LG 부트캠프 10기

프로젝트 결과보고서

# 내가그린그런그림

Linux System반 2팀

이상인 김동언 박성현 박수아

# 목차

# 주제 및 개요

## 다 함께 그리는 게임을 만들어 보자!

### 그림을 그리는 프로그램 개발
- 종이에 펜으로 그림을 그리는 것과 유사한 기능을 목표로 함
  현실에서 그리는 것처럼, 색연필 / 지우개 등의 기능을 지원

### 멀티 플레이 환경 구축
- 혼자 노는 것 뿐만 아니라, 자기가 그린 그림이 무엇인지 맞히는 멀티 플레이를 목표로 개발

### 보드를 이용한 재미있는 게임 컨트롤러 만들기
- 다양한 I/O 장치를 활용하여 사용자에게 재밌고 풍부한 경험 제공
  : 지우개, 색 변경, LED 상태 표시기, ...
- 힌트, 타이머 등의 요소를 추가하여 게임의 완성도 향상

# 개발 목표

## 실시간 터치 입력 기능

- 손가락, 또는 터치펜으로 그림을 그리는 방식으로
  게임 진행 편의성 확보

## 타인 간 멀티 통신 기능

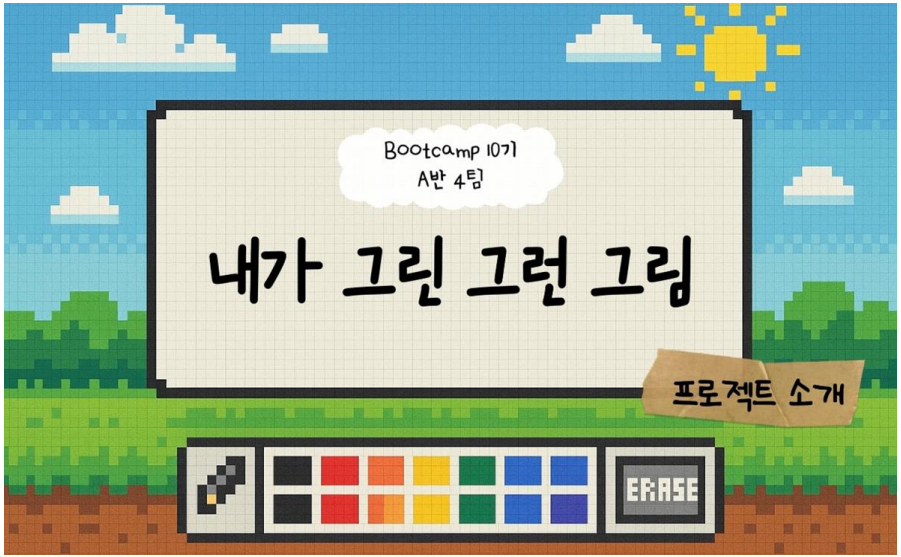
- TCP 통신을 활용하여 실시간으로
  다른 사람과 대전 가능

## 독자적인 코드 개발

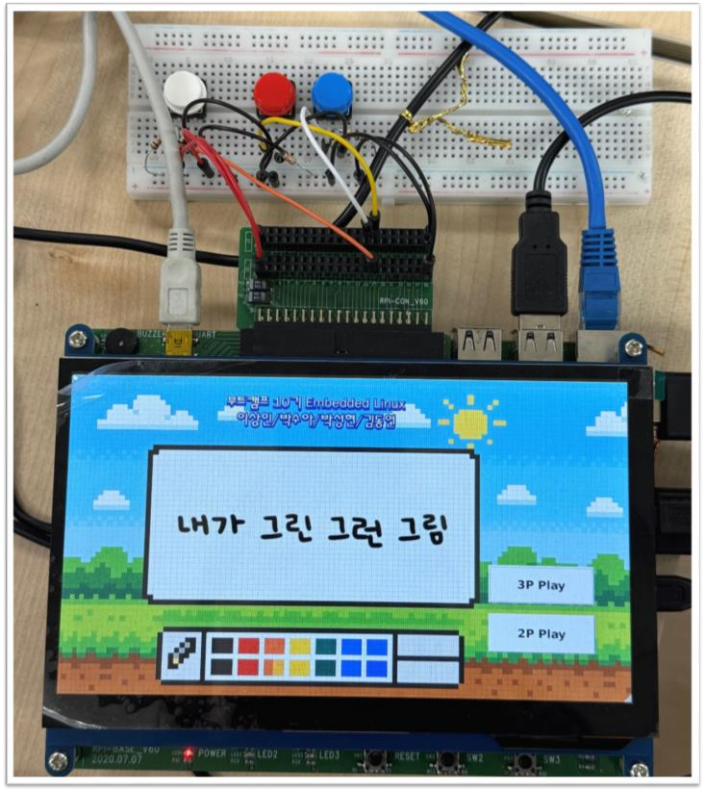
- 패킷 통신 코드를 직접 개발하여
  다양한 이벤트 함수를 구현

## UI 구현

- 키보드, 물리 버튼, LED 등
  게임 정보를 제공, 확인할 수 있는 여러 UI 추가



# 보드 및 게임 로직 설명




SW2 SW3

<버튼 구성>

펜 색깔 변경

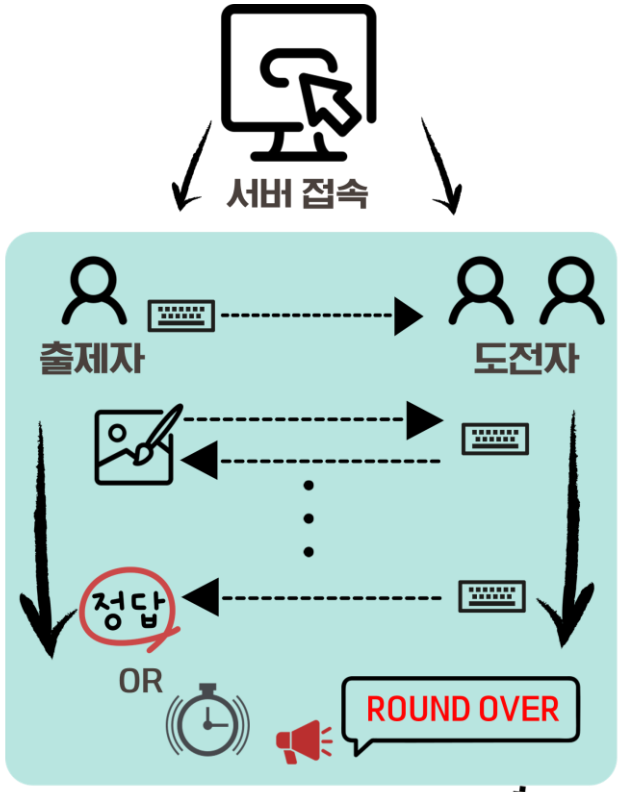
