

# 포트폴리오 - 대학원 연구 주제 및 연구과제

---

# 연구 주제

---

## ☐ Android drawing API를 활용한 그림판 application 최적화

### ■ 개발 환경

- ☐ Ubuntu 16.04, AOSP(Android Open Source Project / JAVA, JNI, C++) , Android Studio

### ■ 개발 주제

- ☐ Android graphicpipeline을 수정하여 기본 Draw API를 활용한 그리기 어플은 렌더링을 최적화 하는 것
  - 현재 그리고 있는 요소만 rendering이 되지 않고 전체 화면이 rendering되어 전력적으로 비효율적이다.
  - 현 Android에서는 View 단위로만 rendering 영역을 잡아 그리기 Application에서는 View 단위 rendering은 좋지 않다.
  - 이를 해결하고자 현재 그리는 영역만 rendering을 하여서 전력적 이점을 가져온다.

### ■ 개발 방법

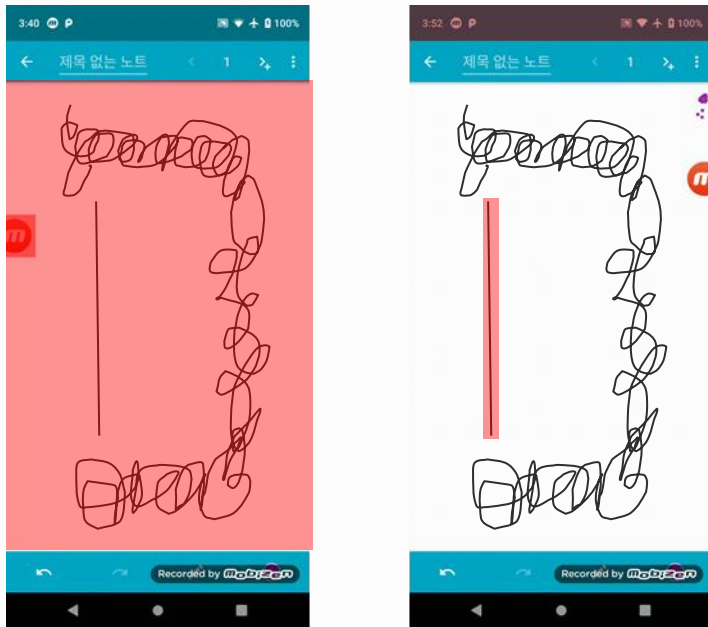
- ☐ AOSP의 코드 수정
  - 사용자의 현 터치 정보를 하이재킹하여 rendering 하는 영역으로 결정한다.

