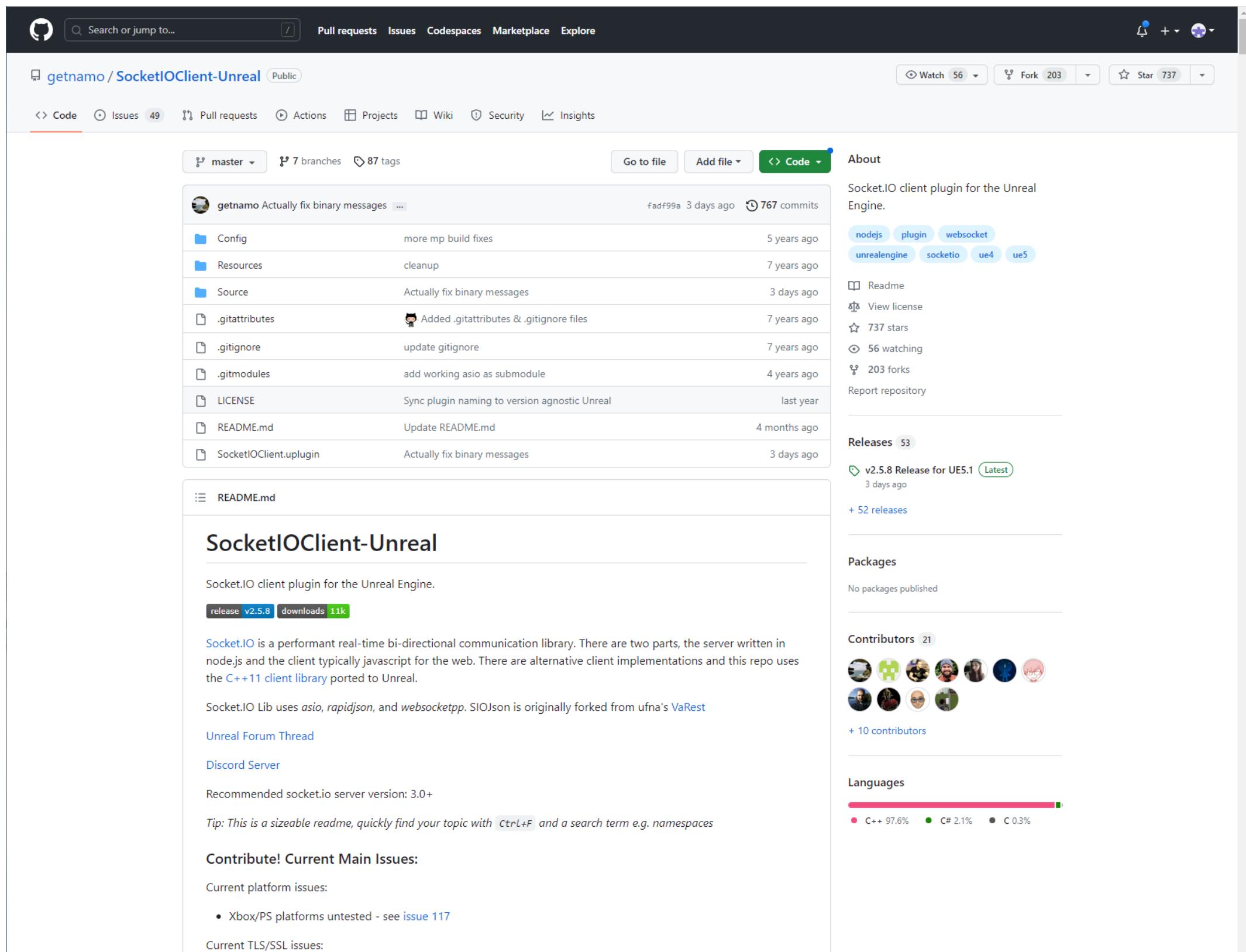


# Redirected Walking

230511 19101188 고은수

# Todo

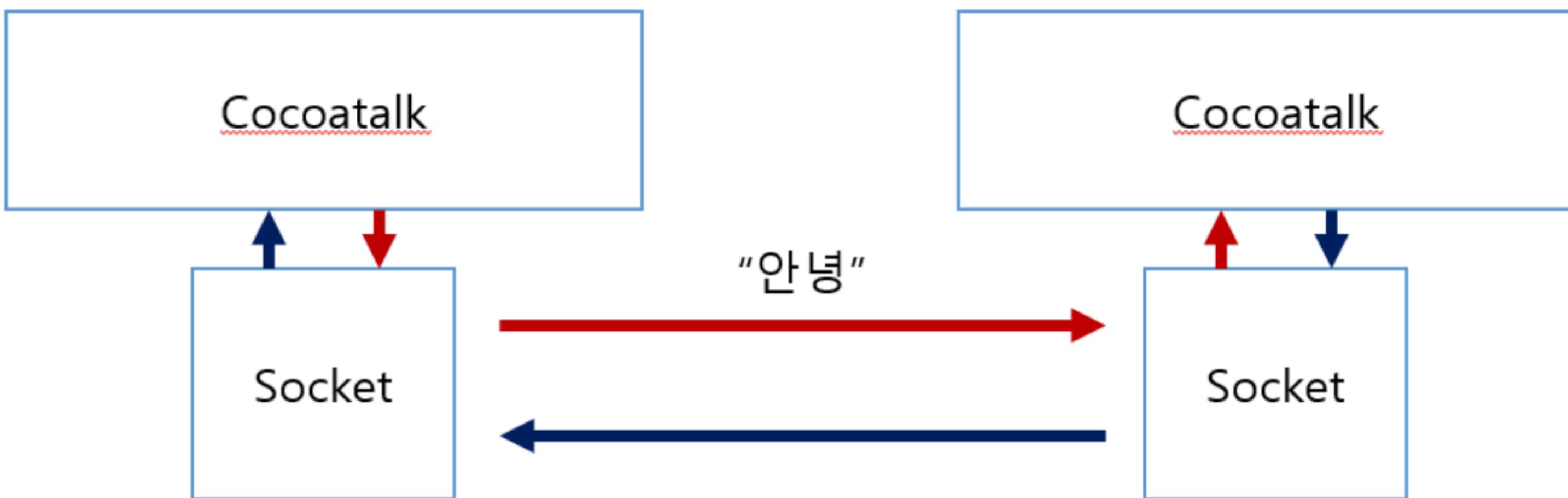
## SocketIO 플러그인 이용하여 Unreal과 외부 서버간의 통신으로 위치좌표 전송하기



# Socket?

소켓은 떨어져있는 두 호스트를 연결해주어 데이터를 주고 받을 수 있게 해주는 도구로써 인터페이스의 역할을 하는 구조체이다.

소켓은 서버와 클라이언트로 나뉘어지며 애플리케이션 자체로는 외부 네트워크와 아무런 정보도 주고받지 않고 소켓들 사이에서만 통신이 이루어진다.



# Socket 이용한 채팅

```
socket > server.py
socket - server.py
server.py x client.py x
from socket import *
import threading
import time

def send(sock):
    while True:
        sendData = input('>>>')
        sock.send(sendData.encode('utf-8'))

def receive(sock):
    while True:
        recvData = sock.recv(1024)
        print('상대방 :', recvData.decode('utf-8'))

port = 8080

serverSock = socket(AF_INET, SOCK_STREAM)
serverSock.bind(('', port))
serverSock.listen(1)

print('%d번 포트로 접속 대기중...' % port)

connectionSock, addr = serverSock.accept()

print(str(addr), '에서 접속되었습니다.')

sender = threading.Thread(target=send, args=(connectionSock,))
```

Server

```
socket > client.py
socket - client.py
client.py x server.py x
from socket import *
import threading
import time

def send(sock):
    while True:
        sendData = input('>>>')
        sock.send(sendData.encode('utf-8'))

def receive(sock):
    while True:
        recvData = sock.recv(1024)
        print('상대방 :', recvData.decode('utf-8'))

port = 8080

clientSock = socket(AF_INET, SOCK_STREAM)
clientSock.connect(('127.0.0.1', port))

print('접속 완료')

sender = threading.Thread(target=send, args=(clientSock,))
receiver = threading.Thread(target=receive, args=(clientSock,))

sender.start()
receiver.start()
```

