

스마트네트워크서비스

# 다중스레드를 이용한 채팅과 캐치마인드

홍심황 조

20203166 홍현지

20203168 황솔희

20213022 심지아

# 주제 선정 이유

1. 대화상자 기반 응용 프로그램 제작을 직접 해봄으로써 구조와 동작 원리 이해하고자함
2. 단순 채팅 시스템이 아니라 기능을 추가하여 다양하게 제작해보는 것은 어떨까?

# 시나리오

## 1. 다중스레드를 이용한 채팅 시스템

: 하나의 서버에 여러개의 클라이언트를 연결하여 일대다 채팅 시스템 제작

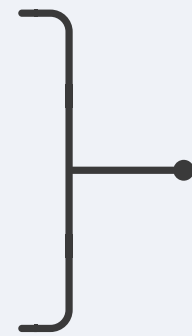
## 2. 캐치마인드 유형 게임 시스템 추가

: 단순히 서버 - 클라이언트 간의 채팅 기능으로 끝나는 것이 아닌 주어진 단어를 그림판을 통해 그려서 문제를 내고 맞출 수 있는 기능 제작

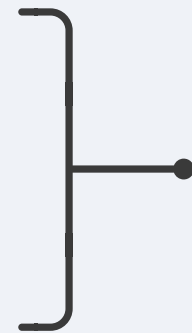
# 시나리오

## 파일 구성

- client.py
- server.py
- clientWindow.py
- serverWindow.py



소켓 통신

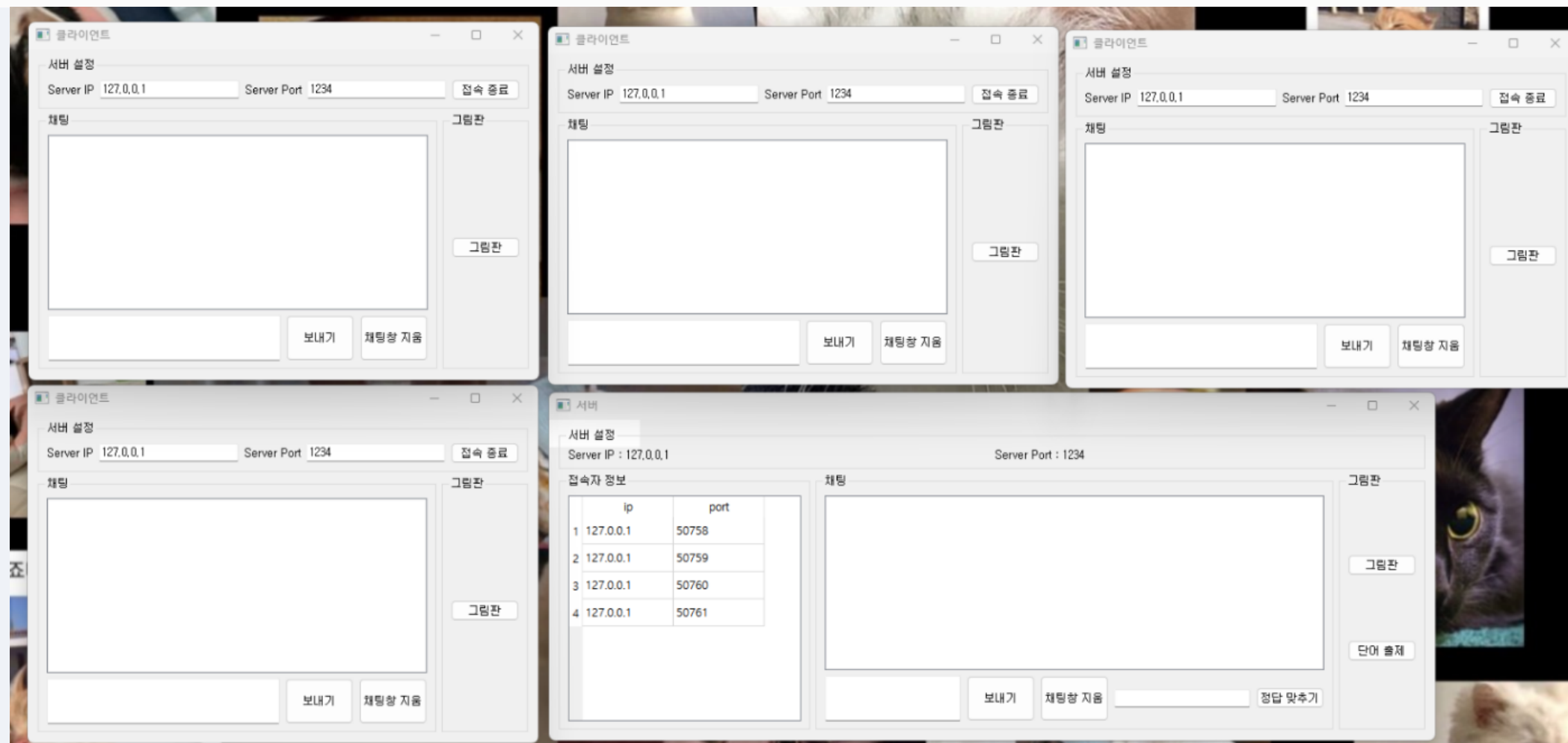


GUI

# 프로그램 구조

## 실행 방법

- 1개의 serveWindow.py (서버 파일) 실행
- N개의 clientWindow.py (클라이언트 파일) 실행



# 프로그램 구조

```
def start(self, ip, port):
    self.server = socket(AF_INET, SOCK_STREAM)

    try:
        self.server.bind( (ip, port))
    except Exception as e:
        print('Bind Error : ', e)
        return False
    else:
        self.bListen = True
        self.t = Thread(target=self.listen, args=(self.server,))
        self.t.start()
        print('Server Listening...')

    return True
```

server.py

```
def connectServer(self, ip, port):
    self.client = socket(AF_INET, SOCK_STREAM)

    try:
        self.client.connect( (ip, port) )
    except Exception as e:
        print('Connect Error : ', e)
        return False
    else:
        self.bConnect = True
        self.t = Thread(target=self.receive, args=(self.client,))
        self.t.start()
        print('Connected')

    return True
```

client.py

서버 소켓 통신 시작, 클라이언트에서 서버로 연결

# 프로그램 구조

```
def stop(self):  
    self.bListen = False  
    if hasattr(self, 'server'):  
        self.server.close()  
        print('Server Stop')
```

server.py

```
def stop(self):  
    self.bConnect = False  
    if hasattr(self, 'client'):  
        self.client.close()  
        del(self.client)  
        print('Client Stop')  
        self.disconnect.disconnect_signal.emit()
```

client.py

**소켓 통신 중단 (접속 종료)**

# 프로그램 구조

```
def send(self, msg):  
    try:  
        for c in self.clients:  
            c.send(msg.encode())  
    except Exception as e:  
        print('Send() Error : ', e)
```

server.py

```
def send(self, msg):  
    if not self.bConnect:  
        return  
  
    try:  
