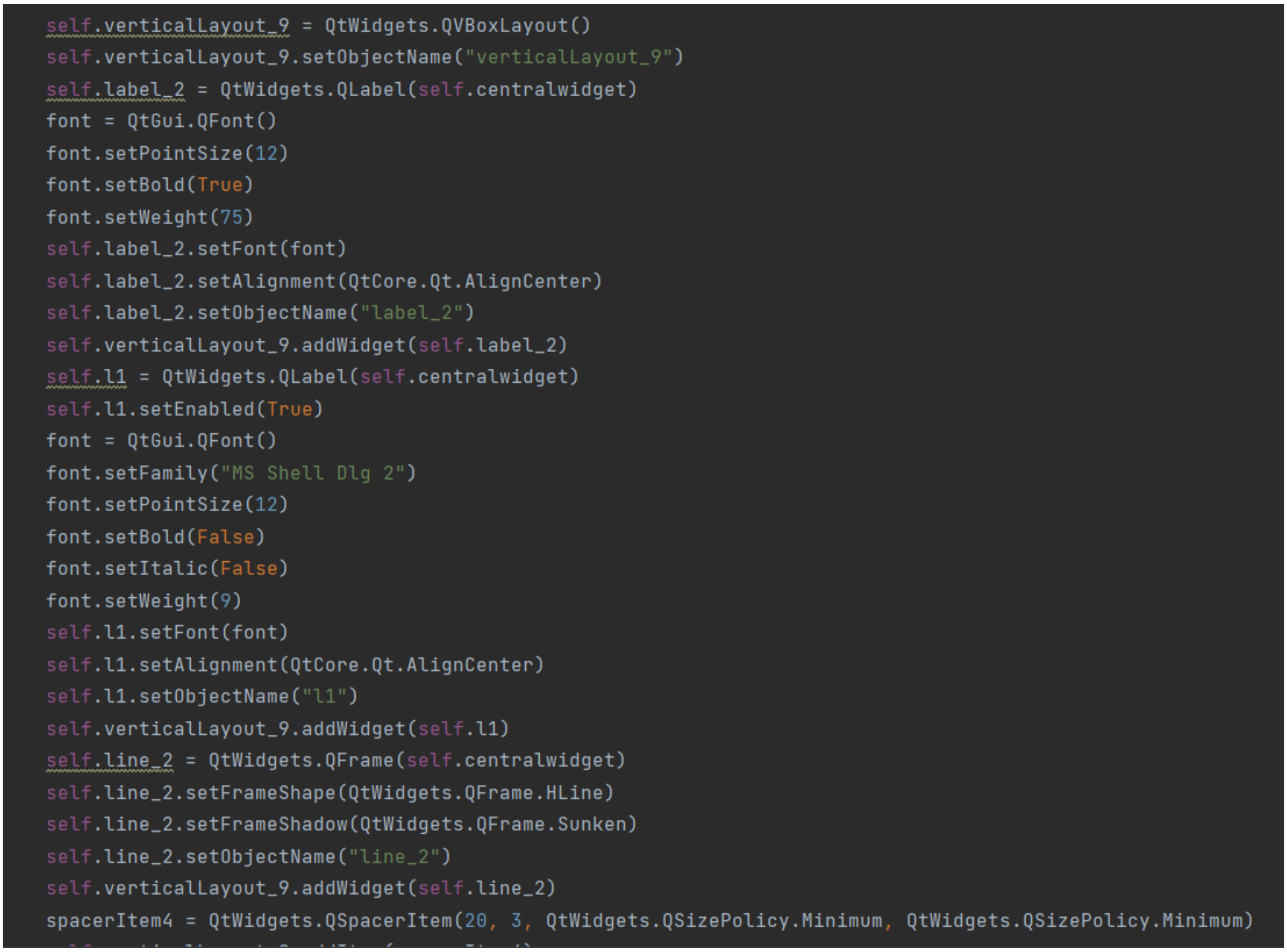
```
self.e3 = QtWidgets.QLineEdit(self.centralwidget)
self.e3.setEnabled(False)
self.e3.setObjectName("e3")
self.horizontalLayout_5.addWidget(self.e3)
self.label_8 = QtWidgets.QLabel(self.centralwidget)
self.label_8.setEnabled(False)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.horizontalLayout_5.addWidget(self.label_8)
self.e4 = QtWidgets.QLineEdit(self.centralwidget)
self.e4.setEnabled(False)
self.e4.setObjectName("e4")
self.horizontalLayout_5.addWidget(self.e4)
spacerItem1 = QtWidgets.QSpacerItem(250, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
self.horizontalLayout_5.addItem(spacerItem1)
self.horizontalLayout.addLayout(self.horizontalLayout_5)
self.verticalLayout.addLayout(self.horizontalLayout)
self.line = QtWidgets.QFrame(self.centralwidget)
self.line.setFrameShape(QtWidgets.QFrame.HLine)
self.line.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line.setObjectName("line")
self.verticalLayout.addWidget(self.line)
self.horizontalLayout_3 = QtWidgets.QHBoxLayout()
self.horizontalLayout_3.setObjectName("horizontalLayout_3")
spacerItem2 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
```



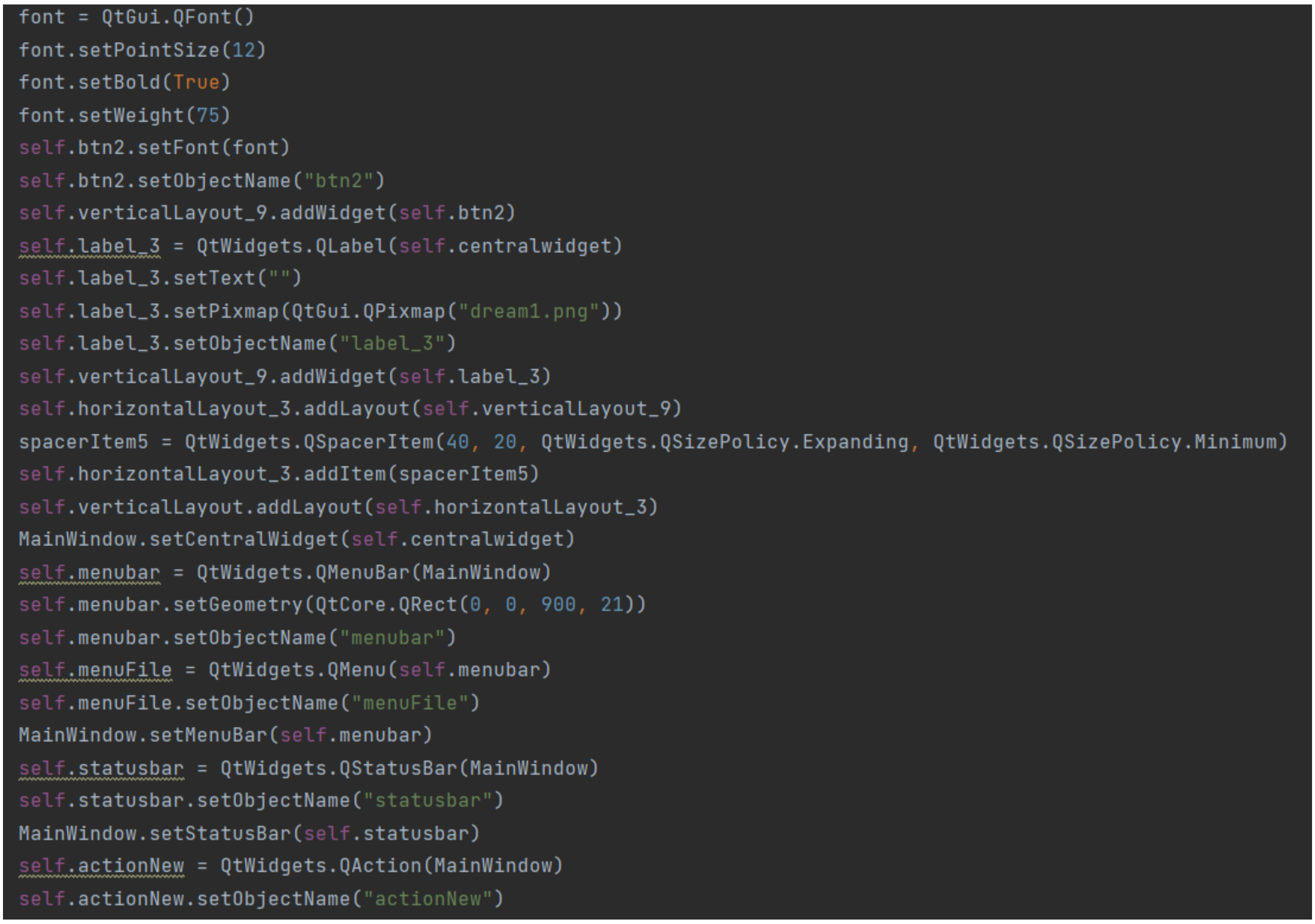
```
self.horizontalLayout_3.addItem(spacerItem2)
self.verticalLayout_8 = QtWidgets.QVBoxLayout()
self.verticalLayout_8.setObjectName("verticalLayout_8")
self.label = QtWidgets.QLabel(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.label.setFont(font)
self.label.setObjectName("label")
self.verticalLayout_8.addWidget(self.label, 0, QtCore.Qt.AlignHCenter)
self.groupBox = QtWidgets.QGroupBox(self.centralwidget)
self.groupBox.setTitle("")
self.groupBox.setFlat(False)
self.groupBox.setCheckable(False)
self.groupBox.setObjectName("groupBox")
self.horizontalLayout_4 = QtWidgets.QHBoxLayout(self.groupBox)
self.horizontalLayout_4.setObjectName("horizontalLayout_4")
self.rb1 = QtWidgets.QRadioButton(self.groupBox)
self.rb1.setObjectName("rb1")
self.horizontalLayout_4.addWidget(self.rb1)
self.rb2 = QtWidgets.QRadioButton(self.groupBox)
self.rb2.setObjectName("rb2")
self.horizontalLayout_4.addWidget(self.rb2)
self.rb3 = QtWidgets.QRadioButton(self.groupBox)
self.rb3.setObjectName("rb3")
self.horizontalLayout_4.addWidget(self.rb3)
self.rb4 = QtWidgets.QRadioButton(self.groupBox)
self.rb4.setObjectName("rb4")
self.horizontalLayout_4.addWidget(self.rb4)
```