펜촉 굵게

펜촉 얇게

SW2 지우개 선택

SW3 전체 지우개

출제자가 제시어를 입력하면
라운드가 시작됩니다.




답을 맞히면 점수를 얻고, X 4 ROUNDS
그 사람이 출제자가 됩니다.
타임 오버 시 출제자가 점수를 얻습니다.

# 시연 영상

# 핵심 기술

1. 최신 커널 포팅 및 신규 rootfs 구축
2. 신규 GPIO 포팅과 Custom ioctl 기반 주변 장치 제어
3. 커널 자원 활용
4. 다중 클라이언트 지원
5. TCP 통신을 통한 멀티 플레이
6. 상황별 이벤트 처리 및 게임 인터페이스 구현

# Kernel 포팅 및 최적화

- **신규 BCM2837–default config 적용 후 (Raspberry pie 3 B), kernel 최적화**
  - 최신 커널 version 6.12 로 upgrade
  - Filesystem ext2 (ext4) 및 nfs, tmpfs 만 지원 가능하게 제한
  - 불필요한 Device driver 제거
    - ex) USB driver, HID driver, sound driver, HDMI driver 등 필요한 최소한의 driver 만 포함
  - 사용하지 않는 Crypto API 제외 (ext2 dependency, rootfs, network 등 필수적인 부분만 남기는 방향)
  - 추가 custom device deriver (GPIO 제어) 포팅
  - Kernel Image size
    - Default BCM2835 => 약 35 MB
    - Wt2837 => 약 25MB
    - 현재 => 약 16MB
    - 부팅 속도 약 3초 이상 개선.