        self.client.send(msg.encode())  
    except Exception as e:  
        print('Send() Error : ', e)
```

client.py

각 클라이언트 / 서버로 메시지 전송



# 프로그램 구조

```
def receive(self, addr, client):
    while True:
        try:
            recv = client.recv(1024)
        except Exception as e:
            print('Recv() Error :', e)
            break
        else:
            msg = str(recv, encoding='utf-8')
            if msg.startswith('Drawing Coordinates:'):
                coordinates_str = msg.split(',')[1].rsplit(',', 1)[0]
                coordinates_list = eval('[' + coordinates_str + ']')
                self.recv_signal.emit("그림이 왔습니다. 확인해 보세요")
                print('[RECV]:', addr, msg)
                self.parent.handle_drawing_coordinates(coordinates_list) # 좌표 리스트만 전달
            elif msg:
                self.recv_signal.emit(msg)
                print('[RECV]:', addr, msg)

self.removeClient(addr, client)
```

server.py

```
def receive(self, client):
    while self.bConnect:
        try:
            recv = client.recv(1024)
        except Exception as e:
            print('Recv() Error :', e)
            break
        else:
            msg = str(recv, encoding='utf-8')
            if msg:
                self.recv.recv_signal.emit(msg)
                print('[RECV]:', msg)

self.stop()
```

client.py

서버 소켓 통신 시작, 클라이언트에서 서버로 연결

# 프로그램 구조

server.py

```
def listen(self, server):
    while self.blisten:
        server.listen(5)
        try:
            client, addr = server.accept()
        except Exception as e:
            print('Accept() Error : ', e)
            break
        else:
            self.clients.append(client)
            self.ip.append(addr)
            self.update_signal.emit(addr, True)
            t = Thread(target=self.receive, args=(addr, client))
            self.threads.append(t)
            t.start()

    self.removeAllClients()
    self.server.close()
```

```
def removeClient(self, addr, client):
    # find closed client index
    idx = -1
    for k, v in enumerate(self.clients):
        if v == client:
            idx = k
            break

    client.close()
    self.ip.remove(addr)
    self.clients.remove(client)

    del(self.threads[idx])
    self.update_signal.emit(addr, False)
    self.resourceInfo()
```

