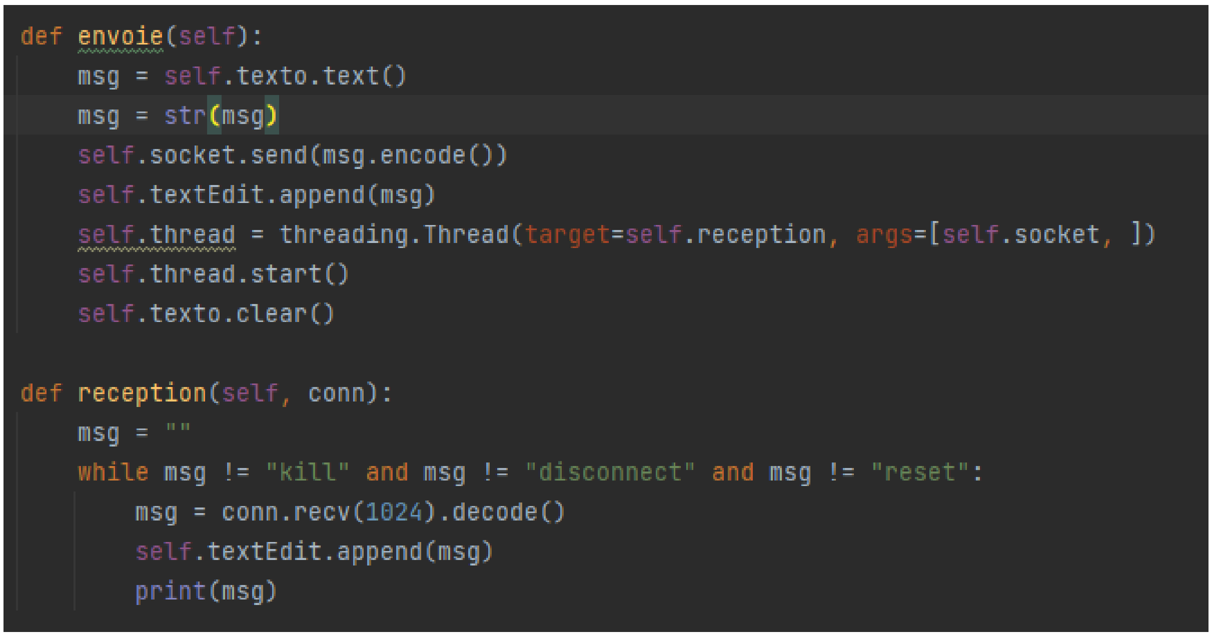
SAE302 Dev manual

In this manual, I will explain in more detail the methods that I have used.

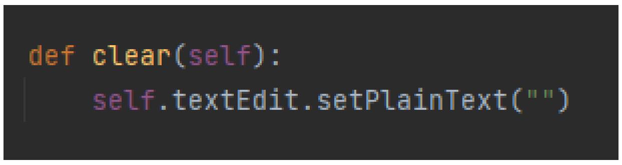


```python
def envoie(self):
    msg = self.texto.text()
    msg = str(msg)
    self.socket.send(msg.encode())
    self.textEdit.append(msg)
    self.thread = threading.Thread(target=self.reception, args=[self.socket, ])
    self.thread.start()
    self.texto.clear()

def reception(self, conn):
    msg = ""
    while msg != "kill" and msg != "disconnect" and msg != "reset":
        msg = conn.recv(1024).decode()
        self.textEdit.append(msg)
        print(msg)
```

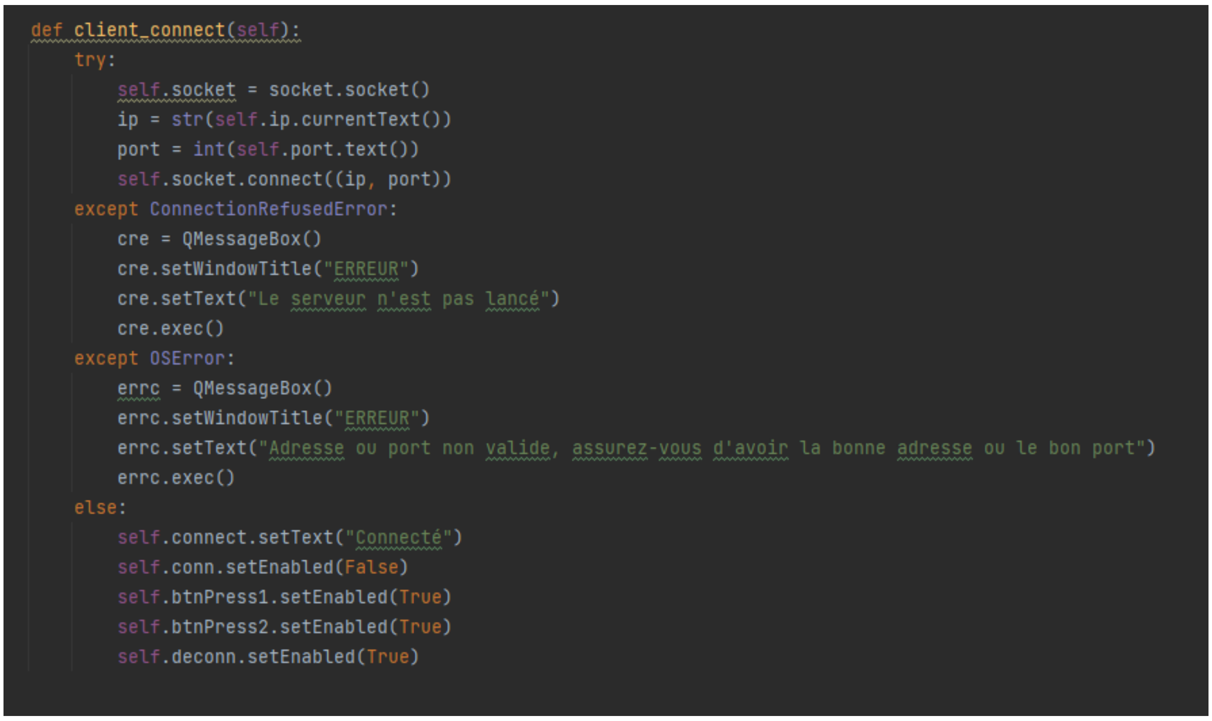
To send: I convert my message to a string and then send it to the server. In addition, I add a thread that makes each message independent. I do not need to wait for the message to be sent before the other one is also sent. I also clear the command chat.