# RFS

- RFS
  - Sound 재생을 위한 Alsa-util 및 QT5 를 위한 QT5Lib, font 정도만 추가.
  - 이후 multi play 지원을 위한 board 각 개별에 대한 정보를 intialize 해주기 위해, data partition 에 script 를 trigger 하도록 설정.
  - SD 카드에 실행 binary 부팅 후 자동 실행.

Init.d/S50local

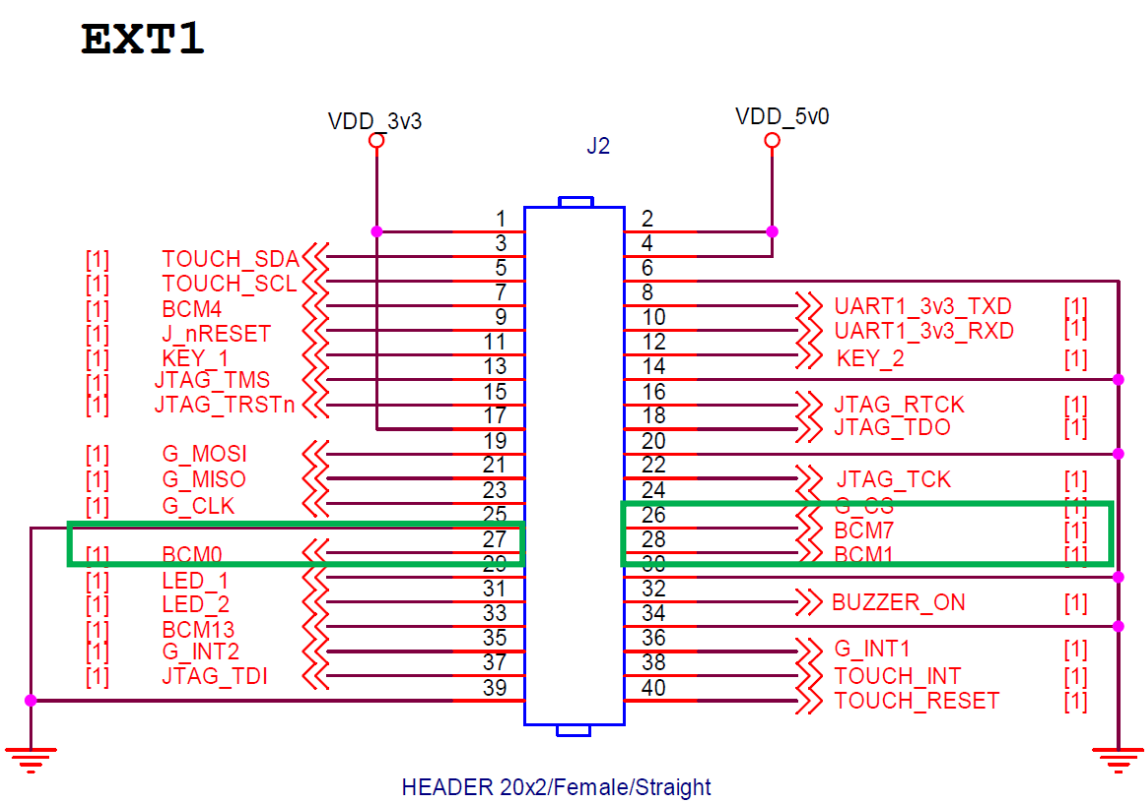
```
# Wait for ethernet device to link
for count in 1 2 3 4 5
do
        link=`cat /sys/class/net/eth0/carrier`
        if [ "$link" = "1" ]; then break; fi
        echo "Wait for ethernet device to link - $count"
        sleep 1
done

mkdir -p /mnt/nfs
mkdir -p /mnt/sd

# mount -t nfs -o nolock 192.168.10.2:/nfsroot /mnt/nfs

if [ -e "/dev/mmcblk0" ]; then

        if [ -e "/dev/mmcblk0p2" ]; then
                mount -t ext2 /dev/mmcblk0p2 /mnt/sd
        else
                echo "there's no /dev/mmcblk0p2"
        fi
else
        echo "SD card is not supported"
fi
```