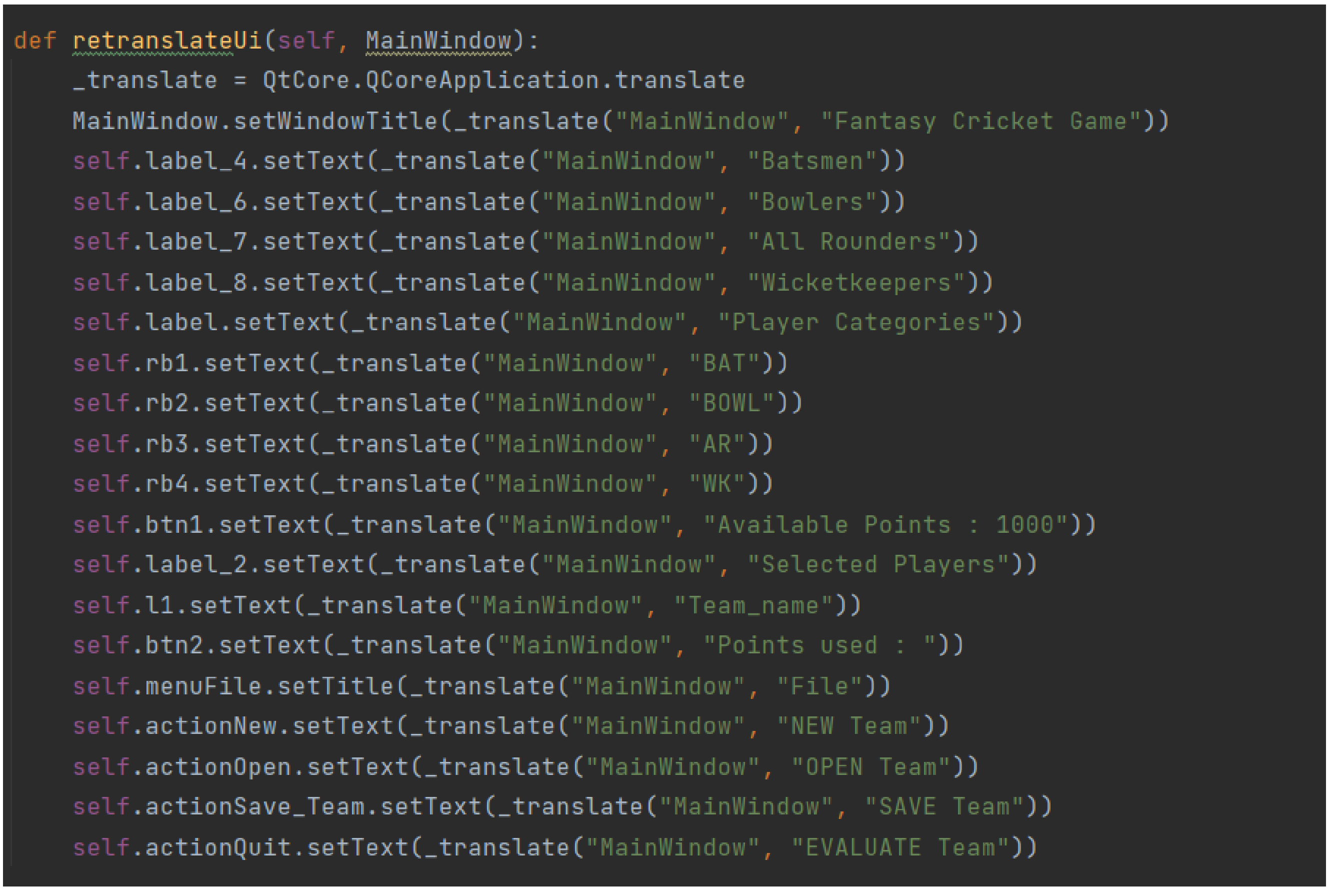
```
self.rb1.toggled.connect(self.ctg)
self.rb2.toggled.connect(self.ctg)
self.rb3.toggled.connect(self.ctg)
self.rb4.toggled.connect(self.ctg)
self.verticalLayout_8.addWidget(self.groupBox)
self.list1 = QtWidgets.QListWidget(self.centralwidget)
self.list1.setAutoFillBackground(True)
self.list1.setStyleSheet("color: rgb(0, 0, 127);\n"
                         "font: 75 12pt \"MS Shell Dlg 2\";")
self.list1.setAutoScroll(True)
self.list1.setObjectName("list1")
self.list1.itemDoubleClicked.connect(self.removelist1)
self.verticalLayout_8.addWidget(self.list1)
self.btn1 = QtWidgets.QPushButton(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.btn1.setFont(font)
self.btn1.setObjectName("btn1")
self.verticalLayout_8.addWidget(self.btn1)
self.label_5 = QtWidgets.QLabel(self.centralwidget)
self.label_5.setText("")
self.label_5.setAlignment(QtCore.Qt.AlignCenter)
self.label_5.setObjectName("label_5")
self.verticalLayout_8.addWidget(self.label_5)
self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
self.horizontalLayout_2.setObjectName("horizontalLayout_2")
self.verticalLayout_2 = QtWidgets.QVBoxLayout()
self.verticalLayout_2.setObjectName("verticalLayout_2")
self.horizontalLayout_2.addLayout(self.verticalLayout_2)
```



