Team. RO:BIT | C++&qt day4

18기 변정욱

RO:BIT
ROBOT SPORT GAME TEAM

INDEX

통신

통신

통신(Communication)의 정의

- 두 개 이상의 개체가 데이터를 주고받는 과정
- 정보 전달 + 의미 해석이 동시에 이루어져야 함

기본 요소

- 송신기(Transmitter)
- 수신기(Receiver)
- 전송 매체(Channel)
- 프로토콜(Protocol, 규칙)

통신

통신 방식 분류

- **전송 방향에 따른 분류**
  단방향(Simplex) : TV 방송처럼 한쪽만 송신 → 수신만 가능
  반이중(Half Duplex) : 무전기처럼 양방향 가능하지만 동시에 불가
  전이중(Full Duplex) : 전화처럼 동시에 송수신 가능
- **회선 구성 방식**
  회선 교환(Circuit Switching) : 통화선 연결 후 통신 (전화망)
  패킷 교환(Packet Switching) : 데이터를 작은 패킷 단위로 나누어 전송 (인터넷)

통신

통신 프로토콜

- **정의** : 통신을 원활히 하기 위해 정해진 규칙과 약속
- 주요 기능

  주소 지정(Addressing) : 누구와 통신하는지 구분
  에러 제어(Error Control) : 오류 검출/복구
  흐름 제어(Flow Control) : 송수신 속도 조절
  동기화(Synchronization) : 데이터 순서/타이밍 맞추기

통신

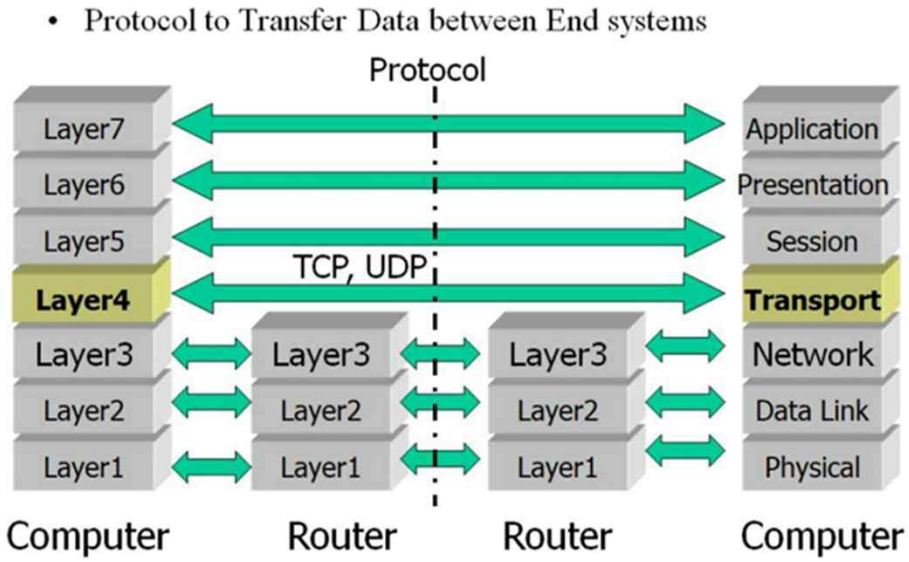
- 유선(Wired) : LAN 케이블(Ethernet), 광케이블 등 → 안정적, 고속
- 무선(Wireless) : Wi-Fi, Bluetooth, 5G 등 → 이동성, 편의성

- 선택 기준 : 속도, 안정성, 범위, 비용

UDP

UDP

- OSI 7계층 중 전송 계층(4계층)에 속하는 통신
- 규약비연결성 & 비신뢰성 특징을 가지며, 빠른 전송을 위해 설계됨



UDP

- **OSI 7계층 모델**
  물리 (Physical) – 신호 전송 (케이블, Wi-Fi)
  데이터 링크 (Data Link) – MAC주소 기반 프레임 전송
  네트워크 (Network) – IP 주소 기반 라우팅
  전송 (Transport) – 종단 간 데이터 전달 (→ TCP / **UDP**)
  세션 (Session) – 연결 관리
  표현 (Presentation) – 데이터 변환, 압축, 암호화
  응용 (Application) – 실제 서비스 (HTTP, FTP, 영상 앱 등)

UDP는 **전송 계층 프로토콜**로서

- 아래(IP 계층)에서 목적지까지의 전송을 지원받고,
- 위(응용 계층)에서 내려온 데이터를 빠르게 보내주는 역할을 함.