Client

실행화면

실행: server x client x

/usr/local/bin/python3.11 /Users/ko/PycharmProjects/socket/server.py

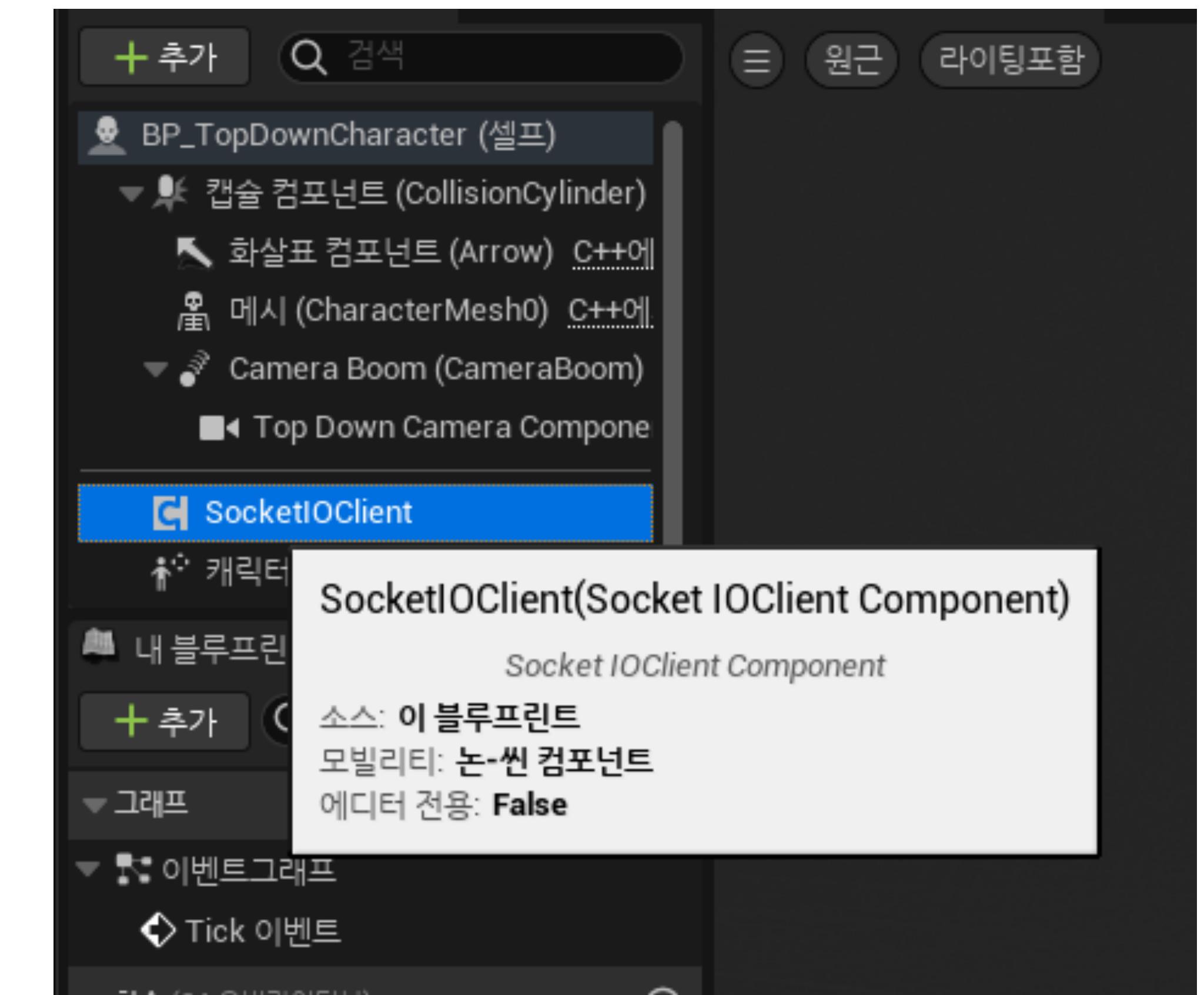
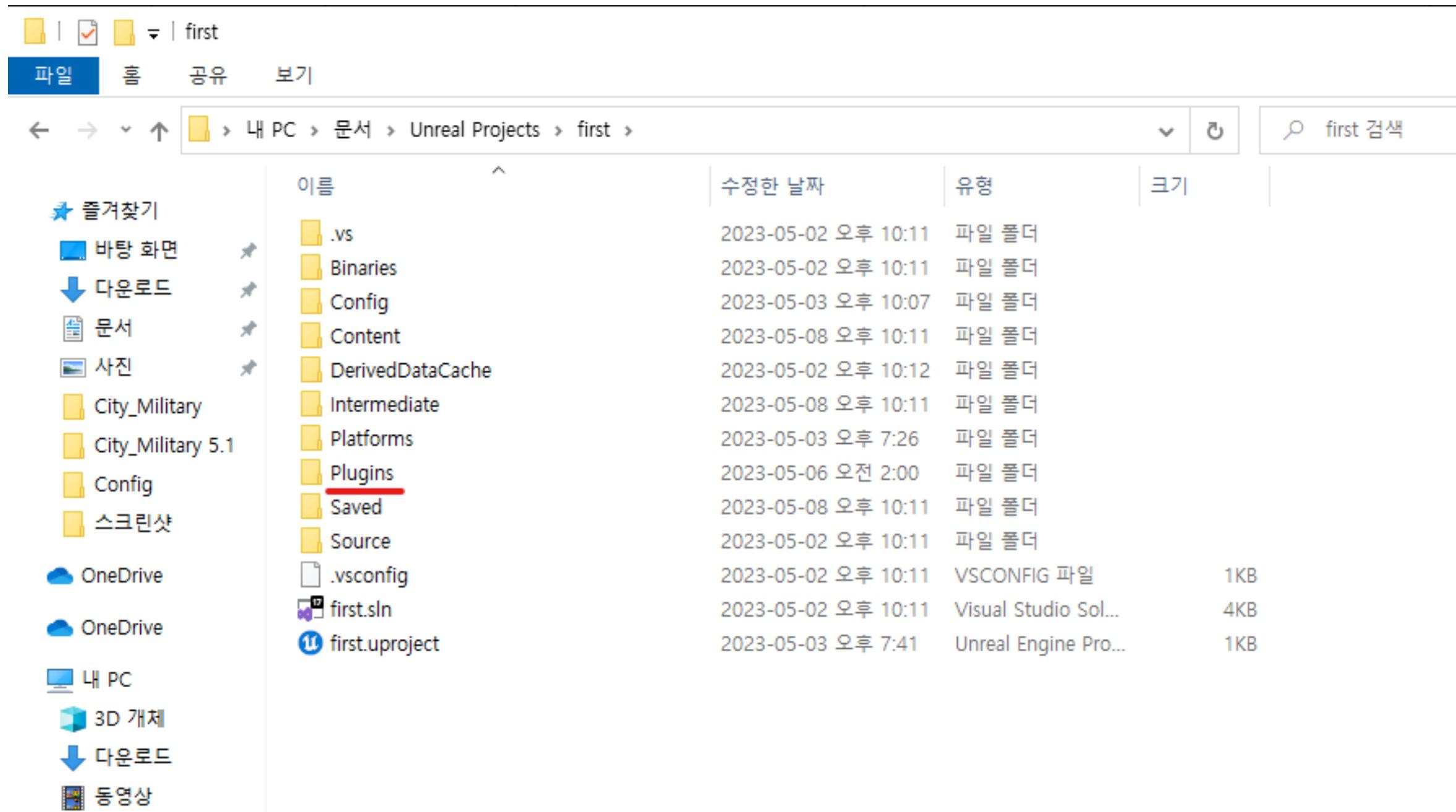
8080번 포트로 접속 대기중...