Init.d/rcS

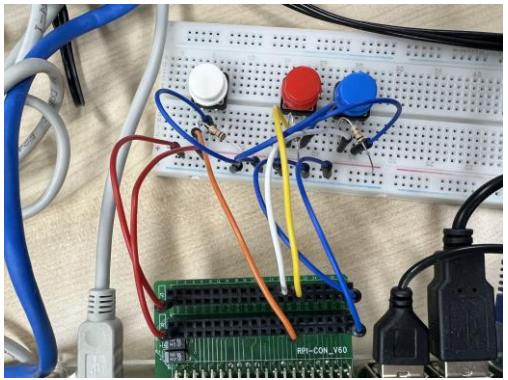
Data 영역에 개별적으로 저장되어 있는 script 실행

```

INIT_DIR="/mnt/sd/initSystem"

if [ -d "$INIT_DIR" ]; then
    for i in $INIT_DIR/S??* ; do
        # Ignore dangling symlinks (if any)
        [ ! -f "$i" ] && continue
        case "$i" in
            *.sh)
                # Source shell script for speed
                (
                    . $i
                )
                ;;
        esac
    done
else
    echo "There's no $INIT_DIR."
fi
```

# 디바이스 제어 - GPIO control




- 기존 porting된 btn 외 EXT_converter의 BCM0,1,7를 GPIO로 추가 사용
- 기존 kernel gpio base 0 -> 신규 kernel gpio base 512로 변경됨에 따라 GPIO NUM도 변경

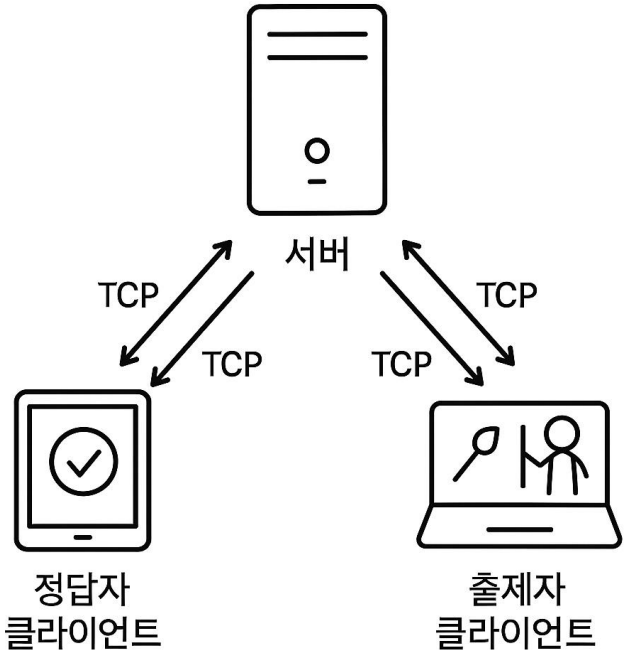


# 디바이스 제어 - interrupt control

- **Custom ioctl 구현 - 특정 LED 패턴 및 GPIO btn 제어**
  - iowrite32 - 특정 register의 bit 값 설정으로 LED 제어
- **각 GPIO 신호는 interrupt로 구현 - 배열로 gpio 번호 관리**
  - request_irq(irq_keys[i], key_clear_isr, IRQF_TRIGGER_FALLING, "key_clear", (void *)(long)i);
- **Work_queue 사용으로 여러 gpio interrupt 신호 scheduling 용이**
  - Btn 입력의 경우, work_queue 사용으로 버튼 입력 순서 보장
  - Kernel worker thread의 Background 실행으로 kernel에서의 여러 입력 비동기 처리 가능
- **Atomic 변수 사용**
  - LED interrupt를 Atomic 변수를 통해 제어
- **Kernel timer 사용**
  - Jiffies timer - kernel 영역에서의 일관적인 timer 제어 가능
  - mod_timer – 해당 timer에 대해 callback 함수 등록
- **신규 kernel device tree 확인 시, BCM 0, 1, 7은 i2c 혹은 spi로 사용되지 않음 -> device tree 수정 필요 X**