# 연구 주제

## □ Android drawing API를 활용한 그림판 application 최적화

### ■ 구현

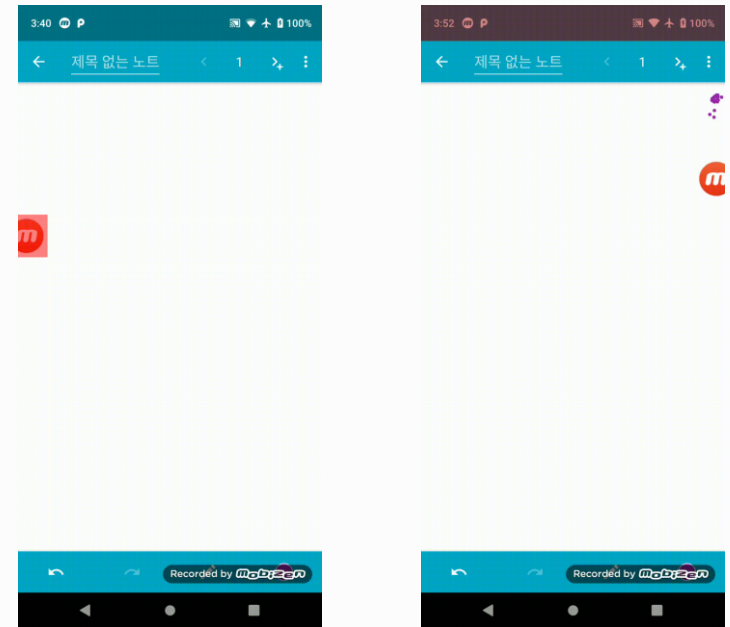
사진



< 순정 OS >

< 수정된 OS >

애니메이션



< 순정 OS >

< 수정된 OS >

 현재 렌더링 영역

# 연구 주제

□ Android drawing API를 활용한 그림판 application 최적화

■ 결과

Som note	Origin OS	Edited OS	Origin OS	Edited OS
Battery	1.920667	1.541667	2.497	1.638
CPU	4.908667	2.823667	9.793333	4.914667
GPU	17.35367	1.644667	16.75267	1.899667

Sketch book	Origin OS	Edited OS	Origin OS	Edited OS
Battery	2.547667	2.122667	2.569333	2.138
CPU	7.365667	5.817333	7.273333	5.695333
GPU	12.06333	0.911333	12.04967	1.058667

	Increase		Increase
Battery	20%	Battery	34%
CPU	42%	CPU	50%
GPU	91%	GPU	89%

	Increase		Increase
Battery	17%	Battery	17%
CPU	21%	CPU	22%
GPU	92%	GPU	91%

Thin thickness  
Default thickness

# 연구 과제

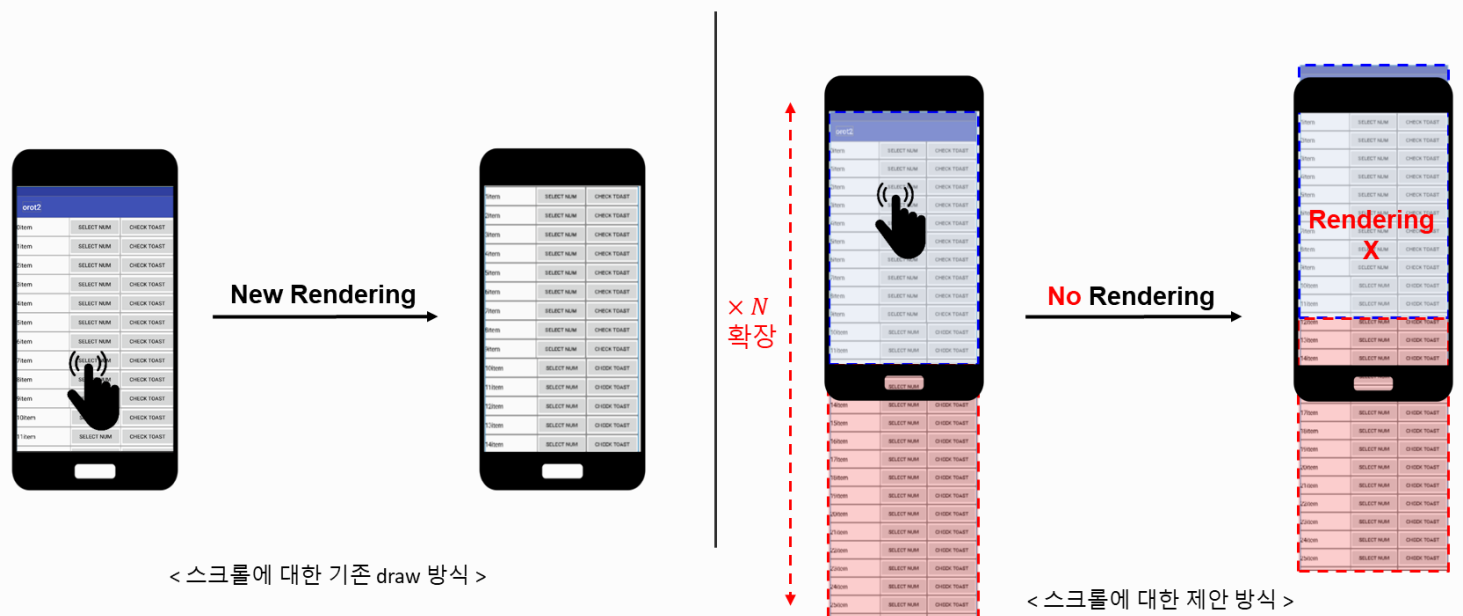
## □ Scroll-Aware Rendering to Reduce Frame Rates on Smartphones

