1. pielikums

QGIS Python makross

```
from PyQt5.QtWidgets import QWidget, QToolButton, QToolBar, QMessageBox
from PyQt5.QtGui import QPixmap, QIcon
from PyQt5.QtCore import QSize, QTimer
from qgis.utils import iface
import socket
import threading
from qgis.core import QgsMessageLog, QgsProject
import subprocess

def openProject():

    proj = QgsProject.instance()

    toolbar=iface.mainWindow().addToolBar('roninToolBar')
    toolbar.setIconSize(QSize(180, 35));

    socketClient = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    socketServer = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    #socketServer.bind(('127.0.0.1', 8887))
    socketServer.bind(('192.168.4.144', 8887))
    udpDestinationSocket = ('192.168.4.130', 8886)

    buttonCarrot = QToolButton()
    buttonCarrot.setToolTip('<div style="background-color: #FEF9E7;"><b>Atpazīt burkānus</b></div>')

    buttonBeet = QToolButton()
    buttonBeet.setToolTip('<div style="background-color: #FEF9E7;"><b>Atpazīt bietes</b></div>')

    buttonPumpkin = QToolButton()
    buttonPumpkin.setToolTip('<div style="background-color: #FEF9E7;"><b>Atpazīt ķirbjus</b><br>Icon made by Freepik from www.flaticon.com</div>')

    buttonZucckini = QToolButton()
    buttonZucckini.setToolTip('<div style="background-color: #FEF9E7;"><b>Atpazīt cukīni</b><br>Icon made by Freepik from www.flaticon.com</div>')

    buttonRadish = QToolButton()
    buttonRadish.setToolTip('<div style="background-color: #FEF9E7;"><b>Atpazīt redīsus</b><br>Icon made by Freepik from www.flaticon.com</div>')

    buttonBlackRadish = QToolButton()
    buttonBlackRadish.setToolTip('<div style="background-color: #FEF9E7;"><b>Atpazīt rutkus</b><br>Icon made by Nikita Golubev from www.flaticon.com</div>')

    buttonDrill = QToolButton()
    buttonDrill.setToolTip('<div style="background-color: #FEF9E7;"><b>Izvēlēties mehānisko ierobežotāju</b><br>Icon made by Freepik from www.flaticon.com</div>')

    buttonLaser = QToolButton()
    buttonLaser.setToolTip('<div style="background-color: #FEF9E7;"><b>Izvēlēties lāzierierobežotāju</b><br>Icon made by Freepik from www.flaticon.com</div>')

    buttonStartObjectDetection = QToolButton()
    buttonStartObjectDetection.setToolTip('<div style="background-color: #FEF9E7;"><b>Veikt augu atpazīšanu</b><br>Icon made by Freepik from www.flaticon.com</div>')

    buttonStartWeeding = QToolButton()
    buttonStartWeeding.setToolTip('<div style="background-color: #FEF9E7;"><b>Veikt nezāļu ierobežošanu</b><br>Icon made by photo3idea_studio from www.flaticon.com</div>')
```

```
        buttonStartObjectDetectionAndWeeding = QToolButton()
        buttonStartObjectDetectionAndWeeding.setToolTip('<div style="background-color: #FEF9E7;"><b>Veikt augu atpazīšanu un nezāļu ierobežošanu</b><br>Icon made by Freepik and photo3idea_studio from www.flaticon.com</div>')

        buttonAddGpsPointToRoute = QToolButton()
        buttonAddGpsPointToRoute.setToolTip('<div style="background-color: #FEF9E7;"><b>Pievienot GPS punktu maršrutam</b><br>Icon made by Freepik from www.flaticon.com</div>')

        buttonResetGpsRoute = QToolButton()
        buttonResetGpsRoute.setToolTip('<div style="background-color: #FEF9E7;"><b>Izdzēst GPS maršrutu</b><br>Icon made by Freepik from www.flaticon.com</div>')

        buttonAddGpsPointToRidge = QToolButton()
        buttonAddGpsPointToRidge.setToolTip('<div style="background-color: #FEF9E7;"><b>Pievienot GPS punktu vagai</b><br>Icon made by Freepik from www.flaticon.com</div>')

        buttonResetGpsRidge = QToolButton()
        buttonResetGpsRidge.setToolTip('<div style="background-color: #FEF9E7;"><b>Izdzēst GPS vagas maršrutu</b><br>Icon made by Freepik from www.flaticon.com</div>')

        buttonStartGpsRoute = QToolButton()
        buttonStartGpsRoute.setToolTip('<div style="background-color: #FEF9E7;"><b>Braukt pēc GPS</b><br>Icon made by Freepik from www.flaticon.com</div>')

        buttonStartGpsRouteAndFollowRow = QToolButton()
        buttonStartGpsRouteAndFollowRow.setToolTip('<div style="background-color: #FEF9E7;"><b>Braukt pēc GPS un sekot vagai</b><br>Icon made by Freepik and Flat Icons from www.flaticon.com</div>')

        buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding = QToolButton()
        buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setToolTip('<div style="background-color: #FEF9E7;"><b>Braukt pēc GPS, sekot vagai, atpazīt augus un ierobežot nezāles</b><br>Icon made by Freepik, Flat Icons and photo3idea_studio from www.flaticon.com</div>')

        buttonEmergencyStop = QToolButton()
        buttonEmergencyStop.setToolTip('<div style="background-color: #FEF9E7;"><b>Avārijas apstāšanās</b></div>')

        buttonStartWeeding.setEnabled(False)
        buttonStartObjectDetectionAndWeeding.setEnabled(False)
        buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setEnabled(False)

        def detectCarrots():
            proj.writeEntry('RONIN', 'detectionMode', 'burkans')
            buttonCarrot.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')
            buttonBeet.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonPumpkin.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonZucckini.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonBlackRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            #socketClient.sendto(b'burkans', ('127.0.0.1', 8888))

        def detectBeets():
            proj.writeEntry('RONIN', 'detectionMode', 'biete')
            buttonCarrot.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonBeet.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')
            buttonPumpkin.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonZucckini.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonBlackRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            #socketClient.sendto(detectionMode.encode(), ('127.0.0.1', 8888))

        def detectPumpkins():
            proj.writeEntry('RONIN', 'detectionMode', 'kirbis')
            buttonCarrot.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonBeet.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonPumpkin.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')
            buttonZucckini.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            buttonBlackRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
            #socketClient.sendto(b'kirbis', ('127.0.0.1', 8888))
```

```
def detectZucchini():
    proj.writeEntry('RONIN', 'detectionMode', 'cukini')
    buttonCarrot.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonBeet.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonPumpkin.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonZucckini.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')
    buttonRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonBlackRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    #socketClient.sendto(b'cukini', ('127.0.0.1', 8888))

def detectRadishes():
    proj.writeEntry('RONIN', 'detectionMode', 'rediss')
    buttonCarrot.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonBeet.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonPumpkin.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonZucckini.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonRadish.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')
    buttonBlackRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    #socketClient.sendto(b'rediss', ('127.0.0.1', 8888))

def detectBlackRadishes():
    proj.writeEntry('RONIN', 'detectionMode', 'rutks')
    buttonCarrot.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonBeet.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonPumpkin.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonZucckini.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonRadish.setStyleSheet('background-color: #F2F4F4; width: 24px;')
    buttonBlackRadish.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')
    #socketClient.sendto(b'rutks', ('127.0.0.1', 8888))

def selectDrill():
    drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)

    if drillSelected:
        proj.writeEntry('RONIN', 'drillSelected', False)