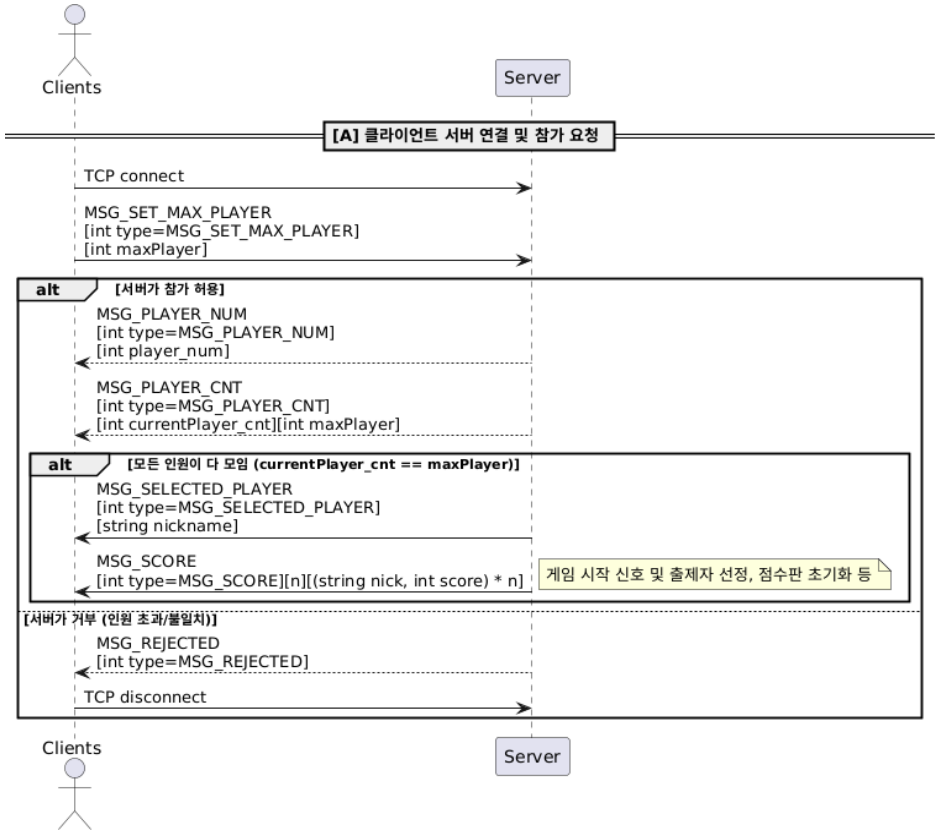
# TCP 프로토콜 기반 실시간 네트워크 통신

- **신뢰성 보장**
  - 12가지의 패킷 타입을 각각 목적에 맞게 사용하며, 각 패킷의 도착 순서와 무결성이 매우 중요.
- **데이터 무결성**
  - 게임의 점수, 정답, 그림 좌표 등 핵심 데이터가 손상되거나 누락되면 게임 진행에 치명적.
  - TCP는 데이터가 정확하게, 순서대로, 완전하게 도착하도록 보장
- **연결 관리**
  - 연결 상태를 관리하여 클라이언트 접속/종료 등 2P/3P 플레이 구성에 효과적으로 대응.
- **실시간 패킷 신뢰성**
  - 그림 좌표는 실시간으로 여러 사용자에게 동기화 필요.
  - 패킷 손실이나 순서 뒤바뀜이 발생하면 그림이 깨질 수 있기에, 패킷 신뢰성이 중요