To receive: The command arrives from the server and therefore I send it to the command logs.



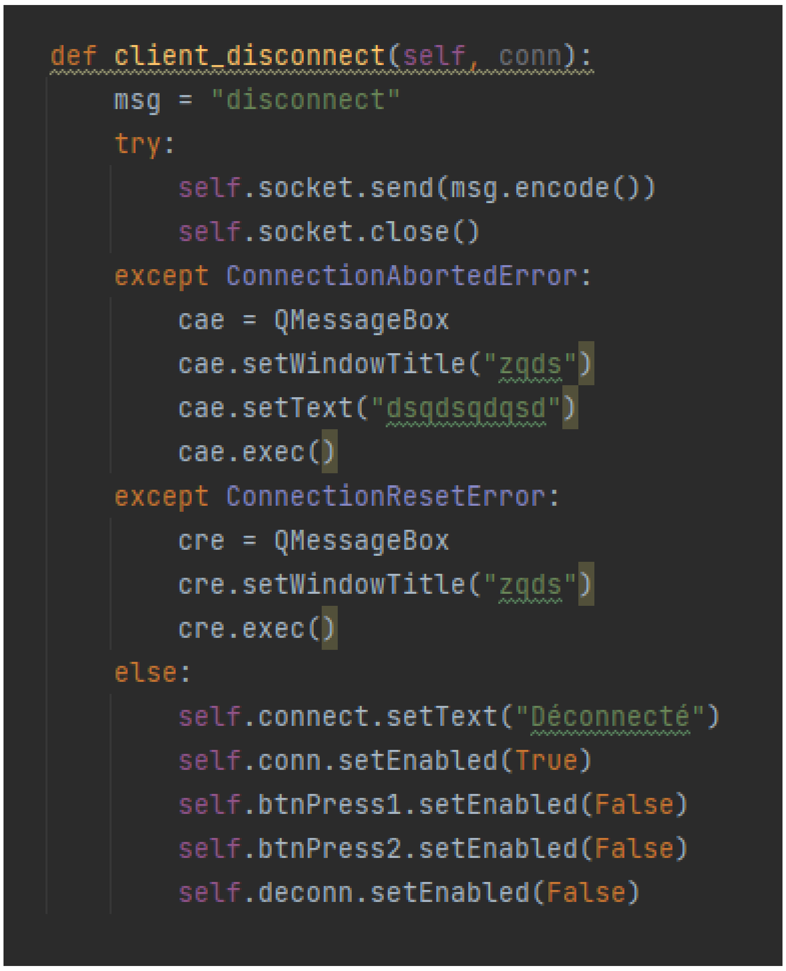
```python
def clear(self):
    self.textEdit.setPlainText("")
```

The clear is very simple, I add nothing to simulate a clear.



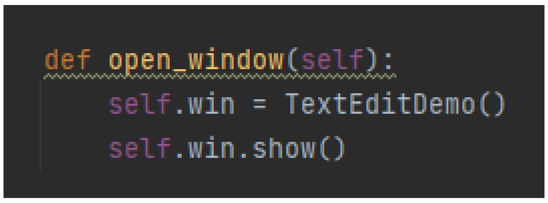
```python
def client_connect(self):
    try:
        self.socket = socket.socket()
        ip = str(self.ip.currentText())
        port = int(self.port.text())
        self.socket.connect((ip, port))
    except ConnectionRefusedError:
        cre = QMessageBox()
        cre.setWindowTitle("ERREUR")
        cre.setText("Le serveur n'est pas lancé")
        cre.exec()
    except OSError:
        errc = QMessageBox()
        errc.setWindowTitle("ERREUR")
        errc.setText("Adresse ou port non valide, assurez-vous d'avoir la bonne adresse ou le bon port")
        errc.exec()
    else:
        self.connect.setText("Connecté")
        self.conn.setEnabled(False)
        self.btnPress1.setEnabled(True)
        self.btnPress2.setEnabled(True)
        self.deconn.setEnabled(True)
```

Client connection:
I open the socket and connect with the correct IP (string because there are ".") and correct port (int because it is a number). This will help fix input bugs later. I use exceptions that prevent the GUI from crashing. ConnectionRefusedError shows that the server is not running. OSError indicates that the IPs are invalid.