        buttonDrill.setStyleSheet('background-color: #F2F4F4; width: 24px;')

        laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)
        if not laserSelected:
            buttonStartWeeding.setEnabled(False)
            buttonStartObjectDetectionAndWeeding.setEnabled(False)
            buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setEnabled(False)
    else:
        proj.writeEntry('RONIN', 'drillSelected', True)
        buttonDrill.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')

        buttonStartWeeding.setEnabled(True)
        buttonStartObjectDetectionAndWeeding.setEnabled(True)
        buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setEnabled(True)

def selectLaser():
    laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)

    if laserSelected:
        proj.writeEntry('RONIN', 'laserSelected', False)
        buttonLaser.setStyleSheet('background-color: #F2F4F4; width: 24px;')

        drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)
        if not drillSelected:
            buttonStartWeeding.setEnabled(False)
            buttonStartObjectDetectionAndWeeding.setEnabled(False)
            buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setEnabled(False)
    else:
        proj.writeEntry('RONIN', 'laserSelected', True)
        buttonLaser.setStyleSheet('background-color: #D5DBDB; border: 1px solid #A6ACAF; border-radius: 3px; width: 27px;')

        buttonStartWeeding.setEnabled(True)
        buttonStartObjectDetectionAndWeeding.setEnabled(True)
        buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setEnabled(True)
```

```
        def startObjectDetection():
            laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)
            drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)
            detectionMode, type_conversion_ok = proj.readEntry('RONIN', 'detectionMode', 'burkans')

            socketClient.sendto((f'atpazinejs,{detectionMode},laserSelected:{laserSelected},drillSelected:{drillSelected}').encode(),
udpDestinationSocket)
            #socketClient.sendto(('atpazinejs,%s' % detectionMode).encode(), udpDestinationSocket)

        def startWeeding():
            laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)
            drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)

            socketClient.sendto((f'ravetajs,laserSelected:{laserSelected},drillSelected:{drillSelected}').encode(), udpDestinationSocket)

        def startObjectDetectionAndWeeding():
            detectionMode, type_conversion_ok = proj.readEntry('RONIN', 'detectionMode', 'burkans')
            laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)
            drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)

            socketClient.sendto((f'atpazinejs,{detectionMode},ravetajs,laserSelected:{laserSelected},drillSelected:{drillSelected}').encode(),
udpDestinationSocket)

        def startGpsRoute():
            socketClient.sendto(b'gps_marsruts', udpDestinationSocket)

        def startGpsRouteAndFollowRow():
            socketClient.sendto(b'gps_marsruts,vaga', udpDestinationSocket)

        def startGpsRouteAndFollowRowAndObjectDetectionAndWeeding():
            detectionMode, type_conversion_ok = proj.readEntry('RONIN', 'detectionMode', 'burkans')
            laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)
            drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)

socketClient.sendto((f'gps_marsruts,vaga,atpazinejs,{detectionMode},ravetajs,laserSelected:{laserSelected},drillSelected:{drillSelected}').encod
e(), udpDestinationSocket)

        def emergencyStop():
            socketClient.sendto(b'stop', udpDestinationSocket)

        def openResult(commandArgument):
            process = subprocess.Popen(f'python3 openResults.py {commandArgument}', shell=True)

        def addGpsPointToRoute():
            socketClient.sendto(b'pievienot_gps_marsruta_punktu', udpDestinationSocket)

        def resetGpsRoute():
            socketClient.sendto(b'izdzest_gps_marsrutu', udpDestinationSocket)

        def addGpsPointToRidge():