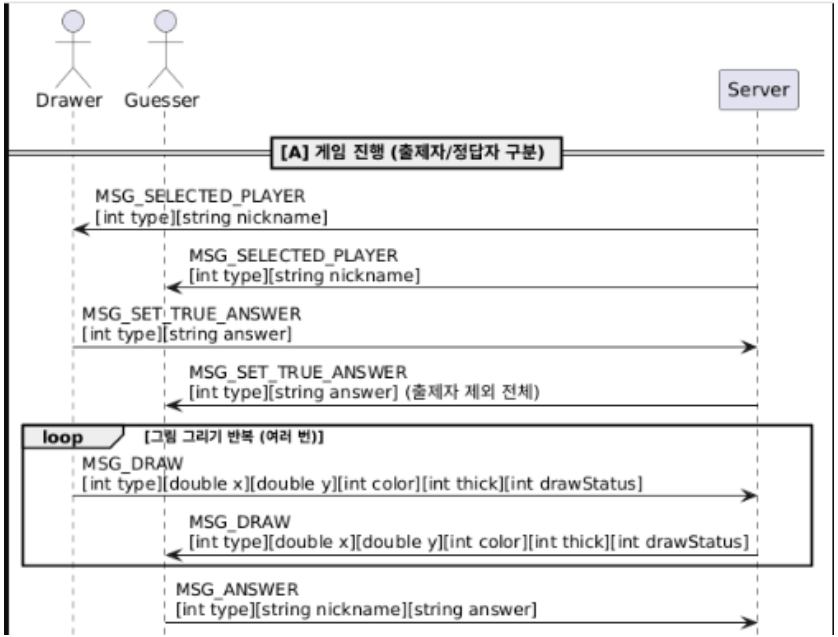


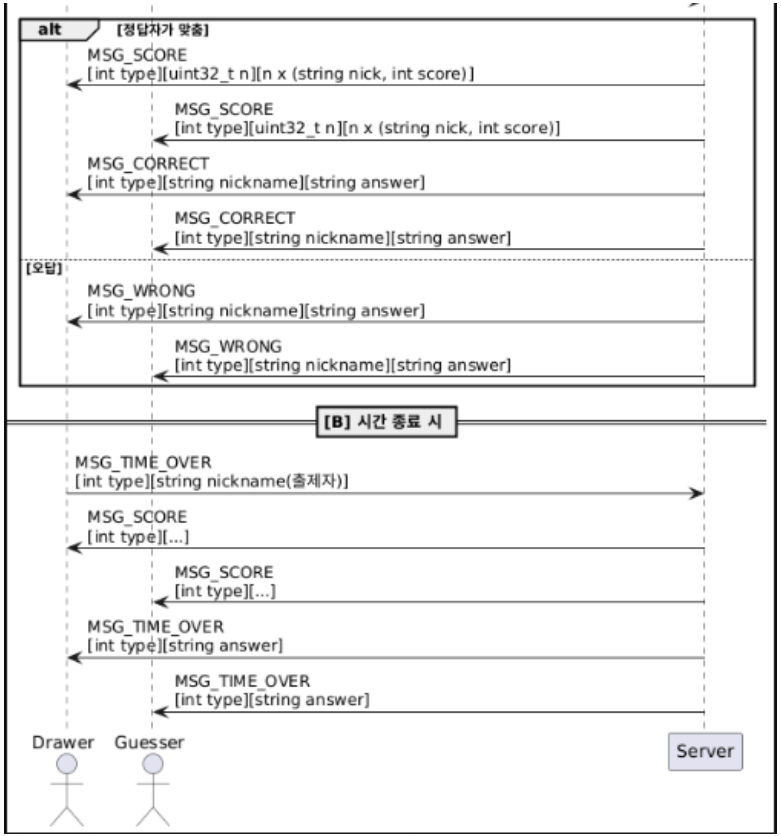
# TCP 기반 네트워킹 Sequence diagram - 서버 접속

- 서버와 각 client는 packet 수신 및 송신용 thraed 생성
- Packet 송신 시, mutex_lock 사용으로 여러 thread의 임계영역 진입 방지
  - 여러 thread가 동시에 send 함수에 접근 시, 송신 packet의 순서가 꼬여 recv 시 정확한 값 수신 불가.



# TCP 기반 네트워킹 Sequence diagram - 게임 플레이



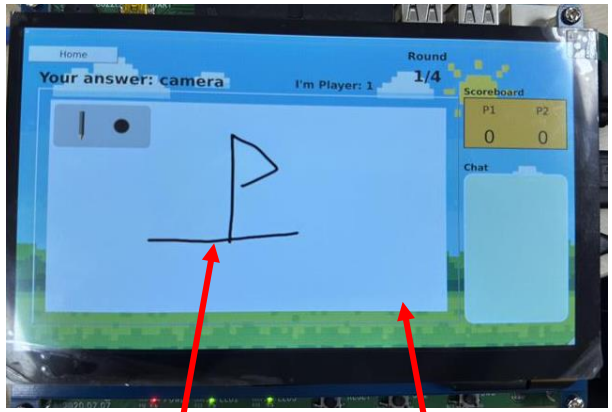



# QT Event Handling 1 : Override

- **기본적인 event에 대한 handling 정의**
  - 터치(touchEvent), 위젯 크기조정(resizeEvent), 그리기(paintEvent)
  - 각 window에서 override하여 다르게 동작