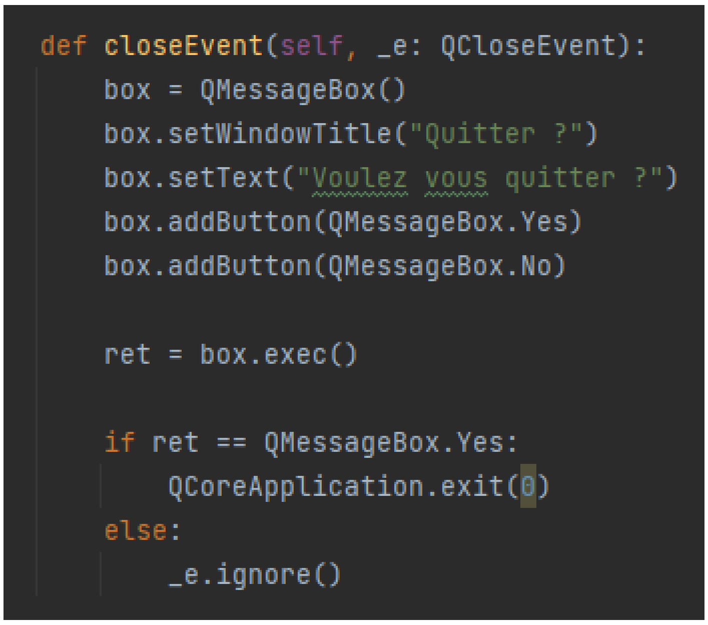
```python
def client_disconnect(self, conn):
    msg = "disconnect"
    try:
        self.socket.send(msg.encode())
        self.socket.close()
    except ConnectionAbortedError:
        cae = QMessageBox
        cae.setWindowTitle("zqds")
        cae.setText("dsqdsqdqsd")
        cae.exec()
    except ConnectionResetError:
        cre = QMessageBox
        cre.setWindowTitle("zqds")
        cre.exec()
    else:
        self.connect.setText("Déconnecté")
        self.conn.setEnabled(True)
        self.btnPress1.setEnabled(False)
        self.btnPress2.setEnabled(False)
        self.deconn.setEnabled(False)
```

Client disconnection:
I just send a "disconnect" message to the command logs and close the session. I also use exceptions so that it doesn't crash but it still does.



```python
def open_window(self):
    self.win = TextEditDemo()
    self.win.show()
```

Very simple as well, I just add a new interface and display it.

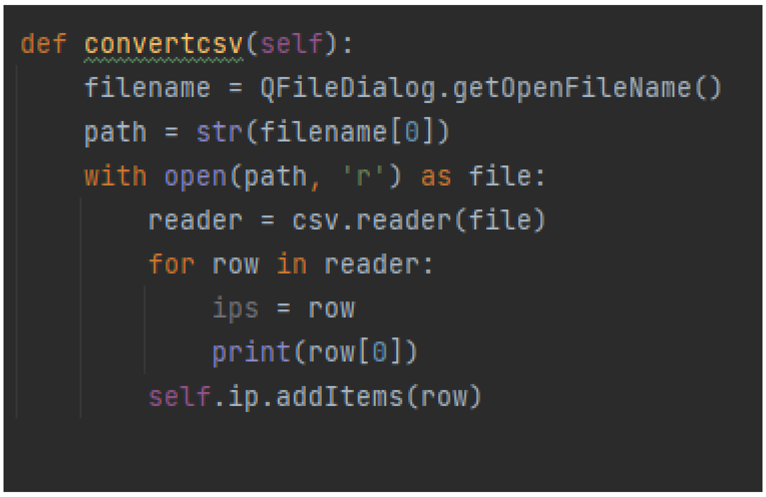


```
def closeEvent(self, _e: QCloseEvent):
    box = QMessageBox()
    box.setWindowTitle("Quitter ?")
    box.setText("Voulez vous quitter ?")
    box.addButton(QMessageBox.Yes)
    box.addButton(QMessageBox.No)

    ret = box.exec()

    if ret == QMessageBox.Yes:
        QCoreApplication.exit(0)
    else:
        _e.ignore()
```

closeEvent:
Sends a dialog box if the interface is closed for confirmation and closes the session.



```
def convertcsv(self):
    filename = QFileDialog.getOpenFileName()
    path = str(filename[0])
    with open(path, 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            ips = row
            print(row[0])
        self.ip.addItems(row)
```

CSV conversion:
This button opens a folder, after choosing the file, it will read it and for each iteration (each ',') it will separate the IPs and put them in the QCombobox. The file can be modified at any time.