### ■ 개발 환경

□ Ubuntu 16.04, AOSP(Android Open Source Project / JAVA, JNI, C++) , Android Studio

### ■ 개발 주제

□ 사용자가 Scroll하는 View에서 사소한 변화에도 View 전체를 Rendering 하는 문제를 해결하는 것이 목표



# 연구 과제

---

## ☐ Scroll-Aware Rendering to Reduce Frame Rates on Smartphones

### ■ 개발인원

☐ 3명

### ■ 개발 방법

☐ AOSP의 코드 수정

■ 기존보다 Rendering 되는 Frame의 크기를 확장해 scroll시 마다 새로 그리는 것이 아닌 재사용한다.

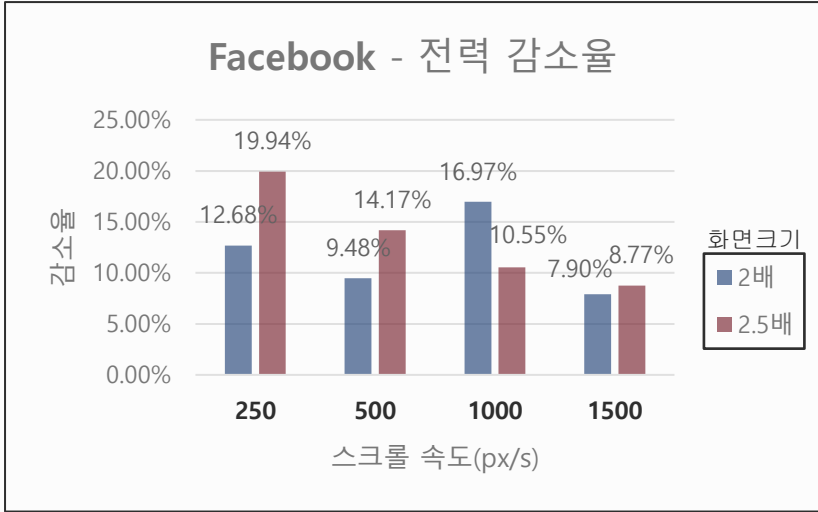
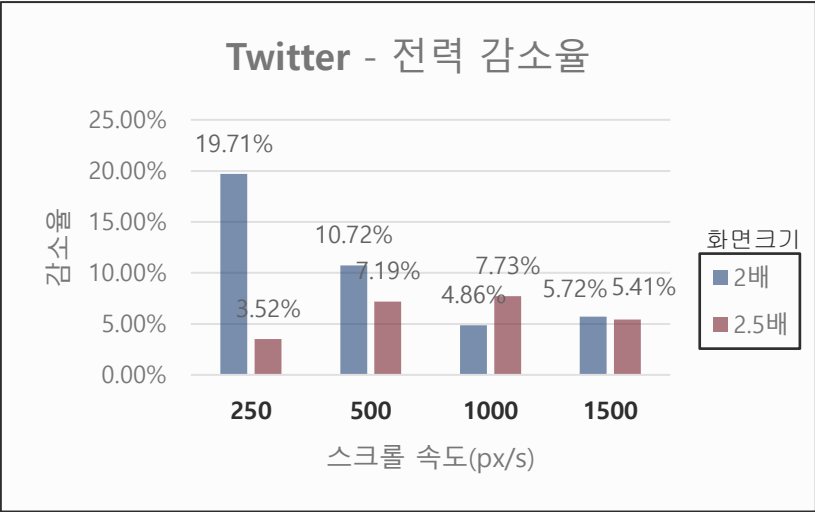
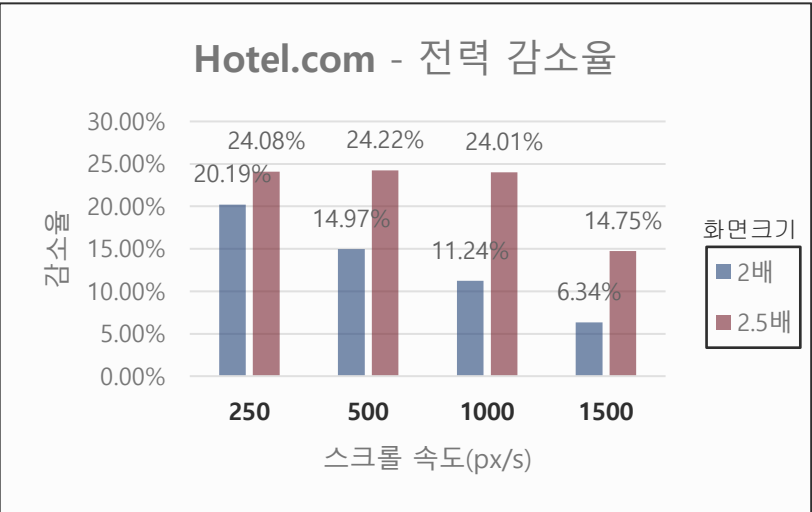
### ■ 담당업무