접속자 정보

	ip	port
1	127.0.0.1	51354
2	127.0.0.1	51355

여러 클라이언트 연결 / 해제 상태 확인

# 프로그램 구조

```
class CWidget(QWidget):
    def __init__(self):
        super().__init__()
        self.random_word = None
        self.coordinates = []

        self.s = server.ServerSocket(self)
        self.drawing_canvas = DrawingCanvas(self)
        self.initUI()

    def initUI(self):
        self.setWindowTitle('서버')

        ipbox = QHBoxLayout()

        gb = QGroupBox('서버 설정')
        ipbox.addWidget(gb)

        box = QHBoxLayout()

        label = QLabel('Server IP : 127.0.0.1')
        box.addWidget(label)

        label = QLabel('Server Port : 1234')
        box.addWidget(label)

        gb.setLayout(box)

        infobox = QHBoxLayout()
        gb = QGroupBox('접속자 정보')
        infobox.addWidget(gb)

        box = QHBoxLayout()

        self.guest = QTableWidget()
        self.guest.setColumnCount(2)
        self.guest.setHorizontalHeaderItem(0, QTableWidgetItem('ip'))
        self.guest.setHorizontalHeaderItem(1, QTableWidgetItem('port'))
```

serverWindow.py

```
class CWidget(QWidget):
    def __init__(self):
        super().__init__()

        self.c = client.ClientSocket(self)
        self.initUI()

    def __del__(self):
        self.c.stop()

    def initUI(self):
        self.setWindowTitle('클라이언트')

        # 클라이언트 설정 부분
        ipbox = QHBoxLayout()

        gb = QGroupBox('서버 설정')
        ipbox.addWidget(gb)

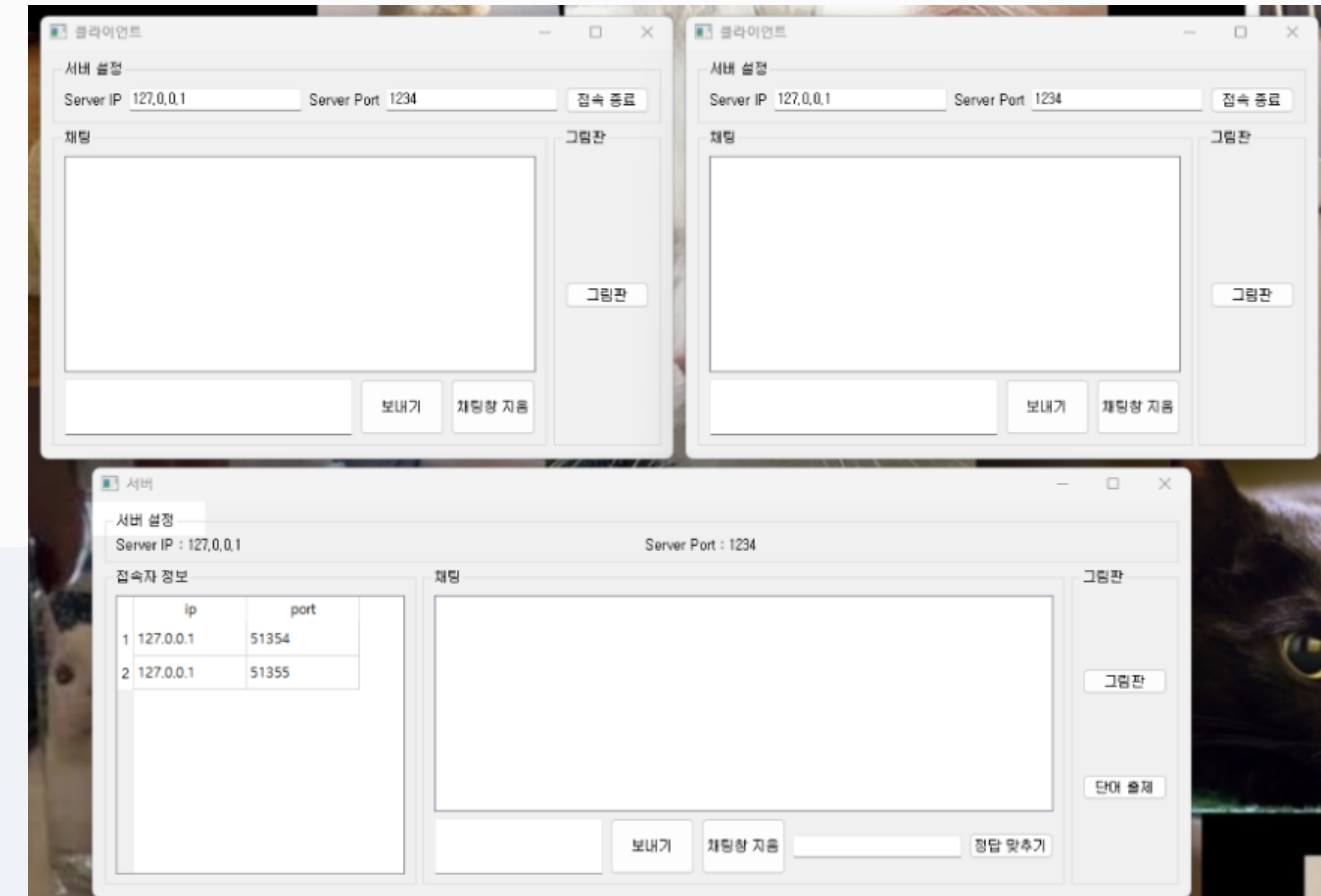
        box = QHBoxLayout()

        label = QLabel('Server IP')
        self.ip = QLineEdit()
        box.addWidget(label)
        box.addWidget(self.ip)

        label = QLabel('Server Port')
        self.port = QLineEdit()
        box.addWidget(label)
        box.addWidget(self.port)

        self.btn = QPushButton('접속')
        self.btn.clicked.connect(self.connectClicked)
        box.addWidget(self.btn)
```