            socketClient.sendto(b'pievienot_gps_vagas_punktu', udpDestinationSocket)

        def resetGpsRidge():
            socketClient.sendto(b'izdzest_gps_vagu', udpDestinationSocket)

        detectionMode, type_conversion_ok = proj.readEntry('RONIN', 'detectionMode', 'burkans')
        if type_conversion_ok:
            if detectionMode == 'burkans':
                detectCarrots()
            elif detectionMode == 'biete':
                detectBeets()
            elif detectionMode == 'kirbis':
                detectPumpkins()
            elif detectionMode == 'cukini':
                detectZucchini()
            elif detectionMode == 'rediss':
                detectRadishes()
            elif detectionMode == 'rutks':
                detectBlackRadishes()
```

```
laserSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'laserSelected', False)
if type_conversion_ok:
    if laserSelected:
        selectLaser()

        timer = QTimer()
        timer.setSingleShot(True)
        timer.singleShot(500, selectLaser)

drillSelected, type_conversion_ok = proj.readBoolEntry('RONIN', 'drillSelected', False)
if type_conversion_ok:
    if drillSelected:
        selectDrill()

        timer = QTimer()
        timer.setSingleShot(True)
        timer.singleShot(500, selectDrill)

class udpServerThread (threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        self.stop_event = threading.Event()
    def run(self):
        while not self.stop_event.is_set():
            QgsMessageLog.logMessage('gaida UDP ziņojumu', 'RONIN', 0)
            data, addr = socketServer.recvfrom(1024) # buffer size is 1024 bytes

            message = data.decode('utf-8')

            if message == 'komanda sanemta':
                QgsMessageLog.logMessage(f'ienākošais UDP ziņojums: {message}', 'RONIN', 0)
                #QMessageBox.information(None, 'Informācija', 'Komanda nosūtīta veiksmīgi')
            elif message == 'atpazisana pabeigta':
                QgsMessageLog.logMessage(f'ienākošais UDP ziņojums: {message}', 'RONIN', 0)
                openResult('detection')
            elif message == 'ierobezosana pabeigta':
                QgsMessageLog.logMessage(f'ienākošais UDP ziņojums: {message}', 'RONIN', 0)
                openResult('treatment')
            else:
                QgsMessageLog.logMessage(f'nederīgs ienākošais UDP ziņojums: {message}', 'RONIN', 0)

            self.stop_event.wait(0.1)

#poga atpazīt burkānus
buttonCarrot.clicked.connect(detectCarrots)
pixmap = QPixmap('icons/carrot.png')
buttonCarrot.setIcon(QIcon(pixmap))
toolbar.addWidget(buttonCarrot)

#poga atpazīt bietes
buttonBeet.clicked.connect(detectBeets)
pixmap = QPixmap('icons/beet.png')
buttonBeet.setIcon(QIcon(pixmap))
toolbar.addWidget(buttonBeet)

#poga atpazīt ķirbi
buttonPumpkin.clicked.connect(detectPumpkins)
pixmap = QPixmap('icons/pumpkin.png')
buttonPumpkin.setIcon(QIcon(pixmap))
toolbar.addWidget(buttonPumpkin)

#poga atpazīt cukīni
buttonZucckini.clicked.connect(detectZucchini)
pixmap = QPixmap('icons/zucchini.png')
buttonZucckini.setIcon(QIcon(pixmap))
toolbar.addWidget(buttonZucckini)

#poga atpazīt redisu
buttonRadish.clicked.connect(detectRadishes)
```

```
pixmap = QPixmap('icons/radish.png')
buttonRadish.setIcon(QIcon(pixmap))
toolbar.addWidget(buttonRadish)

#poga atpazīt rutku
buttonBlackRadish.clicked.connect(detectBlackRadishes)
pixmap = QPixmap('icons/radish_black.png')
buttonBlackRadish.setIcon(QIcon(pixmap))
toolbar.addWidget(buttonBlackRadish)

toolbar.addSeparator()

#poga izvēlēties urbi
buttonDrill.clicked.connect(selectDrill)
pixmap = QPixmap('icons/drill.png')
buttonDrill.setIcon(QIcon(pixmap))
buttonDrill.setStyleSheet('width: 24px')
toolbar.addWidget(buttonDrill)

#poga izvēlēties lāzeri
buttonLaser.clicked.connect(selectLaser)
pixmap = QPixmap('icons/laser.png')
buttonLaser.setIcon(QIcon(pixmap))
buttonLaser.setStyleSheet('width: 24px')
toolbar.addWidget(buttonLaser)

toolbar.addSeparator()

#poga atpazīšanai
buttonStartObjectDetection.clicked.connect(startObjectDetection)
pixmap = QPixmap('icons/eye.png')
buttonStartObjectDetection.setIcon(QIcon(pixmap))