☐ Rendering pipeline 소스 코드 분석, 사용자의 Scroll 정보 하이재킹, Device에서 출력할 Frame의 위치 설정

# 연구 과제

## Android graphic system에서 Scroll의 상황에 Rendering 최적화

### 결과



# 포트폴리오 – 대학원 수업 프로젝트

---



# 인공지능 스피커를 활용한 게임 도우미

---

## ☐ SKT 산학 협력 인공지능 개론

### ■ 개발 환경

- ☐ Node JS, python, mySQL

### ■ 개발 주제

- ☐ 리그오브레전드의 게임을 플레이하는 동안 AI 스피커에게 음성으로 게임의 정보를 얻어올 수 있다.

### ■ 개발 인원

- ☐ 2명

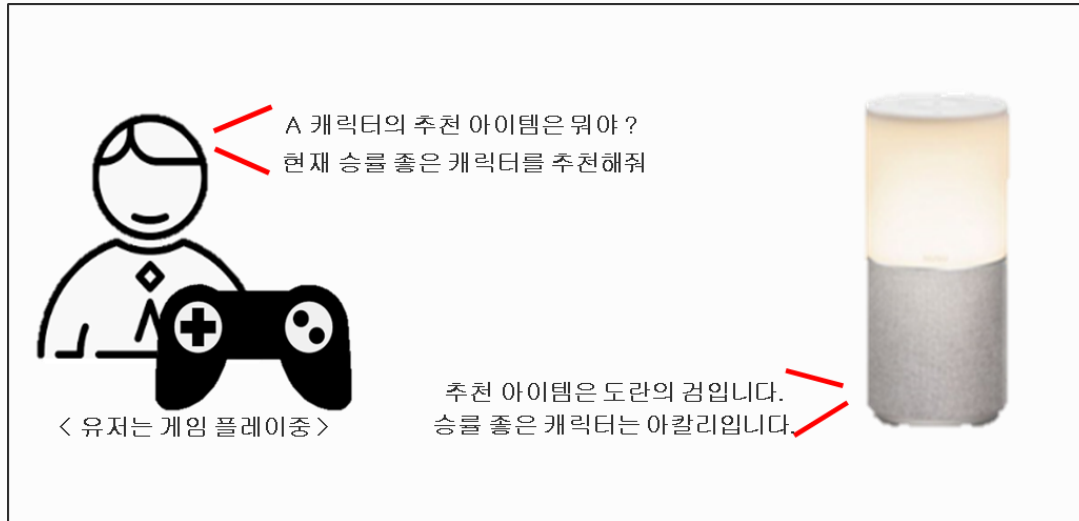
### ■ 담당 업무

- ☐ 파이썬으로 BeautifulSoup을 사용해 게임의 정보가 담겨있는 웹사이트를 크롤링
- ☐ 크롤링한 정보를 DB 서버에 갱신

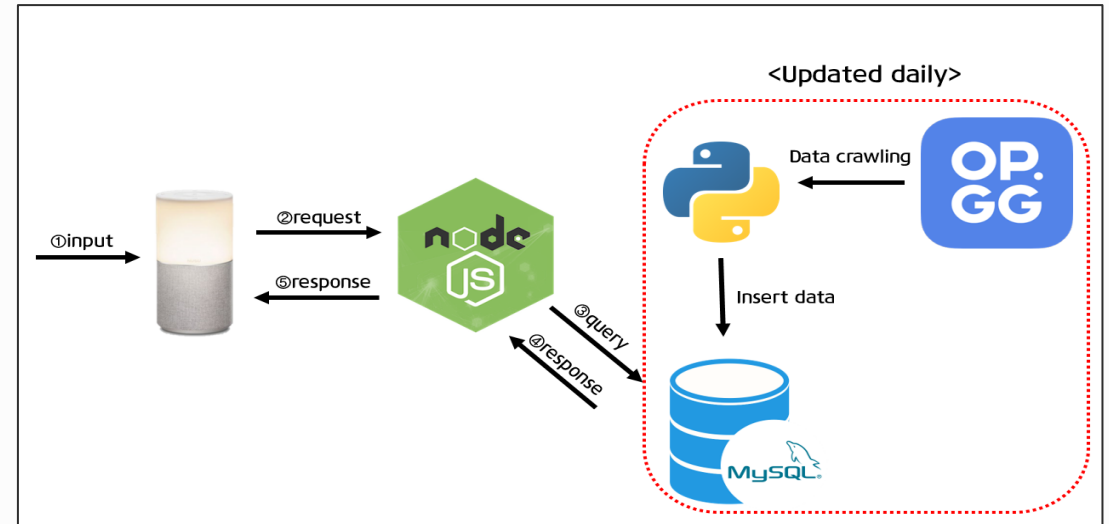
# 인공지능 스피커를 활용한 게임 도우미

## □ SKT 산학 협력 인공지능 개론

### ■ 시나리오 및 구현 방법



< 동작 시나리오 >



< 구현 방법 >

# 포트폴리오

## -학부 프로젝트

---

최성훈

# 리눅스에서 작업 관리자 구현

---

## ■ 개발 환경

- ☐ Ubuntu 16.04 (Bash shell)

## ■ 개발 주제

- ☐ 윈도우 처럼 리눅스에서도 프로세스를 종료할 수 있는 작업관리자 구현

## ■ 기능

- ☐ 현 상태에서 돌아가고 있는 유저의 프로세스 목록 매초 마다 새로 갱신하며 출력
- ☐ 화살표를 이용하여 커서 이동
- ☐ 커서가 가르키는 프로세스를 엔터키를 눌러 퍼미션을 가지고 있다면 종료하는 기능

# 리눅스에서 작업 관리자 구현

## ■ 구현 결과물

- 백그라운드로 Sleep을 돌린 후, 만든 작업 관리자로 종료를 시키면 Slepp이 사라지는 것을 확인 할 수있다.