clientWindow.py



PYQT를 사용하여  
전체적인 GUI 제작

# 프로그램 구조

```
def displayRandomWord(self):
    if (self.random_word == None):
        words = ["사과", "바나나", "오렌지", "포도", "체리", "딸기", "망고", "꽃", "달", "나무", "집"]
        self.random_word = random.choice(words)

        self.updateMsg("게임을 시작합니다.")
        self.updateMsg("단어가 전송되었습니다. 상대방이 전송한 그림을 보고 맞춰주세요.")
        self.updateMsg("그림판 버튼을 누르면 그림을 볼 수 있습니다.")

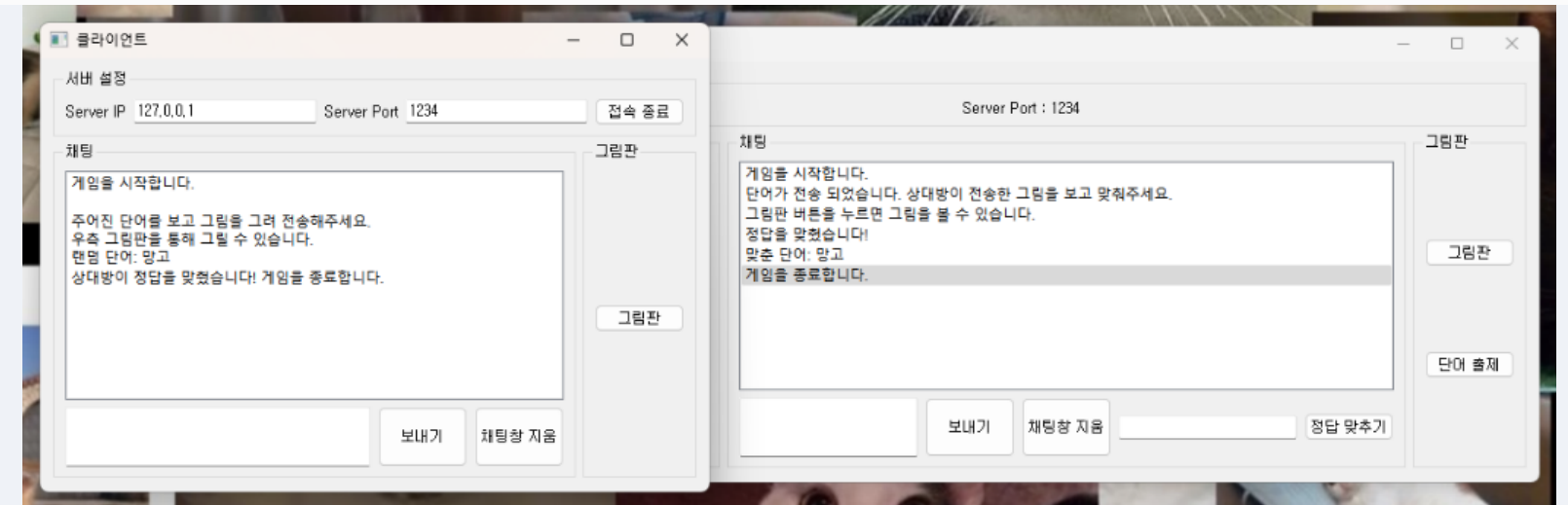
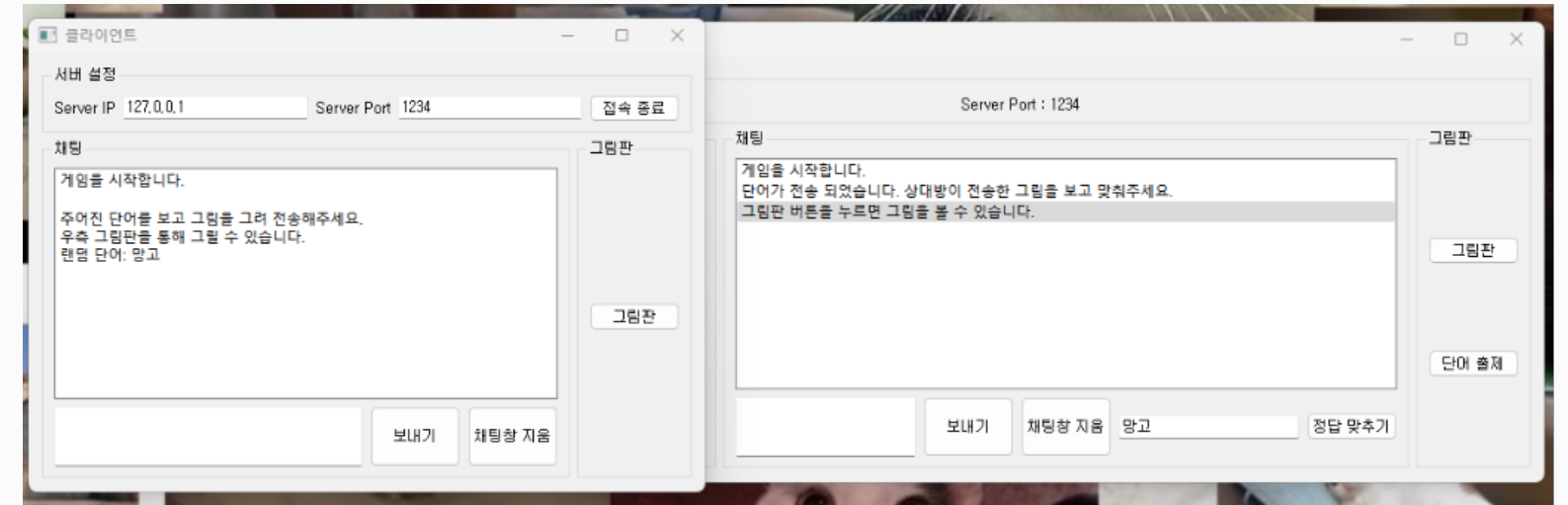
        message1 = f"게임을 시작합니다.\n"
        message2 = f"주어진 단어를 보고 그림을 그려 전송해주세요.\n"
        message3 = f"우측 그림판을 통해 그릴 수 있습니다.\n"
        message4 = f"랜덤 단어: {self.random_word}"

        self.s.send(message1)
        self.s.send(message2)
        self.s.send(message3)
        self.s.send(message4)

    else:
        self.updateMsg("이미 진행 중 입니다.")

def guessWord(self):
    if (self.random_word != None):
        guessed_word = self.word_input.text()
        if guessed_word.lower() == self.random_word.lower():
            self.updateMsg(f"정답을 맞췄습니다!")
            self.updateMsg(f"맞춘 단어: {guessed_word}")
            self.updateMsg(f"게임을 종료합니다.")
            self.word_input.clear()
            self.random_word = None

            message = f"상대방이 정답을 맞췄습니다! 게임을 종료합니다."
            self.s.send(message)
        else:
            self.updateMsg("틀렸습니다. 다시 시도하세요.")
            self.word_input.clear()
    else:
        self.updateMsg("랜덤 단어를 먼저 출제하세요.")
        self.word_input.clear()
```