UDP

- 데이터 전송 단위: 데이터그램 (Datagram)
- 연결 과정 없음 → 수신자가 준비되어 있지 않아도 송신 가능
- 신뢰성 보장 X → 오류 검출은 Checksum으로 최소한만 제공
- 고정 헤더 8바이트 (출발지 포트, 목적지 포트, 길이, 체크섬)
- 속도는 빠르지만, 데이터 순서 보장·재전송 없음
- 실시간성이 중요한 경우 적합 (영상 스트리밍, 온라인 게임, 센서 데이터 전송 등)

UDP

- 로봇 영상 데이터: 초당 15~30fps, 대용량 이미지 전송 → 빠른 전송 필요
- 손실 발생해도 다음 프레임으로 대체 가능 → UDP 적합
- 상태 메시지, 명령어처럼 중요한 데이터 → UDP + CRC8 또는 TCP로 보완

UDP

- **Qudpsocket** 라이브러리 사용

```
QHostAddress ROBOT_IP = HostAddress("192.168.188.100");
QHostAddress OPERATOR_IP = HostAddress("192.168.188.253");

uint16_t TEXT_PORT = 9999;
```

- **UDP 포트 번호(0~65535)는 다음과 같이 구분된다.**
- **0~1023번**: Well-known Port (시스템에서 주요 서비스에 예약됨 → 예: HTTP 80, FTP 21)
- **1024~49151번**: Registered Port (일반 프로그램이나 사용자 애플리케이션이 등록하여 사용)
- **49152~65535번**: Dynamic/Private Port (임시 연결이나 클라이언트 측에서 자동 할당)
- 따라서 **실제 개발 시에는 1024번 이상의 포트 번호를 사용하는 것이 안전하다.**

UDP

- IP, port를 통해 들어온 정보를 연결시킬 소켓을 선언

```
text_socket = new QUdpSocket(this);
```

```
if(text_socket->bind(OPERATOR_IP, TEXT_PORT, QUdpSocket::ShareAddress))
{
  connect(text_socket, SIGNAL(readyRead()), this, SLOT(udp_read()));
}
```

UDP

- Qudpsocket에서 제공하는 데이터그램 패킷의 자료형은 QBytearray로 크기가 1인 바이트 배열의 형태
- QBytearray형의 변수를 선언하고 바이트 배열의 크기를 먼저 지정한 뒤 socket으로 전달된 데이터를 readDatagram 함수로 받음

```
void MainWindow::udp_read()
{
  QByteArray buffer;
  buffer.resize(text_socket->pendingDatagramSize());
  text_socket->readDatagram(buffer.data(), buffer.size(), &ROBOT_IP,
&TEXT_PORT);
}
```

UDP

- 반대로 데이터를 보내기 위해서는 ByteArray 형태의 데이터를 목적지 IP, 목적지 port를 향해 writeDatagram 함수 사용

```
void MainWindow::udp_write(QByteArray text, uint16_t port,  QUdpSocket &socket)
{
  QByteArray packet;
  packet.push_back(text);
  socket.writeDatagram(packet, ROBOT_IP, port);
}
```