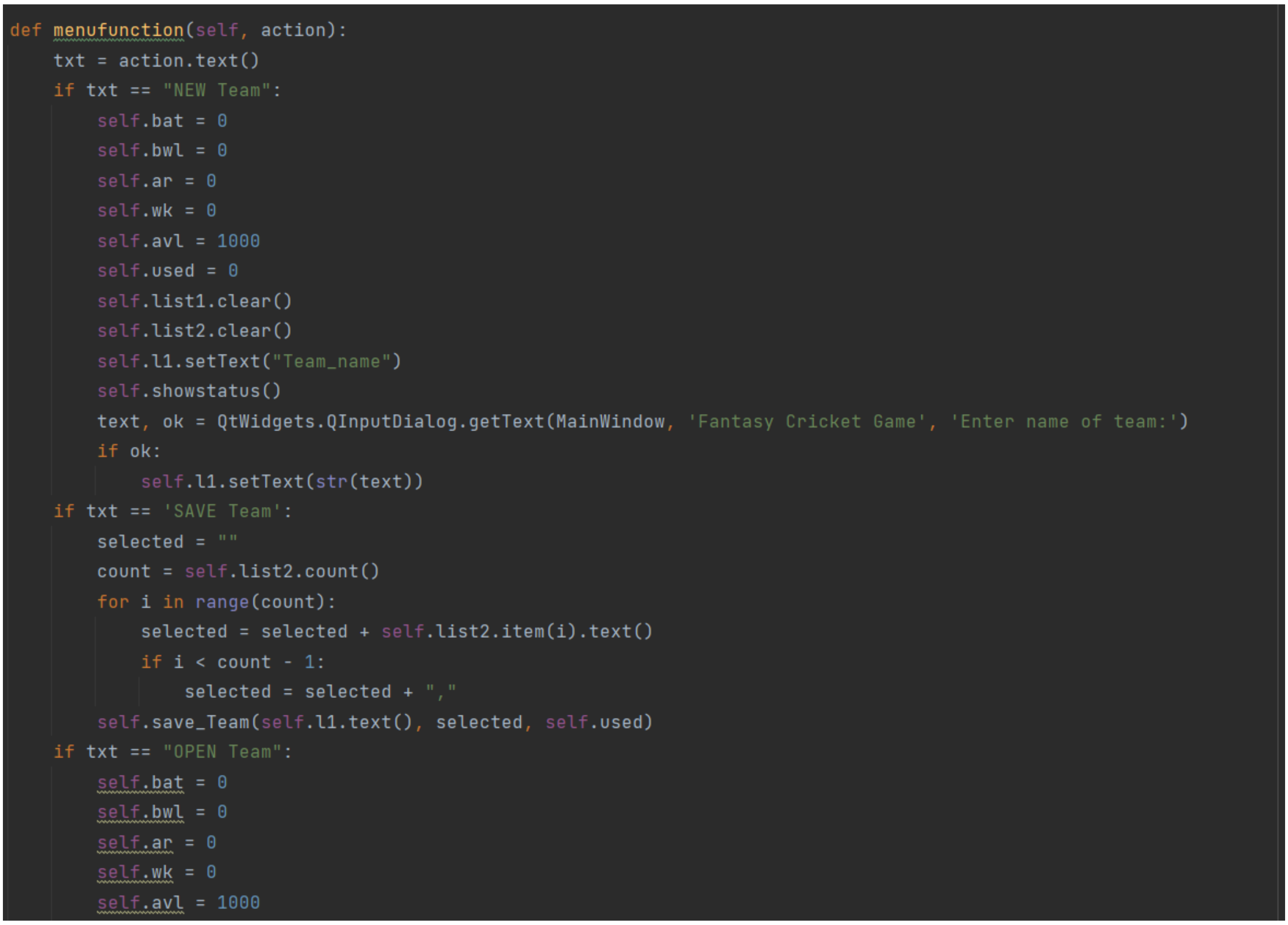
```python
self.verticalLayout_9 = QtWidgets.QVBoxLayout()
self.verticalLayout_9.setObjectName("verticalLayout_9")
self.label_2 = QtWidgets.QLabel(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setAlignment(QtCore.Qt.AlignCenter)
self.label_2.setObjectName("label_2")
self.verticalLayout_9.addWidget(self.label_2)
self.l1 = QtWidgets.QLabel(self.centralwidget)
self.l1.setEnabled(True)
font = QtGui.QFont()
font.setFamily("MS Shell Dlg 2")
font.setPointSize(12)
font.setBold(False)
font.setItalic(False)
font.setWeight(9)
self.l1.setFont(font)
self.l1.setAlignment(QtCore.Qt.AlignCenter)
self.l1.setObjectName("l1")
self.verticalLayout_9.addWidget(self.l1)
self.line_2 = QtWidgets.QFrame(self.centralwidget)
self.line_2.setFrameShape(QtWidgets.QFrame.HLine)
self.line_2.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_2.setObjectName("line_2")
self.verticalLayout_9.addWidget(self.line_2)
spacerItem4 = QtWidgets.QSpacerItem(20, 3, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Minimum)
```



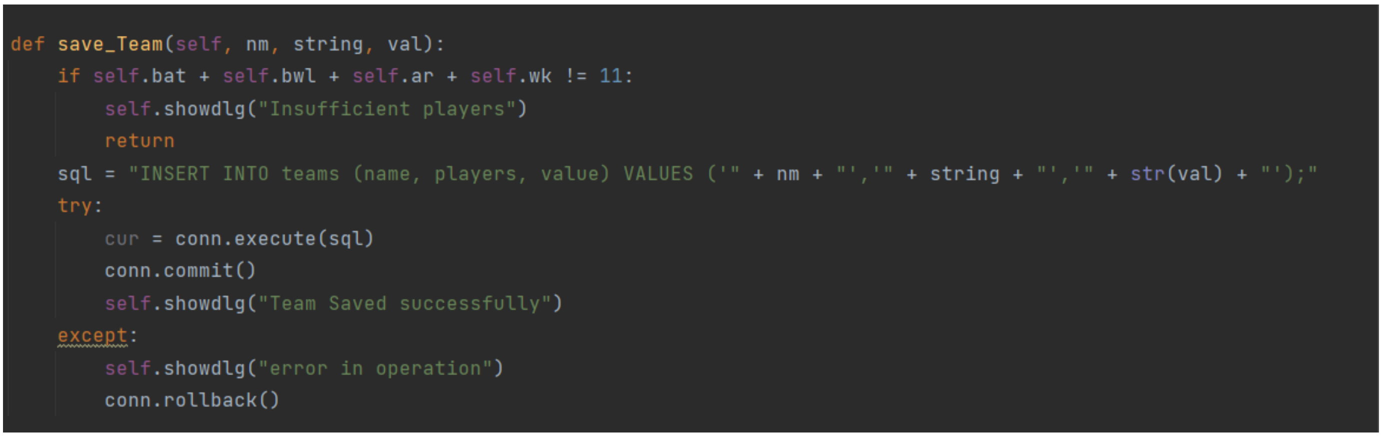
```python
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.btn2.setFont(font)
self.btn2.setObjectName("btn2")
self.verticalLayout_9.addWidget(self.btn2)
self.label_3 = QtWidgets.QLabel(self.centralwidget)
self.label_3.setText("")
self.label_3.setPixmap(QtGui.QPixmap("dream1.png"))
self.label_3.setObjectName("label_3")
self.verticalLayout_9.addWidget(self.label_3)
self.horizontalLayout_3.addLayout(self.verticalLayout_9)
spacerItem5 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
self.horizontalLayout_3.addItem(spacerItem5)
self.verticalLayout.addLayout(self.horizontalLayout_3)
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 900, 21))
self.menubar.setObjectName("menubar")
self.menuFile = QtWidgets.QMenu(self.menubar)
self.menuFile.setObjectName("menuFile")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)
self.actionNew = QtWidgets.QAction(MainWindow)
self.actionNew.setObjectName("actionNew")
```