```
223.194.45.84 - PuTTY
hunCom hun ~ LINUX_P LI_PJ 1 sleep 5000 &
[2] 8595
hunCom hun ~ LINUX_P LI_PJ 2
```

### 1. BackGround로 Sleep 실행

```
Practice
Linux
-NAME-----CMD-----PID-----STIME-----
avahi|B ash|30904|03:37
colord|B hd: hun@pts/1|30903|03:37
hun|B sr/lib/gvfs/gvfsd-ht|25549|7/15
lp|B sr/bin/python3 /usr/|24052|7/13
message+|B sr/lib/x86_64-linux-|9383|7/08
nobody|B date-notifier|8765|7/08
root|F in/bash ./main.sh|8601|03:45
rtkit|B eep 5000|8595|03:45
syslog|B sr/lib/x86_64-linux-|8040|7/08
systemd+|B sr/bin/zeitgeist-dae|8033|7/08
tomcat8|B in/sh -c /usr/lib/x8|8029|7/08
whoopsie|B itgeist-datahub|8022|7/08
|B ome-screenshot|5719|7/31
|B ome-screenshot|5714|7/31
|B sr/lib/gvfs/gvfsd-dn|2191|7/27
|B sr/lib/gvfs/gvfsd-ne|2172|7/27
|B in/bash|1979|7/08
|B ome-pty-helper|1978|7/08
|B sr/lib/x86_64-linux-|1976|7/08
|B sr/bin/python /usr/b|1968|7/08
If you want to exit , Please Type 'q' or 'Q'
```