('127.0.0.1', 60320) 에서 접속되었습니다.

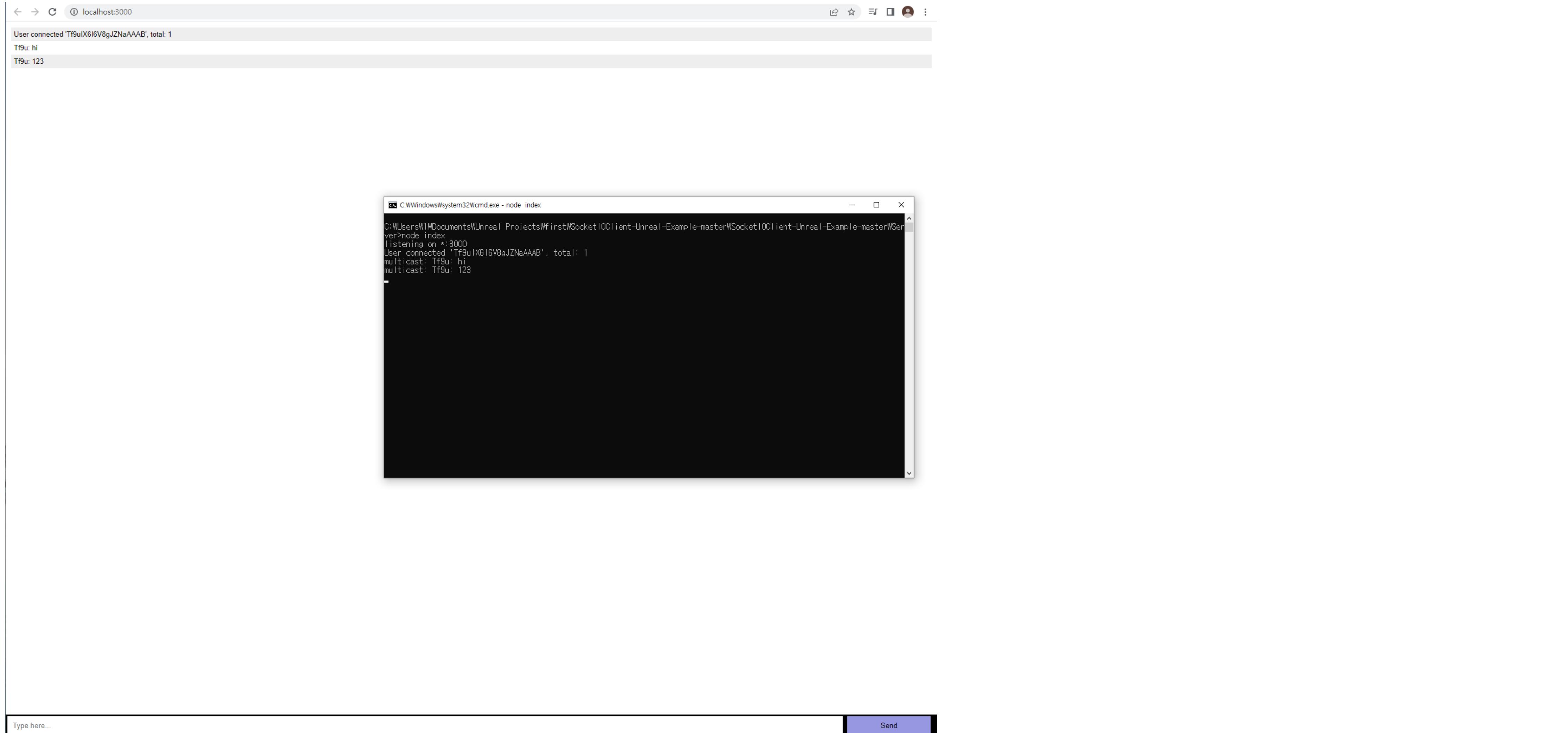
>>>상대방 : hi

상대방 : 123

