*장현주\_2023011923*

|  |
| --- |
| **2024 ALTIS SW 개인 활동 보고서** |

|  |  |
| --- | --- |
| **활동 개요** | (통신) 3D시각화 수정, 지도 매핑 |
| **일시** | 2024년 5월 26일 00시 |
| **작성자** | 김민주, 장현주 |
| **활동 내용** | *※ 어떤 활동을 진행하였는지 자세히 쓰기*  *※ 코드가 있다면 어떤 코드인지 + 코드에 주석 달기*  (visual3D 부분코드)  import sys  from PyQt5.QtWidgets import QApplication, QMainWindow, QOpenGLWidget, QVBoxLayout, QWidget, QPushButton, QLabel  from PyQt5.QtCore import QTimer  from OpenGL.GL import \*  from OpenGL.GLU import \*  import serial  from datetime import datetime  from SerialCommincation import \*  class Communication:  def \_\_init\_\_(self, port: str, baudrate: int):  self.port = port  self.baudrate = baudrate  self.ser = serial.Serial(port, baudrate)  def getData(self):  if self.ser.in\_waiting > 0:  value = self.ser.readline()  decoded = str(value[0:len(value) - 2].decode("utf-8"))  value\_table = decoded.split(",")  value\_table.insert(0, self.getDate())  else:  value\_table = []  return value\_table  def getDate(self):  now = datetime.now()  date = now.strftime("%Y-%m-%d %H:%M:%S")  return date  def isOpen(self):  return self.ser.isOpen()  class GLWidget(QOpenGLWidget):  def \_\_init\_\_(self, parent=None):  super(GLWidget, self).\_\_init\_\_(parent)  self.com = Communication('COM6', 115200)    self.xangle, self.yangle, self.zangle = 0, 0, 0  self.timer = QTimer(self)  self.timer.timeout.connect(self.updateAngles)  self.timer.start(33) # 약 30fps로 업데이트  def initializeGL(self):  glEnable(GL\_DEPTH\_TEST)  glClearColor(0.1, 0.1, 0.1, 1.0)  gluPerspective(45, 800/600, 0.1, 50.0)  glTranslatef(0.0, 0.0, -5)  def resizeGL(self, width, height):  glViewport(0, 0, width, height)  glMatrixMode(GL\_PROJECTION)  glLoadIdentity()  gluPerspective(45, width / height, 0.1, 50.0)  glMatrixMode(GL\_MODELVIEW)  def paintGL(self):  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  glLoadIdentity()  # 오른쪽에서 로켓을 바라보는 카메라 설정  gluLookAt(5, 0, 0, 0, 0, 0, 0, 0, 1)  glPushMatrix()  glRotatef(self.xangle, 1, 0, 0)  glRotatef(self.yangle, 0, 1, 0)  glRotatef(self.zangle, 0, 0, 1)  self.drawRocket()  glPopMatrix()  def drawRocket(self):  body\_length = 2.0  body\_radius = 0.2  cone\_height = 0.6  cone\_radius = 0.2  # 로켓 몸체 (원기둥)  glPushMatrix()  glTranslatef(0, 0, -body\_length / 2)  glColor3f(0.8, 0.8, 0.8) # Light gray  quadric = gluNewQuadric()  gluCylinder(quadric, body\_radius, body\_radius, body\_length, 32, 32)  glPopMatrix()  # 로켓 머리 (원뿔)  glPushMatrix()  glTranslatef(0, 0, body\_length / 2)  glColor3f(1.0, 0.0, 0.0) # Red  quadric = gluNewQuadric()  gluCylinder(quadric, cone\_radius, 0.0, cone\_height, 32, 32)  glPopMatrix()  # 로켓 꼬리 (디스크)  glPushMatrix()  glTranslatef(0, 0, -body\_length / 2)  glRotatef(180, 1, 0, 0)  glColor3f(0.0, 0.0, 1.0) # Blue  gluDisk(quadric, 0.0, body\_radius, 32, 32)  glPopMatrix()  def updateAngles(self):  if self.com.isOpen():  try:  data\_table = self.com.getData()  if len(data\_table) == 4:  self.xangle = float(data\_table[1])  self.yangle = float(data\_table[2])  self.zangle = float(data\_table[3])  self.update() # 