과제 및 프로젝트

과제 1

- 2인 1조로 서로 대화를 주고 받는 문자(카카오톡) 구현

- 영상 및 코드 git에 제출

과제 2

RO:BIT
ROBOT SPORT GAME TEAM

C++&QT 프로젝트

RO:BIT
ROBOT SPORT GAME TEAM

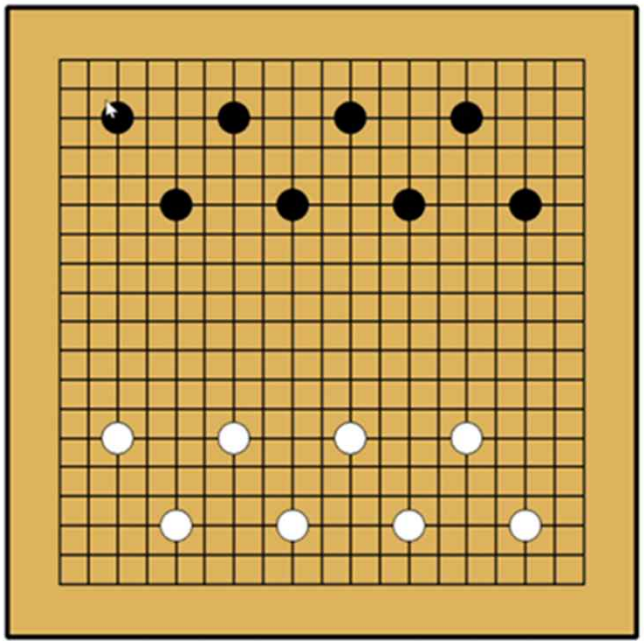
- 2인 1조로 C++, QT, UDP를 사용한 게임 제작
- 9월 10일 수요일 오후 10시 발표예정(ppt도 만들어 오기)
- 발표시간 10분 이내(시연 포함)

C++&QT 프로젝트

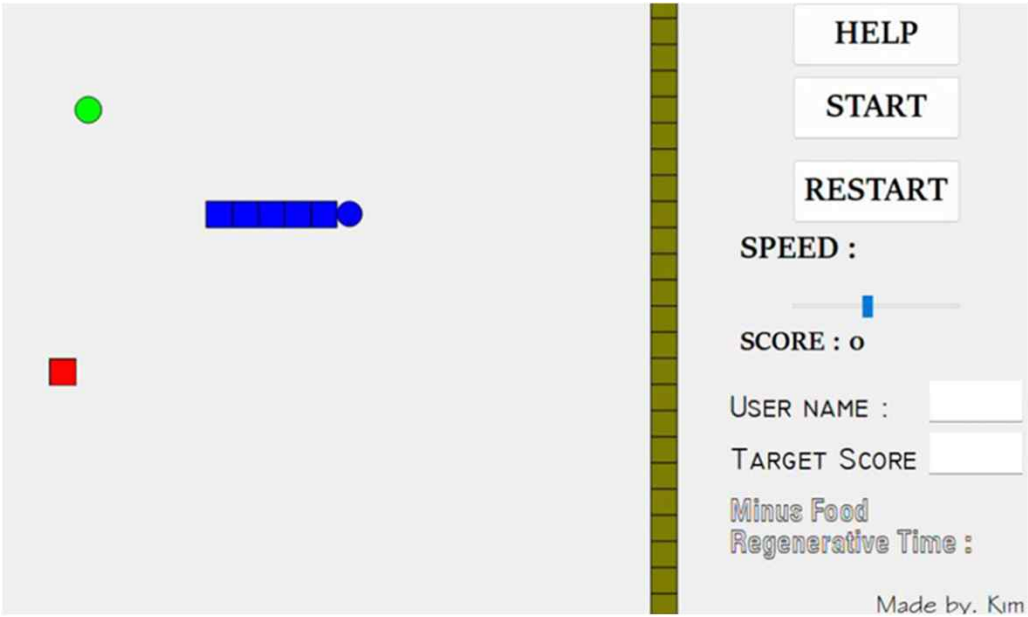
세부 평가 계획

- 발표 태도(20)
- 코드: 클래스 활용(20), 알고리즘 구현(20), 가독성(10), 예외 처리(10)
- 창의성(20)
- 동료평가 예정

C++&QT 프로젝트




restart
black turn
black
8
white
8


HELP
START
RESTART
SPEED :
SCORE : 0
USER NAME :
TARGET SCORE
Minus Food
Regenerative Time :
Made by. Kim

Thank you