serveWindow.py

단어 출제 & 맞추기  
서버 파일에 제작

# 프로그램 구조

```
def sendQuiz(self, word):
    try:
        self.quizClient = self.clients[random.randint(0, len(self.clients)-1)]
        self.quizWord = word
        for c in self.clients:
            if c == self.quizClient:
                msg = f"게임을 시작합니다.\n"
                c.send(msg.encode())
                msg = f"주어진 단어를 보고 그림을 그려 전송해주세요.\n"
                c.send(msg.encode())
                msg = f"우측 그림판을 통해 그릴 수 있습니다.\n"
                c.send(msg.encode())
                msg = f"랜덤 단어: {word}\n"
                c.send(msg.encode())
            else:
                msg = f"게임을 시작합니다.\n"
                c.send(msg.encode())
                msg = f"단어가 Client[{str(self.quizClient.getsockname()[1])}]유저에게 전송 되었습니다. Client[{str(self.quizClient.getsockname()[1])}]유저가 전송한 그림을 보고 맞춰주세요.\n"
                c.send(msg.encode())
                msg = f"그림이 도착한 후 그림판 버튼을 누르면 그림을 볼 수 있습니다.\n"
                c.send(msg.encode())

        self.quizing = True
    except Exception as e:
        print('sendQuizSend() Error : ', e)
```

```
def displayRandomWord(self):
    if (self.random_word == None):
        words = ["사과", "바나나", "오렌지", "포도", "체리", "딸기", "망고", "꽃", "달", "나무", "집" ]
        self.random_word = random.choice(words)

    self.updateMsg("게임을 시작합니다.")
    self.updateMsg("단어가 전송 되었습니다. 상대방이 전송한 그림을 보고 맞춰주세요.")
    self.updateMsg("그림이 도착한 후 그림판 버튼을 누르면 그림을 볼 수 있습니다.")

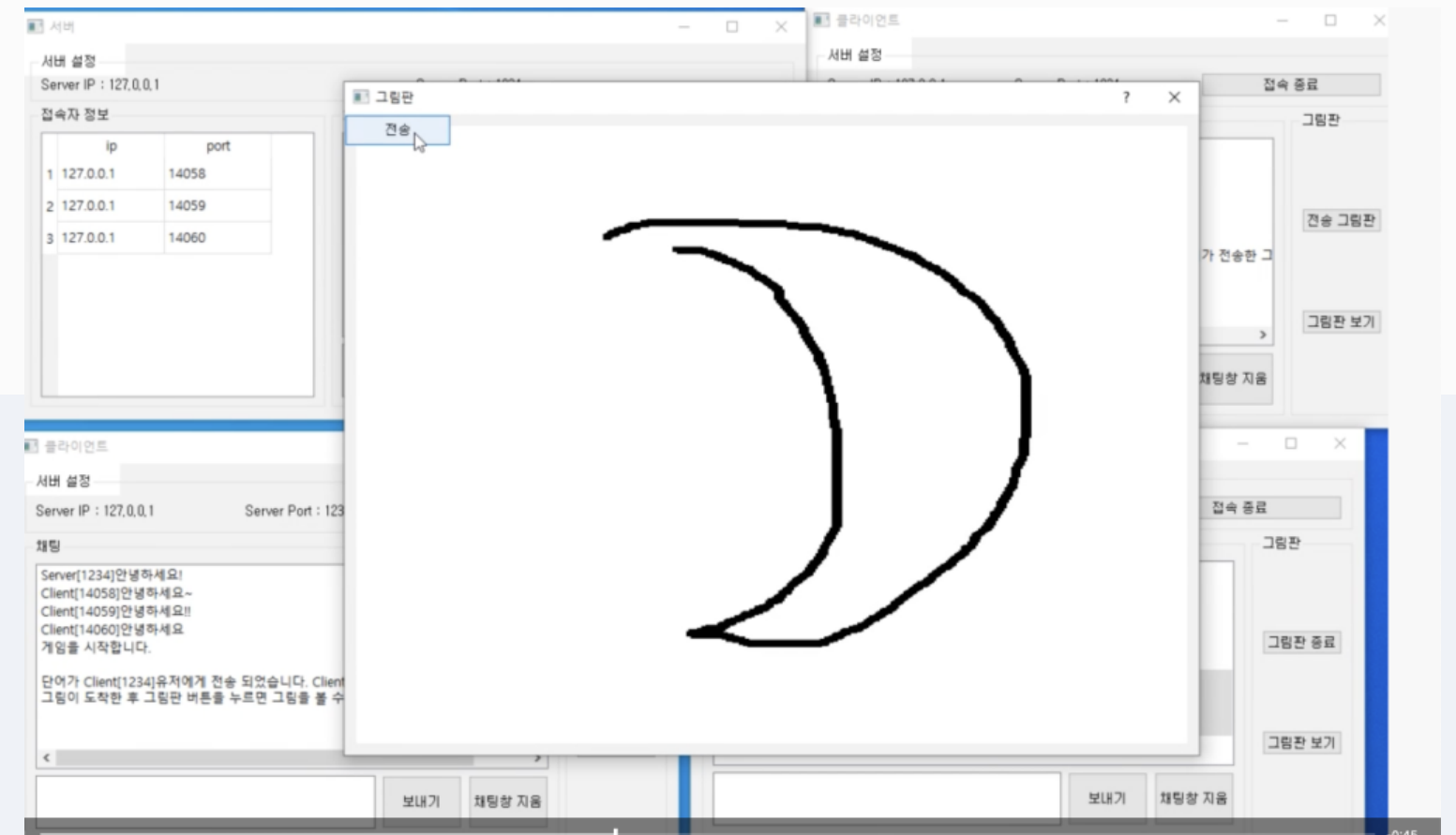
    self.s.sendQuiz(self.random_word)
```

## 단어 출제 & 그릴 사람 지목

# 프로그램 구조

```
def mousePressEvent(self, e):  
    if e.button() == Qt.LeftButton:  
        self.drawing = True  
        self.last_point = e.pos()  
  
def mouseMoveEvent(self, e):  