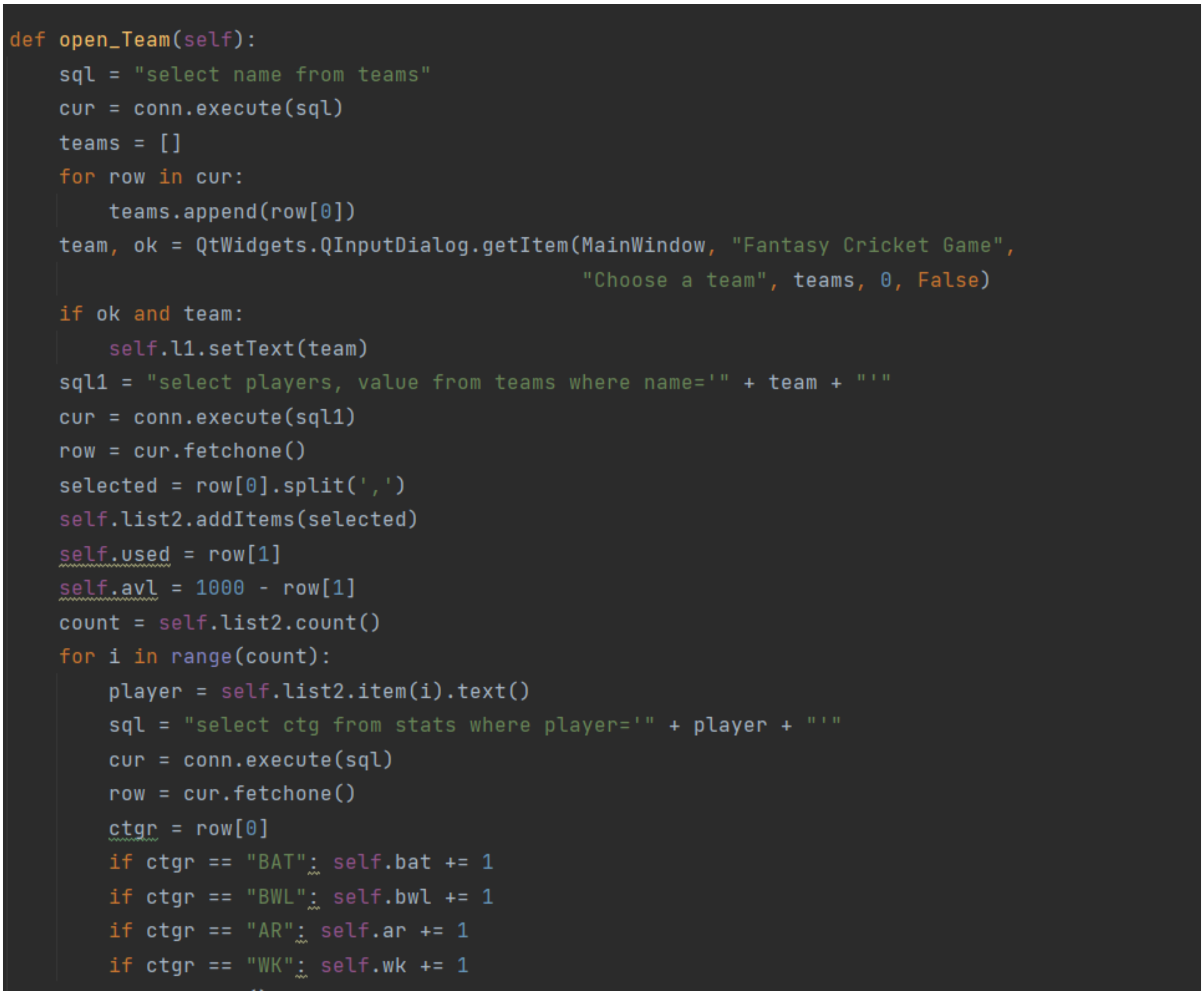
```python
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Fantasy Cricket Game"))
    self.label_4.setText(_translate("MainWindow", "Batsmen"))
    self.label_6.setText(_translate("MainWindow", "Bowlers"))
    self.label_7.setText(_translate("MainWindow", "All Rounders"))
    self.label_8.setText(_translate("MainWindow", "Wicketkeepers"))
    self.label.setText(_translate("MainWindow", "Player Categories"))
    self.rb1.setText(_translate("MainWindow", "BAT"))
    self.rb2.setText(_translate("MainWindow", "BOWL"))
    self.rb3.setText(_translate("MainWindow", "AR"))
    self.rb4.setText(_translate("MainWindow", "WK"))
    self.btn1.setText(_translate("MainWindow", "Available Points : 1000"))
    self.label_2.setText(_translate("MainWindow", "Selected Players"))
    self.l1.setText(_translate("MainWindow", "Team_name"))
    self.btn2.setText(_translate("MainWindow", "Points used : "))
    self.menuFile.setTitle(_translate("MainWindow", "File"))
    self.actionNew.setText(_translate("MainWindow", "NEW Team"))
    self.actionOpen.setText(_translate("MainWindow", "OPEN Team"))
    self.actionSave_Team.setText(_translate("MainWindow", "SAVE Team"))
    self.actionQuit.setText(_translate("MainWindow", "EVALUATE Team"))
```



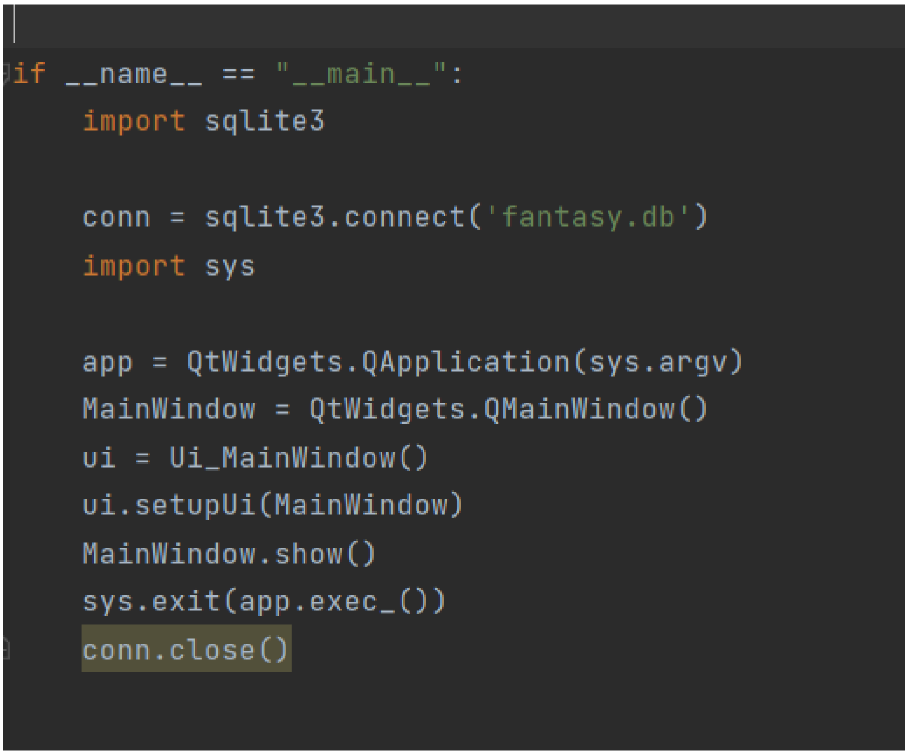
```python
def menufunction(self, action):
    txt = action.text()
    if txt == "NEW Team":
        self.bat = 0
        self.bwl = 0
        self.ar = 0
        self.wk = 0
        self.avl = 1000
        self.used = 0
        self.list1.clear()
        self.list2.clear()
        self.l1.setText("Team_name")
        self.showstatus()
        text, ok = QtWidgets.QInputDialog.getText(MainWindow, 'Fantasy Cricket Game', 'Enter name of team:')
        if ok:
            self.l1.setText(str(text))
    if txt == 'SAVE Team':
        selected = ""
        count = self.list2.count()
        for i in range(count):
            selected = selected + self.list2.item(i).text()
            if i < count - 1:
                selected = selected + ","
        self.save_Team(self.l1.text(), selected, self.used)
    if txt == "OPEN Team":
        self.bat = 0
        self.bwl = 0
        self.ar = 0
        self.wk = 0
        self.avl = 1000
```



```
def save_Team(self, nm, string, val):
    if self.bat + self.bwl + self.ar + self.wk != 11:
        self.showdlg("Insufficient players")
        return
    sql = "INSERT INTO teams (name, players, value) VALUES ('" + nm + "','" + string + "','" + str(val) + "');"
    try:
        cur = conn.execute(sql)
        conn.commit()
        self.showdlg("Team Saved successfully")
    except:
        self.showdlg("error in operation")
        conn.rollback()
```