buttonStartObjectDetection.setStyleSheet('width: 24px')
toolbar.addWidget(buttonStartObjectDetection)

toolbar.addSeparator()

#poga nezāļu ierobežošanai
buttonStartWeeding.clicked.connect(startWeeding)
pixmap = QPixmap('icons/weeding.png')
buttonStartWeeding.setIcon(QIcon(pixmap))
buttonStartWeeding.setStyleSheet('width: 24px')
toolbar.addWidget(buttonStartWeeding)

toolbar.addSeparator()

#poga atpazīšanai un nezāļu ierobežošanai
buttonStartObjectDetectionAndWeeding.clicked.connect(startObjectDetectionAndWeeding)
pixmap = QPixmap('icons/eye_plus_weeding.png')
buttonStartObjectDetectionAndWeeding.setIcon(QIcon(pixmap))
buttonStartObjectDetectionAndWeeding.setStyleSheet('width: 73px')
toolbar.addWidget(buttonStartObjectDetectionAndWeeding)

toolbar.addSeparator()

#poga GPS punkta pievienošanai maršrutam
buttonAddGpsPointToRoute.clicked.connect(addGpsPointToRoute)
pixmap = QPixmap('icons/gpsPointRoute.png')
buttonAddGpsPointToRoute.setIcon(QIcon(pixmap))
buttonAddGpsPointToRoute.setStyleSheet('width: 24px')
toolbar.addWidget(buttonAddGpsPointToRoute)

#poga GPS maršrutam izdzēšanai
buttonResetGpsRoute.clicked.connect(resetGpsRoute)
pixmap = QPixmap('icons/resetGpsRoute.png')
buttonResetGpsRoute.setIcon(QIcon(pixmap))
buttonResetGpsRoute.setStyleSheet('width: 24px')
toolbar.addWidget(buttonResetGpsRoute)

#poga GPS punkta pievienošanai vagai
```

```
                buttonAddGpsPointToRidge.clicked.connect(addGpsPointToRidge)
                pixmap = QPixmap('icons/gpsPointRidge.png')
                buttonAddGpsPointToRidge.setIcon(QIcon(pixmap))
                buttonAddGpsPointToRidge.setStyleSheet('width: 24px')
                toolbar.addWidget(buttonAddGpsPointToRidge)

                #poga GPS vagas maršrutam izdzēšanai
                buttonResetGpsRidge.clicked.connect(resetGpsRidge)
                pixmap = QPixmap('icons/resetGpsRoute.png')
                buttonResetGpsRidge.setIcon(QIcon(pixmap))
                buttonResetGpsRidge.setStyleSheet('width: 24px')
                toolbar.addWidget(buttonResetGpsRidge)

                toolbar.addSeparator()

                #poga braukšanai pēc GPS maršruta
                buttonStartGpsRoute.clicked.connect(startGpsRoute)
                pixmap = QPixmap('icons/route.png')
                buttonStartGpsRoute.setIcon(QIcon(pixmap))
                buttonStartGpsRoute.setStyleSheet('width: 24px')
                toolbar.addWidget(buttonStartGpsRoute)

                toolbar.addSeparator()

                #poga braukšanai pēc GPS maršruta un sekošana vagai
                buttonStartGpsRouteAndFollowRow.clicked.connect(startGpsRouteAndFollowRow)
                pixmap = QPixmap('icons/route_plus_row.png')
                buttonStartGpsRouteAndFollowRow.setIcon(QIcon(pixmap))
                buttonStartGpsRouteAndFollowRow.setStyleSheet('width: 73px')
                toolbar.addWidget(buttonStartGpsRouteAndFollowRow)

                toolbar.addSeparator()

                #poga braukšanai pēc GPS maršruta, sekošana vagai, atpazīšana un nezāļu ierobežošana

buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.clicked.connect(startGpsRouteAndFollowRowAndObjectDetectionAndW
eeding)
                pixmap = QPixmap('icons/route_plus_row_plus_eye_plus_weeding.png')
                buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding.setIcon(QIcon(pixmap))
                toolbar.addWidget(buttonStartGpsRouteAndFollowRowAndObjectDetectionAndWeeding)

                toolbar.addSeparator()

                #poga avārijas apstāšanās
                buttonEmergencyStop.clicked.connect(emergencyStop)
                pixmap = QPixmap('icons/stop.png')
                buttonEmergencyStop.setIcon(QIcon(pixmap))
                buttonEmergencyStop.setStyleSheet('width: 24px')
                toolbar.addWidget(buttonEmergencyStop)

                udpServer = udpServerThread()
                udpServer.daemon = True
                udpServer.start()

                pass

            def saveProject():
                pass

            def closeProject():
                pass
```