    if (e.buttons() & Qt.LeftButton) & self.drawing:  
        painter = QPainter(self.image)  
        painter.setPen(QPen(self.brush_color, self.brush_size,  
                             painter.drawLine(self.last_point, e.pos())  
        self.last_point = e.pos()  
        self.update()  
  
def mouseReleaseEvent(self, e):  
    if e.button() == Qt.LeftButton:  
        self.drawing = False
```

ClientWindow.py



마우스를 이용한 그림판에 그리  
기



# 프로그램 구조

```
def get_image_data(self):  
    byte_array = QByteArray()  
    buffer = QBuffer(byte_array)  
    buffer.open(QIODevice.WriteOnly)  
    self.image.save(buffer, "PNG")  
    return byte_array.data()
```

ClientWindow.py

```
Sent: b'\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\x00\x01\x90\x00\x00\x01\x90  
\xda@\x14@\xd1\xba\xea\xff\xff}\x88\x94\xa6\\\x1c\x03\x0e\xcc\x1e\xaf\x15\x  
\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80  
c8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x  
0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x  
\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04  
\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb  
0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x  
80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\x  
\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022  
\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C  
8\xbe\x1d\xaf\x1c\x82uhjE\x8b\xa5;\xff\xb1\xaebd&\xac\xe3\xba\x1c\xaf\xd4\x  
1:.\xe3\x159\x82\x05d\x08\xd6Q\x9c\x9d\x07\x8dW\x14\t\x16\x90!X@\x86`\x1d\x  
f\x16XLF\xbb\x0e\xc4\x02\x8b:\xc1\x022\x04\x0b\xc8\x10\xaciY`1\x1f\xc1:\n\x  
7f\xde\xfd\x02\xe0?_\xef\xc9pC\x06gLX@\x86`1.\x0fM\xe5\x8c`\x01\x19\x82\x05  
\x16\x0cE\xbb\x8I\xad\x18\x8d`q\x9dZ1 0k\xe0\x9cT1,\x13\x16\xff\xac\xdc\x14
```