- **touchEvent, paintEvent**
  - 좌표가 업데이트될 때마다 경로를 저장하고 (touch)
  - 실시간으로 화면에 그리도록 함 (paint)
  - 도전자 widget에서는 touch로 그려지지 않음

- **resizeEvent**
  - 따로 생성된 drawing widget을 동적으로 resize



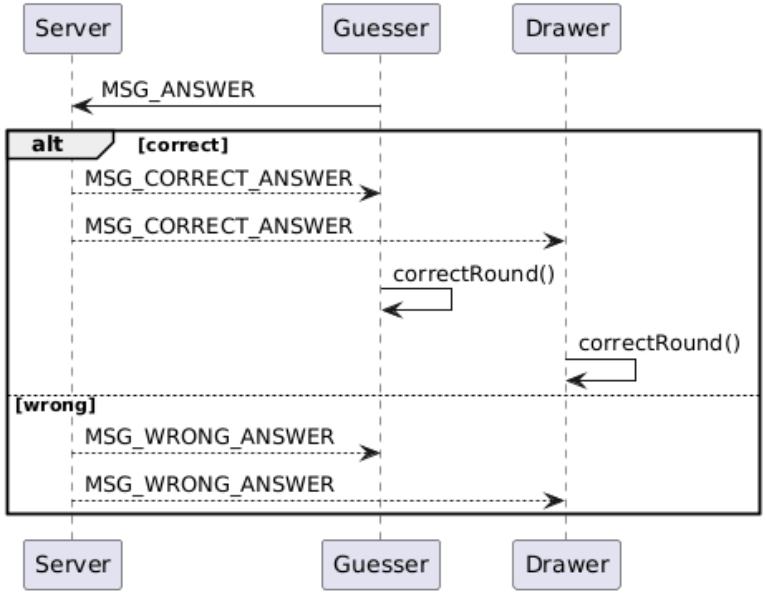

**touch, paint**

**resize**

# QT Event Handling 2 : QT Library

- **Signal, SLOT**
  - 비동기 호출을 가능하게 하는 Qt Library 함수
  - Button에 대한 동작
    (backToMain -> backToMainRequested)

- **invokeMethod**
  - Signal 없이 SLOT을 비동기 호출할 때 사용
  - 서버 <-> 클라이언트 간 패킷을 받았을 때, packet type에 따라 SLOT을 호출
  - 타이머 구현

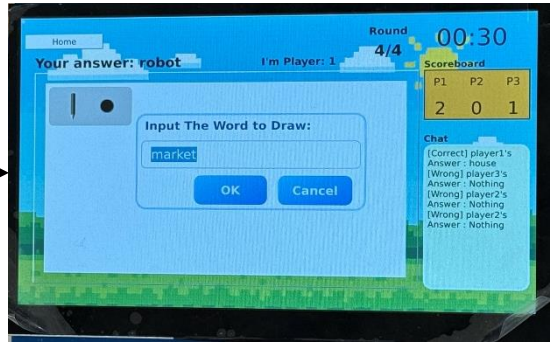


# 핵심 기술: UI별 구성 및 역할 분리

- **첫 라운드**
  **→ 서버에서 랜덤하게 Drawer/Guesser 역할 배정**
- **두번째 라운드 이후**
  **→ 정답을 맞춘 클라이언트 혹은 맞춘 사람이 없다면 기존의 클라이언트가 다음 라운드의 Drawer 배정**

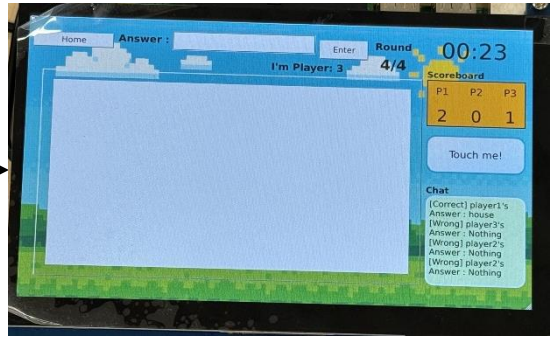

## Main 화면

- 클라이언트가 2인/3인 모드를 선택
- 접속 대기/진행 상태 표시
- 게임 진행 화면으로 진입하는 역할




## Drawer 화면

- Drawer(출제자)가 **단어를 입력**하고 그림을 그림
- 그리기 도구 모듈 제공
- Draw widget에 입력된 좌표를 서버에 송신하고 서버는 Guesser에게 전달
- 점수판,라운드,타이머, 채팅 등 게임 주요 진행 컨트롤