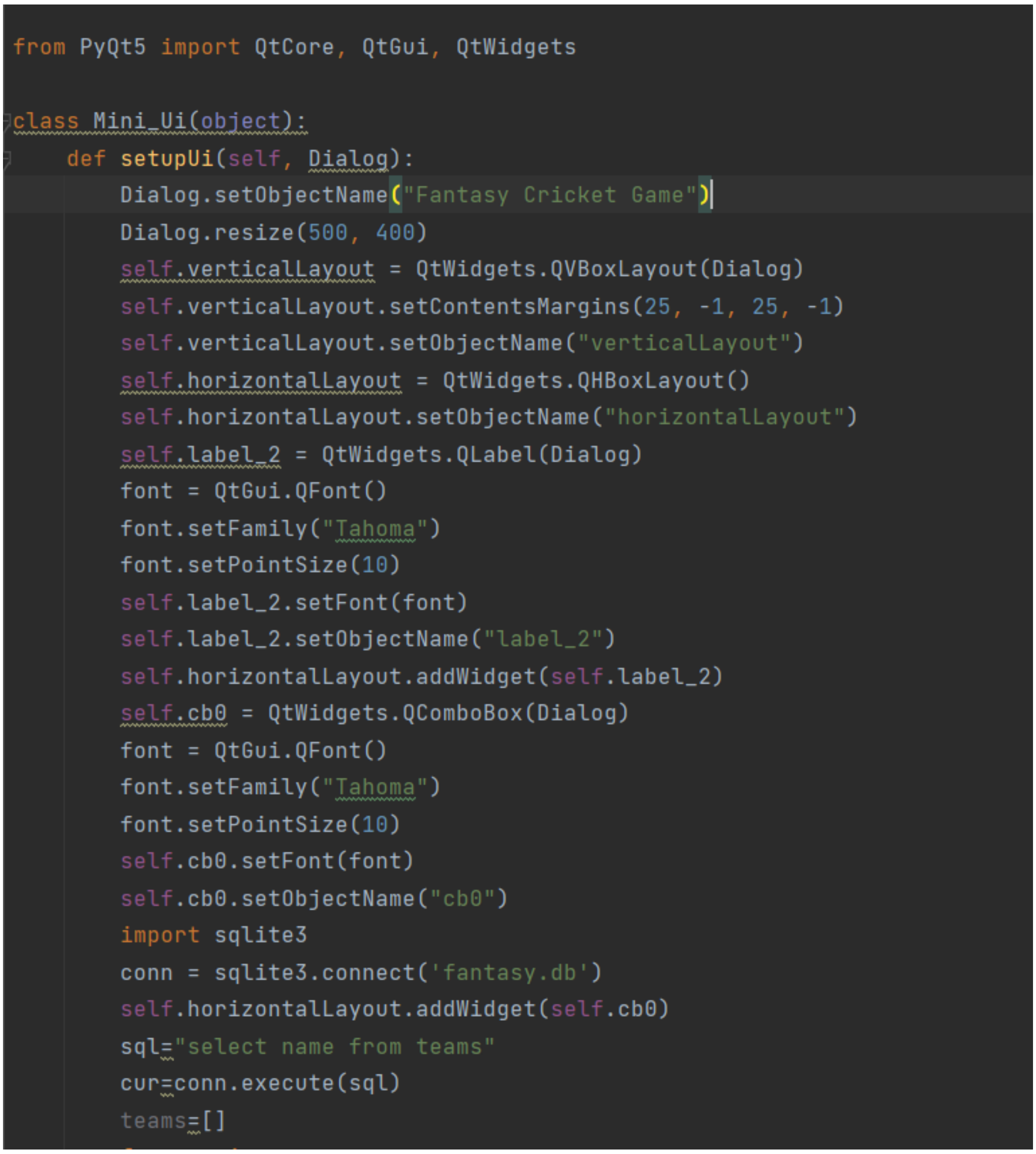


```
def open_Team(self):
    sql = "select name from teams"
    cur = conn.execute(sql)
    teams = []
    for row in cur:
        teams.append(row[0])
    team, ok = QtWidgets.QInputDialog.getItem(MainWindow, "Fantasy Cricket Game",
                                              "Choose a team", teams, 0, False)
    if ok and team:
        self.l1.setText(team)
    sql1 = "select players, value from teams where name='" + team + "'"
    cur = conn.execute(sql1)
    row = cur.fetchone()
    selected = row[0].split(',')
    self.list2.addItems(selected)
    self.used = row[1]
    self.avl = 1000 - row[1]
    count = self.list2.count()
    for i in range(count):
        player = self.list2.item(i).text()
        sql = "select ctg from stats where player='" + player + "'"
        cur = conn.execute(sql)
        row = cur.fetchone()
        ctgr = row[0]
        if ctgr == "BAT": self.bat += 1
        if ctgr == "BWL": self.bwl += 1
        if ctgr == "AR": self.ar += 1
        if ctgr == "WK": self.wk += 1
```