그림 데이터를 PNG 이미지로 저  
장 및 바이트 데이터 반환

# 프로그램 구조

```
def receive(self, addr, client):
    try:
        while True:
            data = client.recv(4096)
            if not data:
                break
            print(data)
            if data.startswith(b'\x89PNG\r\n\x1a\n') | b:
                self.parent.handle_drawing_coordinates

            else:
                msg = data.decode('utf-8')
                self.recv_signal.emit(msg)

    except Exception as e:
        print(f"Error receiving data from {addr}: {e}")
    finally:
        client.close()
        self.removeClient(addr, client)
```

serverWindow

```
b'\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\x00\x01\x90\x00\x00\x01\x90\x08\x02\x00\x00\x00\x0f\xdd\x
x14@\xd1\xba\xea\xff\xff}\x88\x94\xa6\\\x1c\x03\x0e\xcc\x1e\xaf\xf5\x18E\n\x0f\xce\xd6\x99\x831\x
022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10
, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc
\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x
10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\
c1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc
\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0
\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\
c8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\
0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0b\xc8\x10, C\xbb\x80\x0c\xc1\x022\x04\x0
x1d\xaf`\x1c\x82uhjE\x8b\xa5;\xff\xb1\xaebd&\xac\xe3\xba\x1c\xaf\xd4\x8a\xc1\t\xd6A\xa9\x15E\x82\x
3\x159\x82\x05d\x08\xd6Q\x9c\x9d\x07\x8dW\x14\t\x16\x90!X@\x86`\x1d\x82\xf3 s\x10, C\xbb\x80\x0c\x
LF\xbb\x0e\xc4\x02\x8b:\xc1\x022\x04\x0b\xc8\x10\xaciY`1\x1f\xc1:\n\x0b,& X@\x86`\x01\x19\x82\x05d\
```