## Guesser 화면

- 그림을 보고 퀴즈 **정답을 맞추는** 화면
- 서버로부터 받은 좌표를 화면에 표시
- 점수판, 라운드. 타이머, 채팅 등 게임 주요 진행 컨트롤

# 부가 기능

- **실시간 채팅 지원** 
  - 모든 클라이언트 간 채팅 메시지 UI 제공
  - 정답과 무관한 소통 가능
- **점수판 실시간 업데이트** 
  - 각 라운드/이벤트 후 점수판 자동 갱신
  - 모든 라운드 종료 후, 클라이언트별 최종 결과 출력
- **게임 접속 관리** 
  - 인원초과/게임방 이미 생성 시 경고 메시지 팝업
  - 플레이어가 퇴장 시 UI 상태 갱신
- **라운드별 타이머 관리** 
  - 라운드별 남은 시간 표시(카운트 다운)
  - 10초 임박시 타이머 색상 변경 및 플래시 처리
- **Guesser 측 힌트 제공** 
  - 10초 임박 시 정답 첫 글자와 문자 길이 힌트 제공
  - 터치/클릭으로 힌트 일시적으로 노출 가능
- **Drawer 단어 입력 다이얼로그** 
  - Drawer에게 단어 예시 랜덤하게 제공
  - 자신이 출제하고 싶은 단어 자유롭게 입력 가능
- **유연한 게임 관리를 위한 Disable 처리** 
  - Timeout/Correct 등의 상황에서 Enter 키 비활성화
  - Guesser의 정답창은enter키 입력 후 clear 처리

# 결과 요약 및 기대 효과

- **디바이스 제어 및 커널 자원 활용**
  - 자원 낭비 없이 커널 자원을 효율적으로 활용하여 하드웨어 제어의 유연성 및 안정성 확보
- **다중 클라이언트 간 데이터 동기화**
  - 클라이언트 수에 관계없이 모든 데이터 일관성 보장
- **커널 최적화**
  - 1GB 인 현재 board 에서 RAM runtime 사용량 축소. App이 running 할 수 있는 쾌적한 환경 제공
  - 부팅속도 개선.
- **사용자에게 풍부한 경험 제공**
  - BGM / 기믹(힌트, 타이머) / UI를 통해
    추억의 게임 '캐치마인드'를 연상시켜 향수를 불러일으킴
  - LED, 효과음, 버튼 등을 통해 다양한 인터페이스를 제공하여 즐거운 경험을 선사

# 향후 연구 과제

## 화이트보드 성능 향상

- 통신 좌표 데이터 전달 속도, 화면에 그려지는 속도 개선

## H/W 관련 이슈

- 중복 터치 입력, 터치 중 보드 멈춤, 버튼 중복 입력, 리셋 근처 버튼(SW2) 누를 시 리셋 발생...

## 멀티 플레이 환경 개선

- 4인 이상 멀티 지원, 게임 환경 설정 기능 제공
- 원하는 대기실에 접속/퇴장 기능 추가

## 게임 몰입 요소 추가

- 되돌리기(CTRL+Z), 다양한 색상, 더 많은 펜촉 등 폭 넓은 편의성을 제공
- 여러 모드 (주제/테마 제시어, 한글 허용 등) 추가

# 프로젝트 수행 후기

**구현한 기능**

- 게임 내 BGM/효과음 적용
- 체계적인 게임 로직 구현
- 신규 버전 커널 포팅
- 서버-클라이언트 관리
- GPIO 제어
- LED 및 커널 타이머 제어

**교육 때 배운 내용**

- ALSA 드라이버 사용
- QT Library, Signal-SLOT
- 커널 포팅
- TCP 통신 및 커널 동기화
- Key 인터럽트 / Work_queue 사용
- HW 인터페이스 및 커널 자원 활용

# Git source directory

- **Kernel source directory**
  - https://github.com/inniestar97/GreonGrimBoot
- **QT source directory**
  - https://github.com/0112shpark/LG-bootcamp-girin-gurim
- **Server & device control source directory**
  - https://github.com/sssss1031/bootcamp_sketch