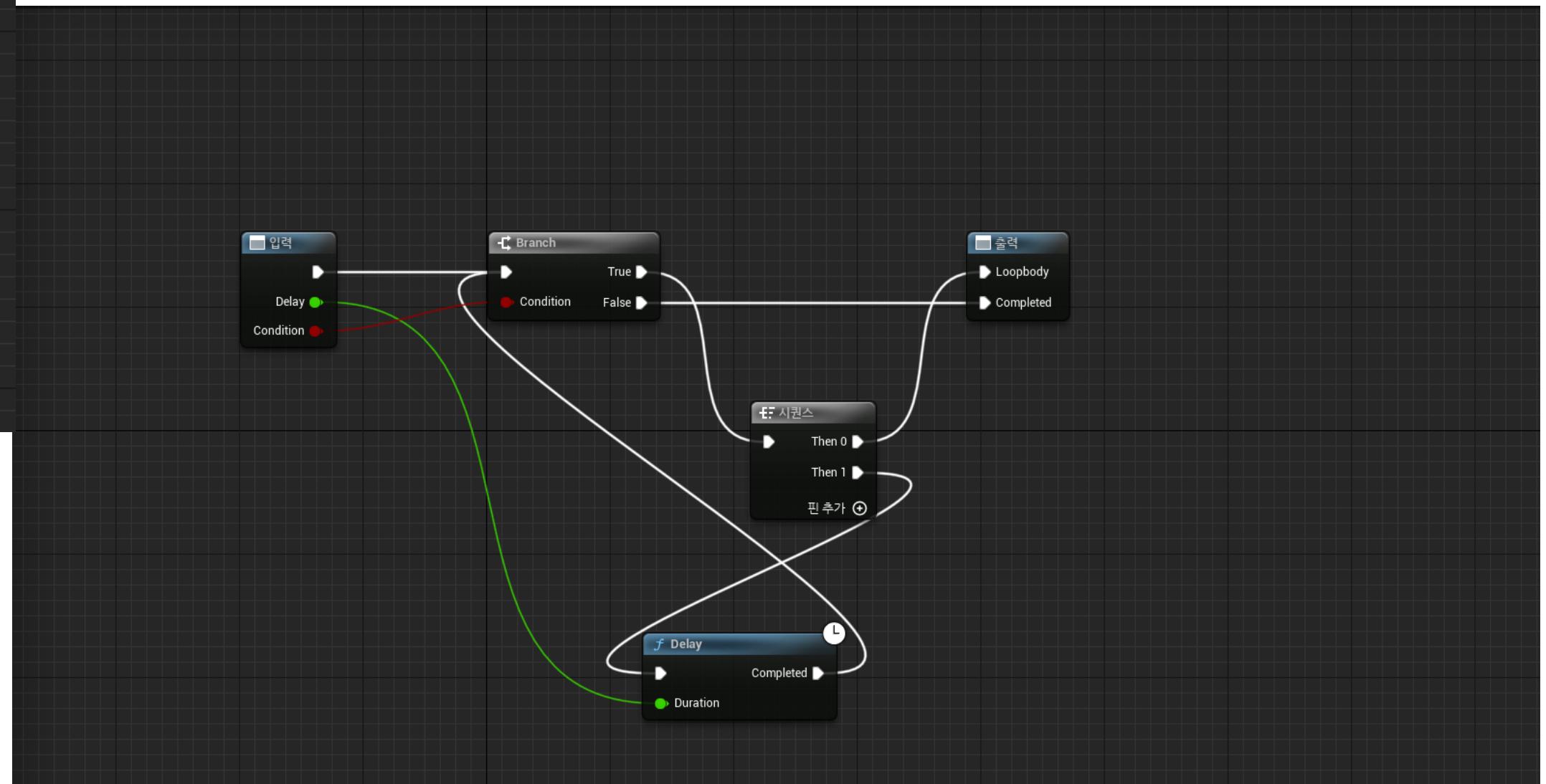
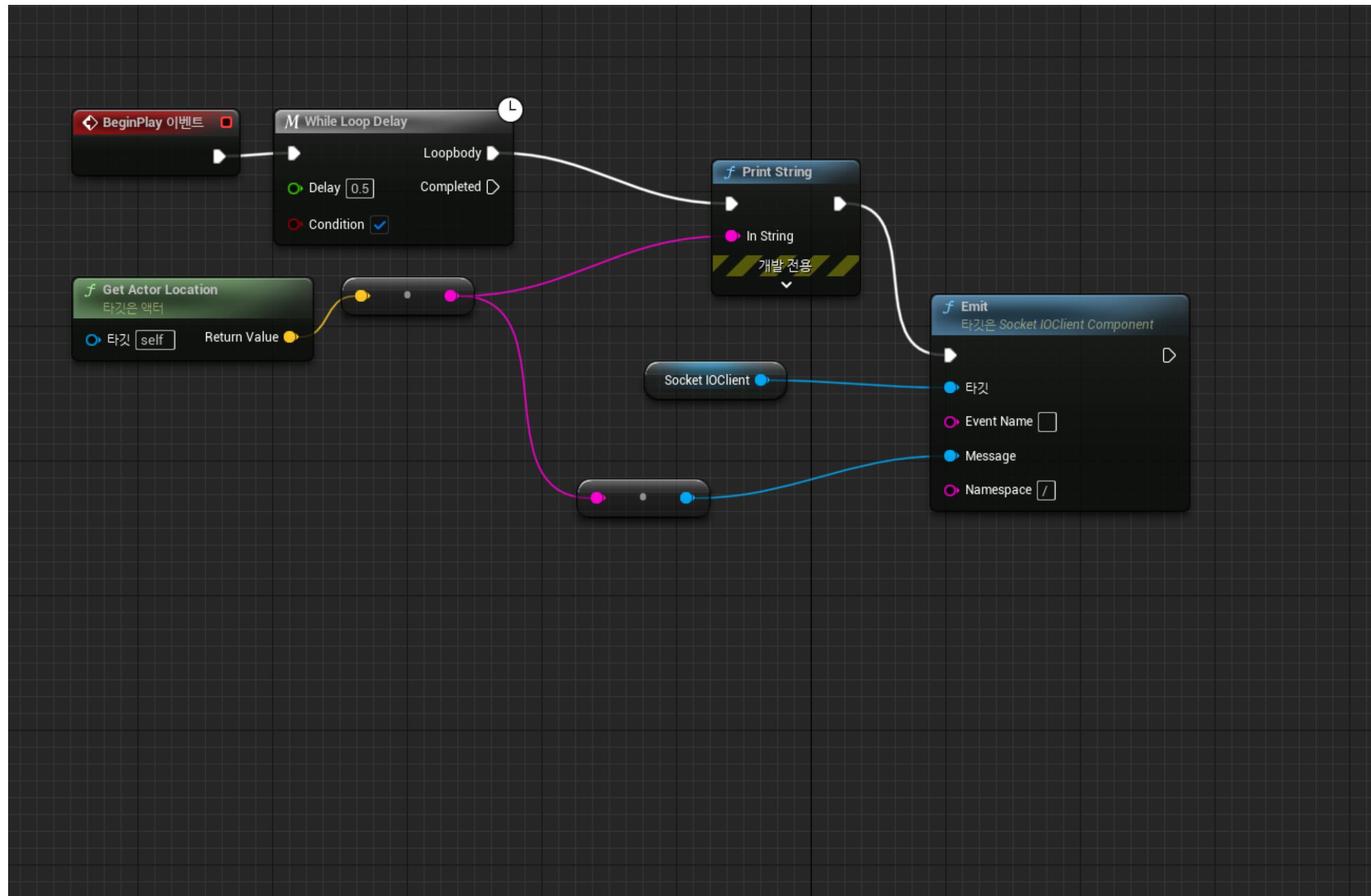
# SocketIO Plugin 설치