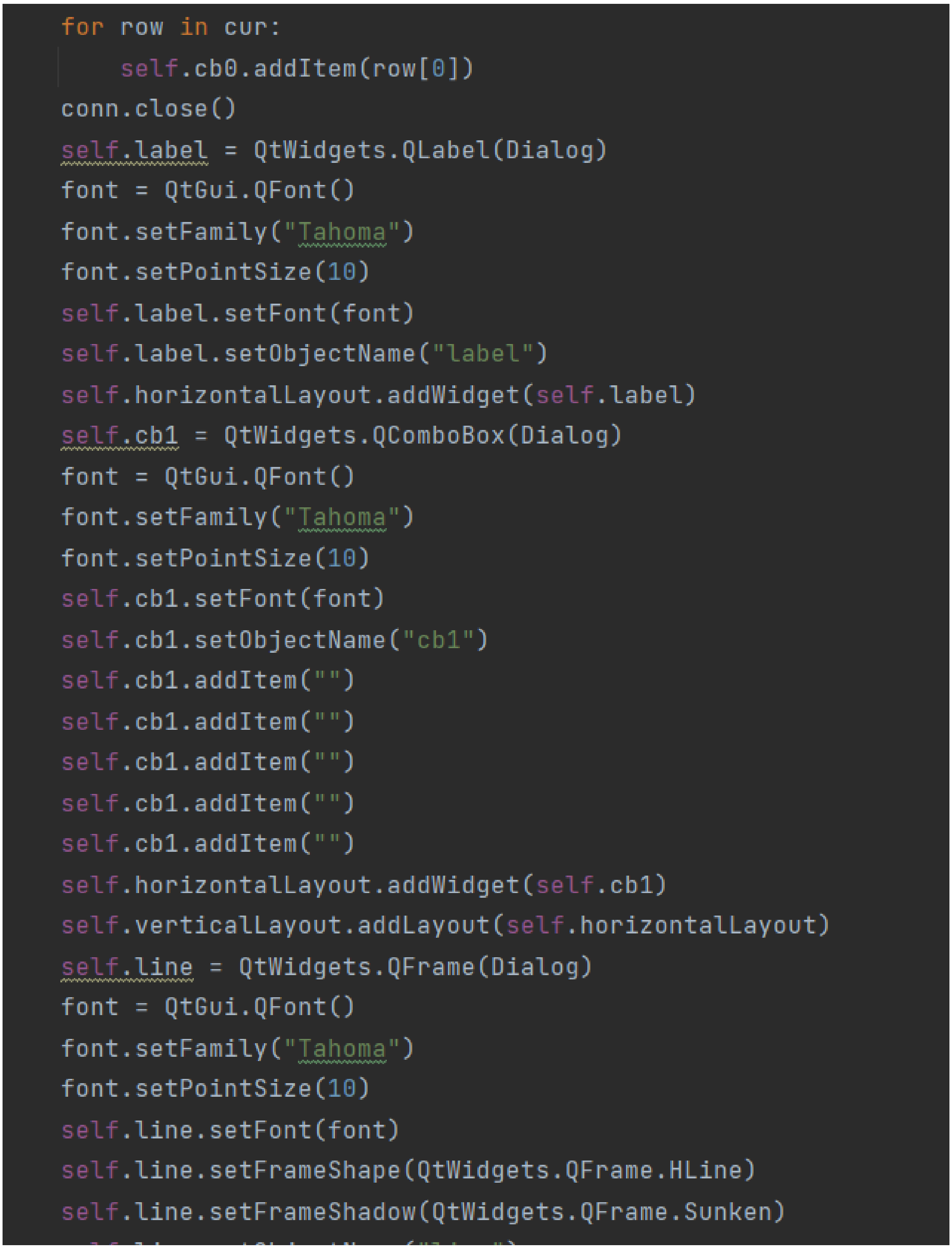


```python
if __name__ == "__main__":
    import sqlite3

    conn = sqlite3.connect('fantasy.db')
    import sys

    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
    conn.close()
```