새 각도로 다시 그리기  except ValueError:  pass # Serial 통신 오류를 처리  (지도 코드에서 메인코드 가져옴)  import sys #인터프리터와 관련된 기능을 제공  from PyQt5.QtWidgets import \*  from PyQt5.QtGui import QIcon  from PyQt5.QtCore import QTimer, QDateTime  from PyQt5.QtWebEngineWidgets import QWebEngineView  from PyQt5.QtCore import QUrl  from bluetooth import BluetoothGPS  from PyQt5.QtWidgets import QLabel  import random  import folium #지도생성  import os  from visual3D import \*  from altitude import \*  from a3 import \*  from a2 import \*  from groupbutton import \*  from gps import \*  #블루투스GPS장치로부터 위도와 경도 데이터를 가져옴  def get\_gps\_data():  try:  bluetooth\_gps = BluetoothGPS(serial\_port="COM14") # 블루투스 GPS 클래스 인스턴스 생성  bluetooth\_gps.connect() # 블루투스 GPS와 연결  latitude, longitude = bluetooth\_gps.get\_gps\_data() # GPS 데이터 가져오기  bluetooth\_gps.disconnect() # 블루투스 GPS 연결 해제  return latitude, longitude  except Exception as e:  print(f"GPS 데이터 수신 실패: {e}")  return None, None # 데이터를 가져오지 못한 경우 None 반환    #GPS위치와 지도를 처리하는 클래스를 정의  class gps:  def \_\_init\_\_(self):  self.default\_latitude = 35.152835721922976 # 초기 위도  self.default\_longitude = 128.1041700584298 # 초기 경도  self.fixed\_latitude = 35.15416150364563 # 고정 위도  self.fixed\_longitude = 128.0928933620453 # 고정 경도  self.current\_latitude = self.default\_latitude  self.current\_longitude = self.default\_longitude  self.map\_file = "map.html" # 지도를 가져오는 부분  self.label = None # QLabel 위젯 초기화  self.create\_map() # 초기 지도 생성  def update\_location(self, latitude, longitude):  if latitude is not None and longitude is not None:  self.current\_latitude = latitude  self.current\_longitude = longitude  self.create\_map()  def create\_map(self):  map = folium.Map(location=[self.current\_latitude, self.current\_longitude], zoom\_start=13)  folium.Marker([self.current\_latitude, self.current\_longitude], tooltip='Current Location').add\_to(map)  folium.Marker([self.fixed\_latitude, self.fixed\_longitude], tooltip='Fixed Location', icon=folium.Icon(color='red')).add\_to(map)  if self.label:  self.label.setText(f'Latitude: {self.current\_latitude}, Longitude: {self.current\_longitude}')  else:  self.label = QLabel(f'Latitude: {self.current\_latitude}, Longitude: {self.current\_longitude}')  self.label.setFixedSize(200, 50) # QLabel 크기 설정  self.label.move(10, 10) # QLabel 위치 설정  map.save(self.map\_file)  def get\_map\_url(self):  return QUrl.fromLocalFile(os.path.abspath(self.map\_file))  #GPS를 표시하는 위젯  class gpsWidget(QWidget):  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.gps = gps()  self.view = QWebEngineView()  self.label = self.gps.label  self.initUI()  def initUI(self):  vbox = QVBoxLayout()  vbox.addWidget(self.view)  vbox.addWidget(self.label) # 라벨 추가  self.setLayout(vbox)  self.update\_map()  def update\_map(self):  self.gps.create\_map()  self.view.setUrl(self.gps.get\_map\_url())  def update\_location(self, latitude, longitude):  self.gps.update\_location(latitude, longitude)  self.update\_map()  class guiMain(QMainWindow):  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.initUI()  