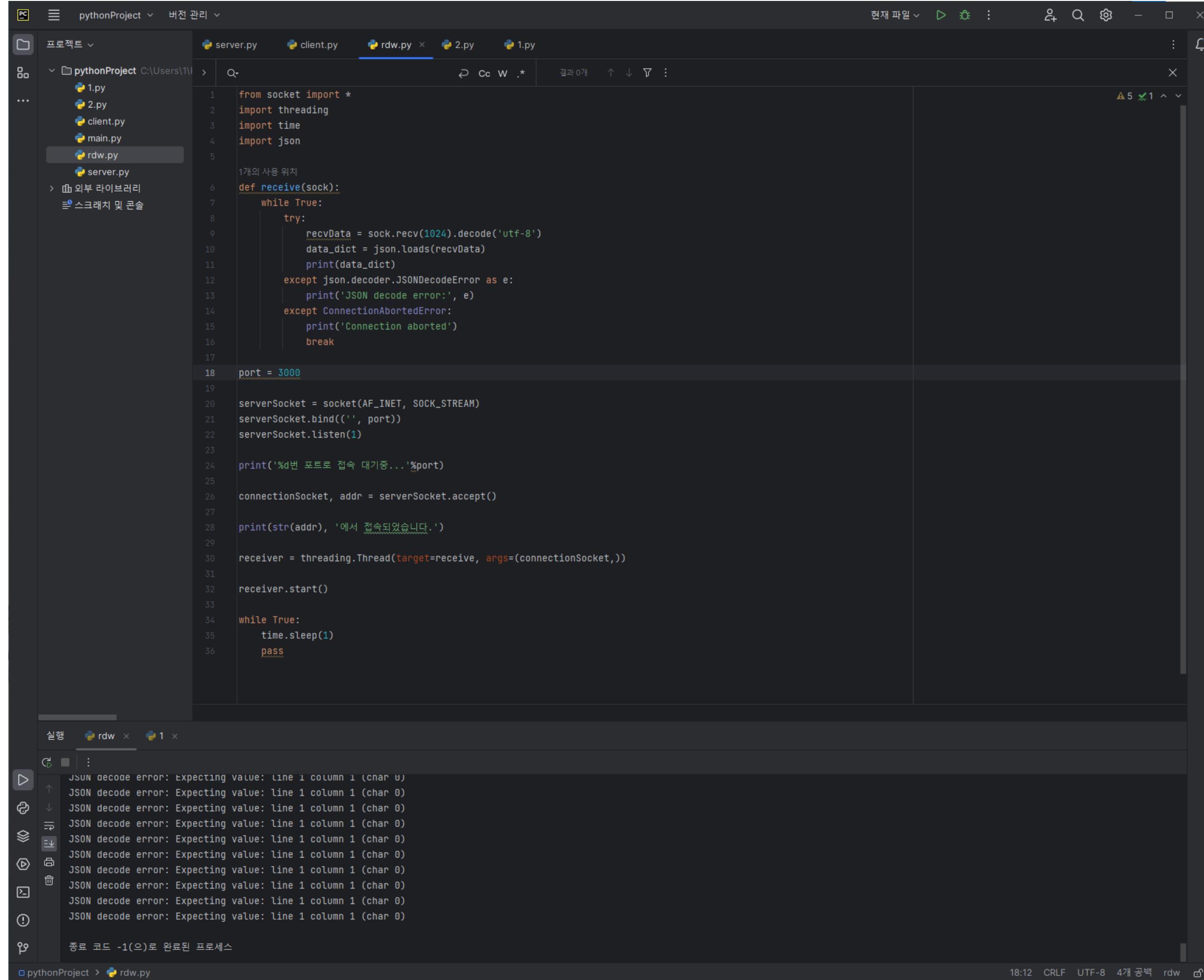
# 예제



# Unreal Charactor의 위치좌표 전송



# 채팅 예제를 이용해서 만든 Python Server (Error)



```
pythonProject 버전 관리
pythonProject C:\Users\1\ ...
1.py 2.py client.py rdw.py 2.py 1.py

...
1. py
2. py
client.py
main.py
rdw.py
server.py
> 일부 외부 라이브러리
스크래치 및 콘솔

1
from socket import *
import threading
import time
import json

def receive(sock):
    while True:
        try:
            recvData = sock.recv(1024).decode('utf-8')
            data_dict = json.loads(recvData)
            print(data_dict)
        except json.decoder.JSONDecodeError as e:
            print('JSON decode error:', e)
        except ConnectionAbortedError:
            print('Connection aborted')