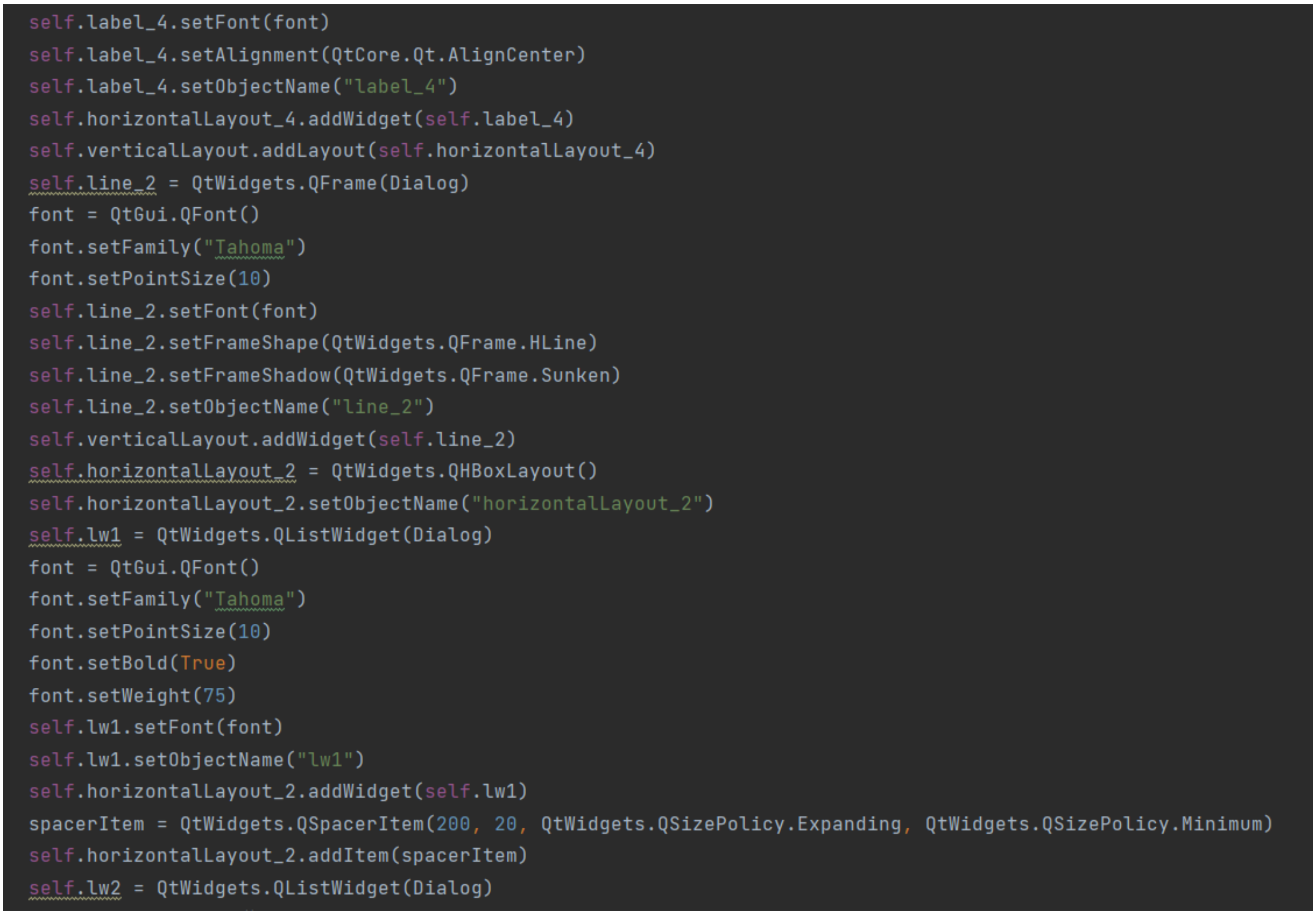
evaluateTeam.py



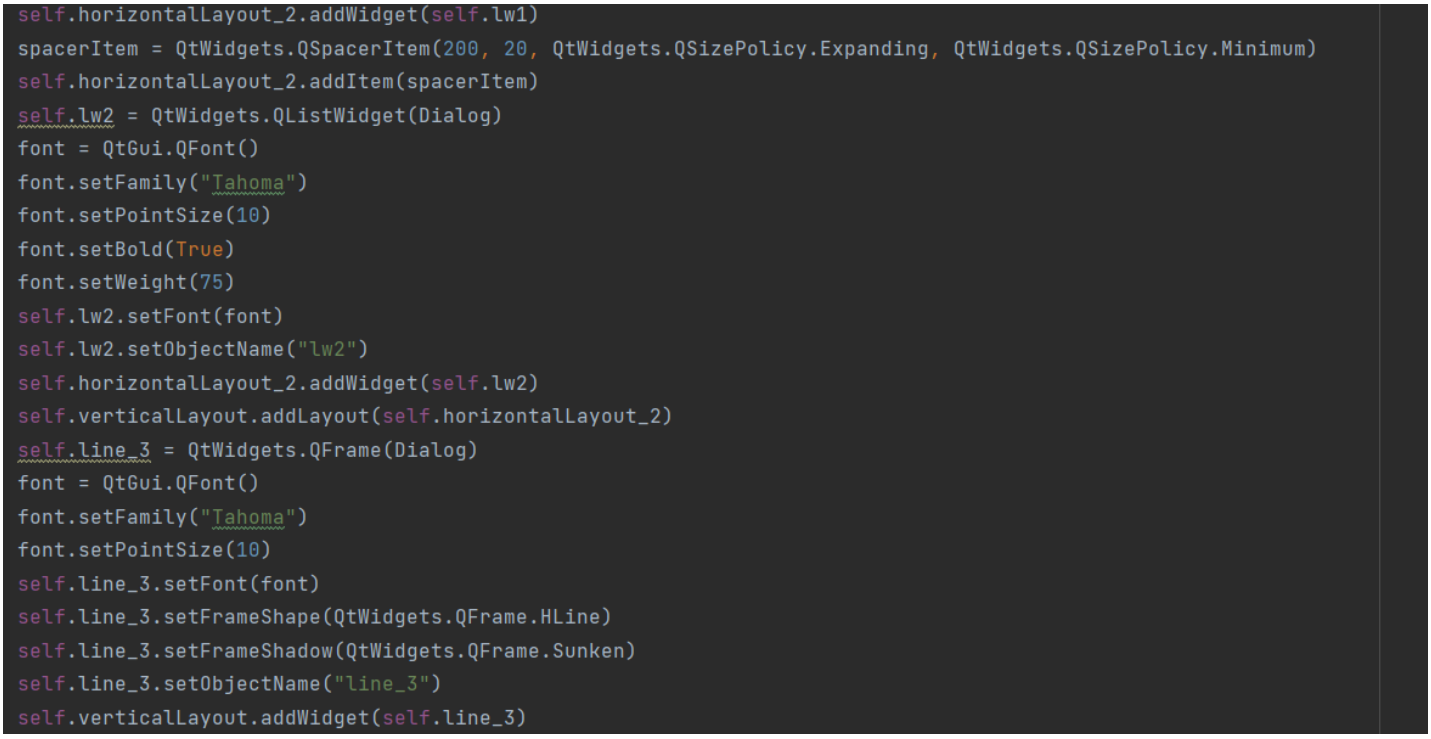
```python
from PyQt5 import QtCore, QtGui, QtWidgets

class Mini_Ui(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Fantasy Cricket Game")
        Dialog.resize(500, 400)
        self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)
        self.verticalLayout.setContentsMargins(25, -1, 25, -1)
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.label_2 = QtWidgets.QLabel(Dialog)
        font = QtGui.QFont()
        font.setFamily("Tahoma")
        font.setPointSize(10)
        self.label_2.setFont(font)
        self.label_2.setObjectName("label_2")
        self.horizontalLayout.addWidget(self.label_2)
        self.cb0 = QtWidgets.QComboBox(Dialog)
        font = QtGui.QFont()
        font.setFamily("Tahoma")
        font.setPointSize(10)
        self.cb0.setFont(font)
        self.cb0.setObjectName("cb0")
        import sqlite3
        conn = sqlite3.connect('fantasy.db')
        self.horizontalLayout.addWidget(self.cb0)
        sql="select name from teams"
        cur=conn.execute(sql)
        teams=[]
```



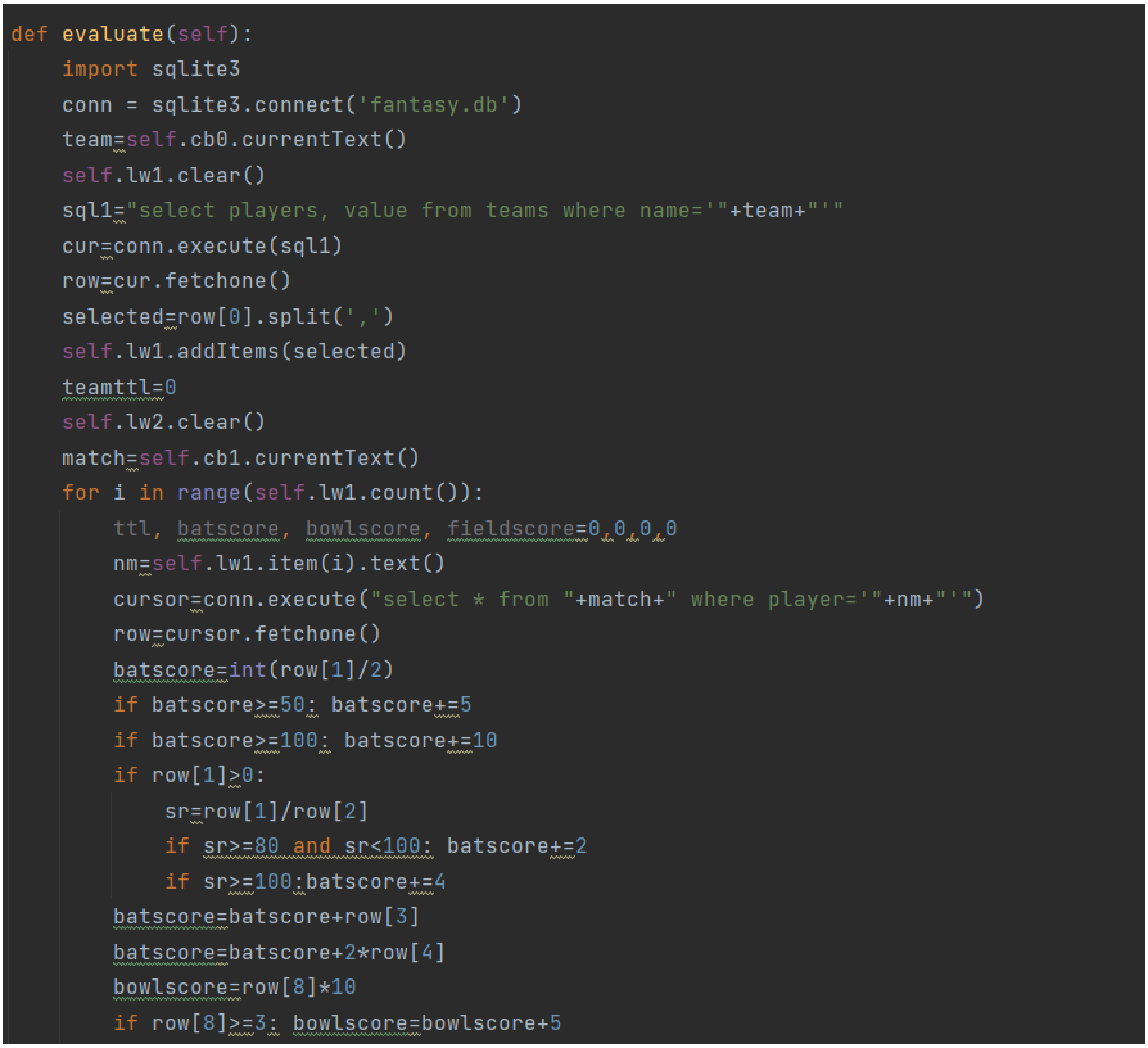
```
for row in cur:
    self.cb0.addItem(row[0])
conn.close()
self.label = QtWidgets.QLabel(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
self.label.setFont(font)
self.label.setObjectName("label")
self.horizontalLayout.addWidget(self.label)
self.cb1 = QtWidgets.QComboBox(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
self.cb1.setFont(font)
self.cb1.setObjectName("cb1")
self.cb1.addItem("")
self.cb1.addItem("")
self.cb1.addItem("")
self.cb1.addItem("")
self.cb1.addItem("")
self.horizontalLayout.addWidget(self.cb1)
self.verticalLayout.addLayout(self.horizontalLayout)
self.line = QtWidgets.QFrame(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
self.line.setFont(font)
self.line.setFrameShape(QtWidgets.QFrame.HLine)
self.line.setFrameShadow(QtWidgets.QFrame.Sunken)
```