self.setupBluetooth() # 블루투스 설정 함수 호출  def initUI(self): # 창 제속이나 아이콘, 크기, 위치를 설정  self.setWindowTitle('GUI')  self.setWindowIcon(QIcon('.vscode/windowicon.jpg'))  self.setGeometry(150, 100, 2000, 1500)  # Timer to update the status bar with the current date and time  self.timer = QTimer(self)  self.timer.timeout.connect(self.updateStatusBar)  self.timer.start(1000) # Update every second  mainLayout = QVBoxLayout()  vbox1 = QVBoxLayout()  vbox2 = QVBoxLayout()  vbox3 = QVBoxLayout()  hbox1 = QHBoxLayout()  #Layout - attribute  altitude\_lay = self.setAltitudeGraph()  a2\_lay = self.seta2Graph()  a3\_lay = self.seta3Graph()  visual3D\_lay = self.Visual3DMotion()  groupbutton\_lay = self.setgroupButton()  gps\_lay = self.setGPSMap()  vbox1.addLayout(altitude\_lay,2)  vbox1.addLayout(a2\_lay,2)  vbox1.addLayout(a3\_lay,2)  vbox2.addLayout(visual3D\_lay,4)  vbox3.addLayout(gps\_lay, 4)  vbox3.addLayout(groupbutton\_lay,2)  hbox1.addLayout(vbox1, 3)  hbox1.addLayout(vbox2, 4)  hbox1.addLayout(vbox3, 2)  mainLayout.addLayout(hbox1)  # Create a central widget and set the layout  centralWidget = QWidget()  centralWidget.setLayout(mainLayout)  self.setCentralWidget(centralWidget)  #self.setLayout(vbox)  self.show()  #상태창  def updateStatusBar(self):  current\_date\_time = QDateTime.currentDateTime().toString('yyyy-MM-dd HH:mm:ss')  self.statusBar().showMessage(current\_date\_time)  #고도 그래프  def setAltitudeGraph(self):  self.altitude = graph\_altitude()  vbox = QHBoxLayout()  vbox.addWidget(self.altitude)  return vbox    #3D 시각화 로켓 자세  def Visual3DMotion(self):  self.visual3D = visual3D()    vbox = QHBoxLayout()  vbox.addWidget(self.visual3D)  return vbox  # group버튼  def setgroupButton(self):  self.groupButton = group\_button()    vbox = QHBoxLayout()  vbox.addWidget(self.groupButton)  return vbox  # 그래프 2  def seta2Graph(self):  self.a2 = graph\_a2()  vbox = QHBoxLayout()  vbox.addWidget(self.a2)  return vbox  # 그래프 3  def seta3Graph(self):  self.a3 = graph\_a3()  vbox = QHBoxLayout()  vbox.addWidget(self.a3)  return vbox  def drawGraphs(self):  random\_number = random.randint(1,10)  self.graph.update(random\_number)  # 블루투스 설정  def setupBluetooth(self):  self.bluetoothTimer = QTimer(self)  self.bluetoothTimer.timeout.connect(self.updateGPS)  self.bluetoothTimer.start(1000) # 1초마다 업데이트  # GPS 업데이트 함수  def updateGPS(self):  latitude, longitude = get\_gps\_data()  self.gpsWidget.update\_location(latitude, longitude)  # GPS 지도 설정  def setGPSMap(self):  self.gpsWidget = gpsWidget()  vbox = QVBoxLayout()  vbox.addWidget(self.gpsWidget)  return vbox  if \_\_name\_\_ == '\_\_main\_\_':  app = QApplication(sys.argv)  ex = guiMain()  sys.exit(app.exec\_()) |
| **활동 사진** | *※ 활동하고 있는 모습, 회로도, 완성된 모습, 실행화면 등..* |
| **활동 결과** | *※ 활동 내용을 통해 나온 결과 정리*  3D창을 띄우는 것까지 성공하였습니다.  지도 위에 마커를 2개가 찍힐 수 있게 수정하였고, GPS가 연결되면, (**35.152835721922976, 128.1041700584298)에 설정된 마커가 움직일 수 있게 하였다. 하지만 GPS를 1초마다 받아오게 설정하여서 지도도 같이 새로 고침이 되어 버린다.** |
| **계획** | *※ 다음 활동 시 해야할 일* |

2024년 5 월 26 일

작성자 : 김민주, 장현주 (인)