            break

port = 3000

serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', port))
serverSocket.listen(1)

print('%d번 포트로 접속 대기중...' % port)

connectionSocket, addr = serverSocket.accept()

print(str(addr), '에서 접속되었습니다.')

receiver = threading.Thread(target=receive, args=(connectionSocket,))

receiver.start()

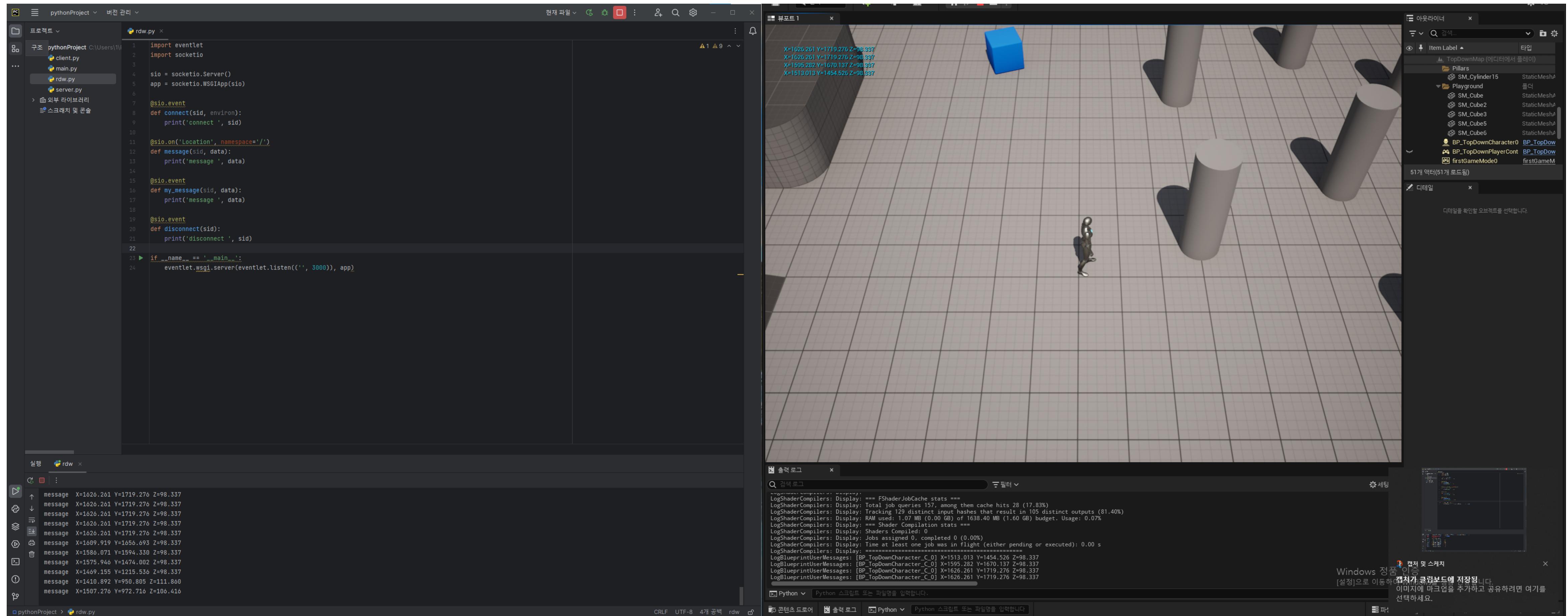
while True:
    time.sleep(1)
    pass
```

실행 rdw x 1 x

JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
JSON decode error: Expecting value: line 1 column 1 (char 0)  
종료 코드 -1(으)로 원료된 프로세스

pythonProject > rdw.py

# Error 해결하고 unreal -> client 위치좌표 전송



# Python server -> Unreal client로 전송

위치좌표를 각각 +100, +100, +100해서 다시 unreal client로 전송