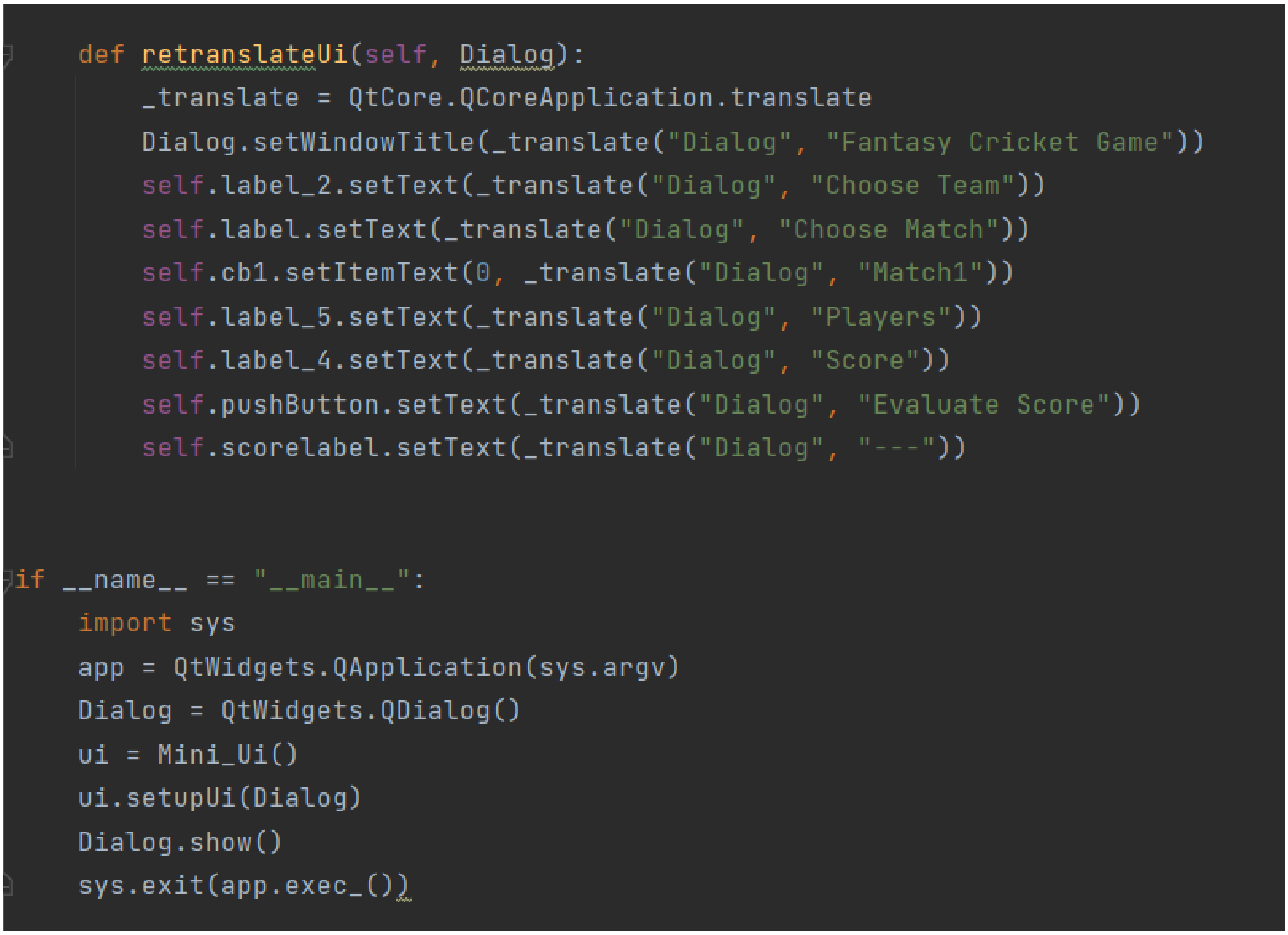
```
self.label_4.setFont(font)
self.label_4.setAlignment(QtCore.Qt.AlignCenter)
self.label_4.setObjectName("label_4")
self.horizontalLayout_4.addWidget(self.label_4)
self.verticalLayout.addLayout(self.horizontalLayout_4)
self.line_2 = QtWidgets.QFrame(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
self.line_2.setFont(font)
self.line_2.setFrameShape(QtWidgets.QFrame.HLine)
self.line_2.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_2.setObjectName("line_2")
self.verticalLayout.addWidget(self.line_2)
self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
self.horizontalLayout_2.setObjectName("horizontalLayout_2")
self.lw1 = QtWidgets.QListWidget(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
font.setBold(True)
font.setWeight(75)
self.lw1.setFont(font)
self.lw1.setObjectName("lw1")
self.horizontalLayout_2.addWidget(self.lw1)
spacerItem = QtWidgets.QSpacerItem(200, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
self.horizontalLayout_2.addItem(spacerItem)
self.lw2 = QtWidgets.QListWidget(Dialog)
```



```
self.horizontalLayout_2.addWidget(self.lw1)
spacerItem = QtWidgets.QSpacerItem(200, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
self.horizontalLayout_2.addItem(spacerItem)
self.lw2 = QtWidgets.QListWidget(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
font.setBold(True)
font.setWeight(75)
self.lw2.setFont(font)
self.lw2.setObjectName("lw2")
self.horizontalLayout_2.addWidget(self.lw2)
self.verticalLayout.addLayout(self.horizontalLayout_2)
self.line_3 = QtWidgets.QFrame(Dialog)
font = QtGui.QFont()
font.setFamily("Tahoma")
font.setPointSize(10)
self.line_3.setFont(font)
self.line_3.setFrameShape(QtWidgets.QFrame.HLine)
self.line_3.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_3.setObjectName("line_3")
self.verticalLayout.addWidget(self.line_3)
```



```
def evaluate(self):
    import sqlite3
    conn = sqlite3.connect('fantasy.db')
    team=self.cb0.currentText()
    self.lw1.clear()
    sql1="select players, value from teams where name='"+team+"'"
    cur=conn.execute(sql1)
    row=cur.fetchone()
    selected=row[0].split(',')
    self.lw1.addItems(selected)
    teamttl=0
    self.lw2.clear()
    match=self.cb1.currentText()
    for i in range(self.lw1.count()):
        ttl, batscore, bowlscore, fieldscore=0,0,0,0
        nm=self.lw1.item(i).text()
        cursor=conn.execute("select * from "+match+" where player='"+nm+"'")
        row=cursor.fetchone()
        batscore=int(row[1]/2)
        if batscore>=50: batscore+=5
        if batscore>=100: batscore+=10
        if row[1]>0:
            sr=row[1]/row[2]
            if sr>=80 and sr<100: batscore+=2
            if sr>=100:batscore+=4
        batscore=batscore+row[3]
        batscore=batscore+2*row[4]
        bowlscore=row[8]*10
        if row[8]>=3: bowlscore=bowlscore+5
```



```
    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Fantasy Cricket Game"))
        self.label_2.setText(_translate("Dialog", "Choose Team"))
        self.label.setText(_translate("Dialog", "Choose Match"))
        self.cb1.setItemText(0, _translate("Dialog", "Match1"))
        self.label_5.setText(_translate("Dialog", "Players"))
        self.label_4.setText(_translate("Dialog", "Score"))
        self.pushButton.setText(_translate("Dialog", "Evaluate Score"))
        self.scorelabel.setText(_translate("Dialog", "---"))


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Dialog = QtWidgets.QDialog()
    ui = Mini_Ui()
    ui.setupUi(Dialog)
    Dialog.show()
    sys.exit(app.exec_())
```

Conclusion

- I believe the trial has shown conclusively that it is both possible and desirable to use Python as the principal teaching language.

- It is Free (as in both cost and source code).

- It is trivial to install on a Windows PC allowing students to take their interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated.

- It is a flexible tool that allows both the teaching of traditional procedural programming and modern OOP; It can be used to teach a large number of transferable skills.

- It appears to be quicker to learn and, in combination with its many libraries, this offers the possibility of more rapid student development allowing the course to be made more challenging and varied.

- Most importantly, its clean syntax offers increased understanding and enjoyment for students.

BIBLIOGRAPHY

https://trainings.internshala.com/python-training

https://www.w3schools.com/python/

https://wiki.python.org/moin/PyQt/Tutorials

https://www.tutorialspoint.com/pyqt/

https://www.tutorialspoint.com/sqlite/sqlite_quick_guide.html