### 2. 작업 관리자로 Sleep을 Kill

```
[2]+ 죽었음 sleep 5000
hunCom hun ~ LINUX_P LI_PJ
```

### 3. 작업 관리자 종료 후 Sleep이 죽은 것을 확인

# 리눅스에서 파일 탐색기 및 make 파일 생성기 구현

---

## ■ 개발 환경

- ☐ Ubuntu 16.04 (Bash shell)

## ■ 개발 주제

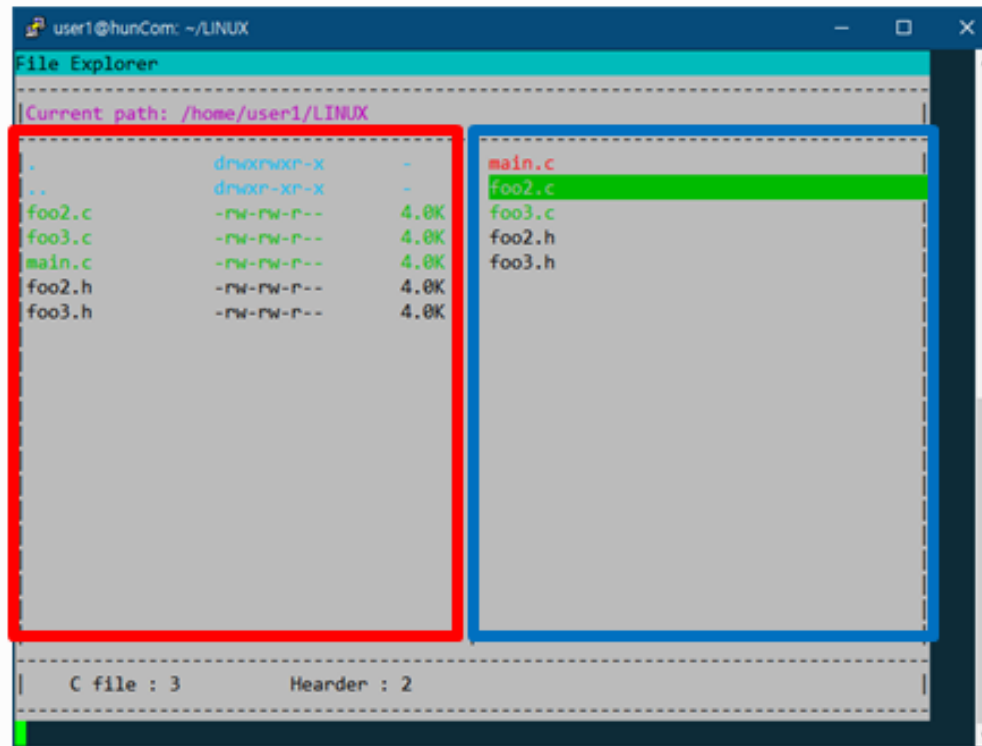
- ☐ 터미널창에서 파일 탐색기 및 make 파일 생성기 구현하기

## ■ 기능

- ☐ 키입력을 바탕으로 화살표를 움직임
- ☐ 현 경로가 가르키는 파일 리스트 출력
- ☐ 엔터키를 이용하여 경로를 이동
- ☐ .c와 .h 파일 그리고 디렉토리를 색으로 구별하여 출력한다.
- ☐ Make 파일로 만들고자 하는 .c와 .h 파일들을 리스트에 갱신해, 추가된 목록으로 make 파일로 만드는 기능

# 리눅스에서 파일 탐색기 및 make 파일 생성기 구현

## ■ 구현 결과물



현재 경로의 파일목록



추가된, Make 할 파일 목록