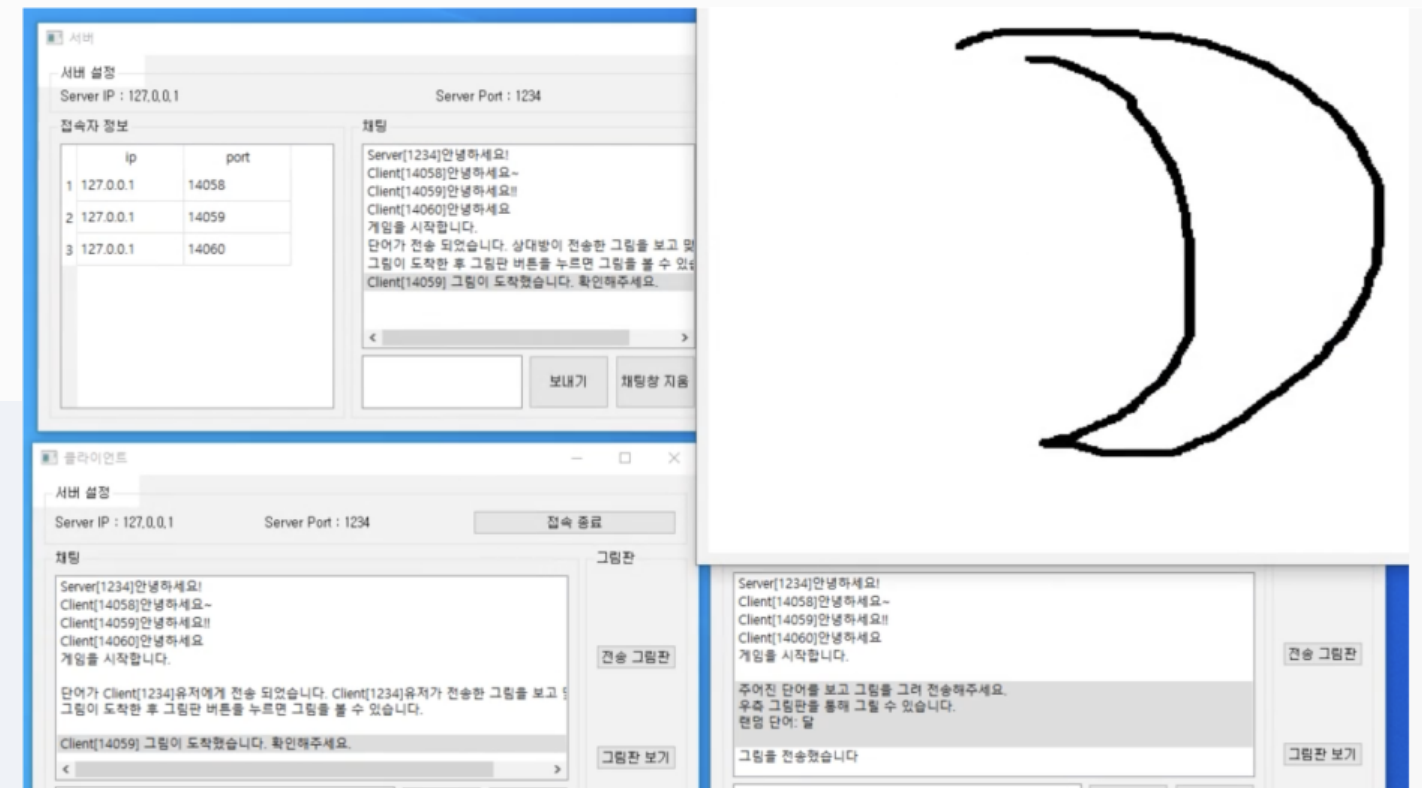
클라이언트에서 넘어온 데이터  
와 동일



# 프로그램 구조

```
def update_drawing(self, image_data):  
    image = QImage.fromData(image_data)  
    self.image = image  
    self.update()
```

window.py



클라이언트에서 받은 바이트 데이터  
를 서버에 전달

# 프로그램 구조

```
def sendMsg(self):
    if not self.s.bListen:
        self.sendmsg.clear()
        return
    sendmsg = "Server[" + str(self.s.server.getsockname()[1]) + "]" + self.sendmsg.text()
    self.updateMsg(sendmsg)
    print(sendmsg)
    self.s.sendmsg(sendmsg)
    if self.sendmsg.text() and self.random_word:
        if self.random_word in self.sendmsg.text():
            self.updateMsg(f"서버가 정답을 맞췄습니다!")
            self.updateMsg(f"맞춘 단어: {self.random_word}")
            self.updateMsg(f"게임을 종료합니다.")
            sendmsg = f"Server[{str(self.s.server.getsockname()[1])}]님이 정답을 맞췄습니다! : 맞춘단어:{self.random_word}\n"
            self.s.sendmsg(sendmsg)
            self.random_word = None
            self.s.quizing=False
            self.s.quizWord=None
            self.s.quizClient=None
    self.sendmsg.clear()
```

정답맞추기 - 서버  
serverWindow.py

# 프로그램 구조

```
else:
    msg = data.decode('utf-8')
    if self.quizing:
        if self.quizWord in msg:
            self.quiz_signal.emit(msg)
            self.quizCorrect(client)
        else:
            self.msg_signal.emit(msg)
            self.sendmsg(msg, client)
```

receive method

```
def quizCorrect(self, client=None):
    try:
        for c in self.clients:
            if c == client:
                msg = f"축하합니다. 정답을 맞췄습니다! : 맞춘 단어:{self.quizWord}.\n"
                c.send(msg.encode())
                msg = f"게임을 종료합니다.\n"
                c.send(msg.encode())
            else:
                msg = f"Client[{str(client.getsockname()[1])}]님이 정답을 맞췄습니다! : 맞춘 단어:{self.quizWord}.\n"
                c.send(msg.encode())
                msg = f"게임을 종료합니다.\n"
                c.send(msg.encode())
        self.quizing=False
        self.quizWord=None
        self.quizClient=None
    except Exception as e:
        print('quizCorrectSend() Error : ', e)
```

quizCorrect method

## 정답맞추기 - 클라이언트

# 프로그램 구조

```
def receive(self, addr, client):
    try:
        while True:
            data = client.recv(4096)
            if not data:
                break
            print(data)
            print("receiving")
            if data.startswith(b'\x89PNG\r\n\x1a\n') or data.startswith(b'\xFF\xD8\xFF\xE0') or data.startswith(b'\xFF\xD8\xFF\xE1'):
                self.parent.handle_drawing_coordinates(data)
                self.sendIMG(data, client)
            else:
                msg = data.decode('utf-8')
                if self.quizing:
                    if self.quizWord in msg:
                        self.quiz_signal.emit(msg)
                        self.quizCorrect(client)
                    else:
                        self.msg_signal.emit(msg)
                        self.sendmsg(msg, client)
                else:
                    self.msg_signal.emit(msg)
                    self.sendmsg(msg, client)
        except Exception as e:
            print(f"Error receiving data from {addr}: {e}")
        finally:
            client.close()
            self.removeClient(addr, client)
```

```
def sendmsg(self, msg, client=None):
    try:
        if client:
            for c in self.clients:
                if c != client:
                    c.send(msg.encode())
        else:
            for c in self.clients:
                c.send(msg.encode())
    except Exception as e:
        print('Send() Error : ', e)
```

```
def sendIMG(self, msg, client=None):
    try:
        if client:
            for c in self.clients:
                if c != client:
                    c.sendall(msg)
        else:
            for c in self.clients:
                c.sendall(msg)
    except Exception as e:
        print('imgSend() Error : ', e)
```

채팅방 에코