

Head-Up Display module for OpenDS

# 츠퍼카

KMU CS | 8조 | Purple Ocean

## Contents

I . 추진배경 & 목표

II . HUD App

III . 시스템구조도

IV . HUD API

V . HUD Layout Tool

VI . 기대효과

## I. 추진배경 & 목표

### HUD (Head-Up Display)



차량의 현재 속도, 연료 잔량, 내비게이션 길 안내 정보 등을  
운전자 바로 앞 유리창 부분에 그래픽 이미지로 투영해 주는 디스플레이 장치

## OpenDS (Open Source Driving Simulator)



OpenDS는 jME3 게임엔진을 이용해 독일에서 **오픈 소스로 개발된 Driving Simulator**로 자동차 주행에 대한 기능과 도로의 교통상황, 날씨 등을 반영할 수 있도록 만들어졌다.  
현재 OpenDS에선 HUD에 대해서는 지원하지 않는다.

### 추진배경

- OpenDS에는 HUD 기능이없음
- OpenDS에서 HUD 기능을 구현하고자하는 개발자들이 많음

**OpenDS에서 HUD기능을 쉽게 구현할 방법이 필요**



**OpenDS에 HUD 모듈을 구성**

누구나 HUD모듈을 사용하여 쉽게 HUD를 개발 할 수 있다

## 목표

Open Source  
Driving Simulator 에  
**HUD module 개발**



OpenDS에서  
누구나 쉽게  
**HUD 기능을 개발**하고  
test 가능

직접 만든  
HUD module을 사용



**Mobile 기기와 연동**하고  
**모션 인식 입력**을 통해  
HUD 기능을  
만들어 봄

## II . HUD App

## App 종류

- Navigation
- SMS
- Music Player
- BSA(Blind Spot Assist)
- Call

- HUD에 App의 형태로 추가될수있도록 interface를 만들었고 sample로 만든 기능은 5가지인데 Navigation, Call은 외부에서 오는것이므로 메뉴로 만들지 않았음.
- 이 외에도 개발자들이 원하는 기능을 App으로 추가 할 수 있음.

## 특색

### 1) App을 메뉴바로 만들어서관리



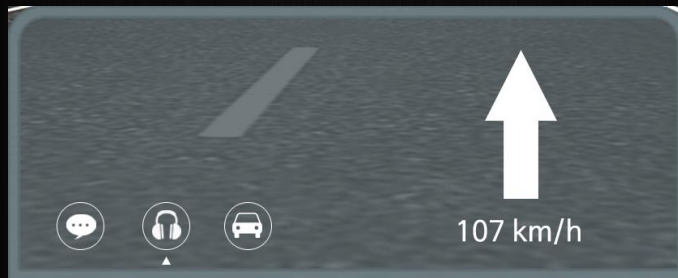
HUD의App을사용할수있도록  
메뉴바를구현  
메뉴바는좌우로 옮겨가며탐색하고  
선택하여App을실행가능

### 2)G-HUB를사용



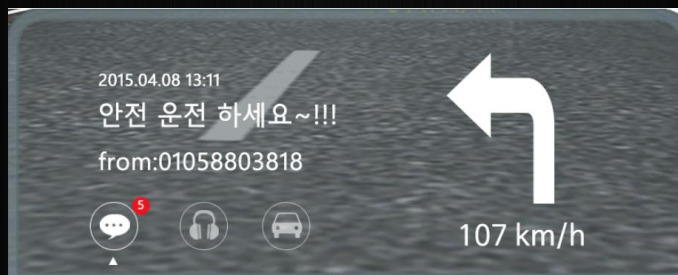
동작인식센서기술을 탑재한 입력장치.  
HUD interface에서  
G-HUB를 사용하여 입력을 컨트롤

## Navigation



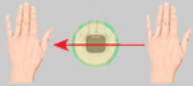

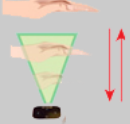
메뉴를 선택하지 않은 기본 화면  
**현재 속도, 진행방향** 출력

## SMS

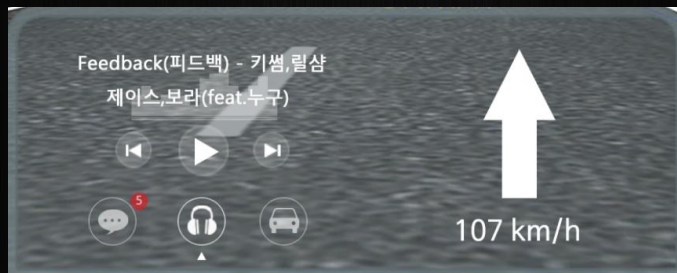


주행 중 문자가 오면 메뉴창에 수신된 **문자 개수**를 बै지로 **표시**  
 메뉴에서 SMS창을 선택하면 **SMS 확인** 가능

## SMS mapping

	Key	Motion
이전 SMS	F2	
다음 SMS	F3	
메뉴 나가기	N	

## Music Player



Music List에서 모션인식으로 곡을 선택하면  
Music play 화면이 뜨고 **음악이 재생**됨

## Music Player mapping

Music List	Music play	Key	Motion
커서 위로	이전곡재생	F2	
커서 아래로	다음곡재생	F3	
선택	재생/일시정지	N	
메뉴나가기	List로 이동	Z	

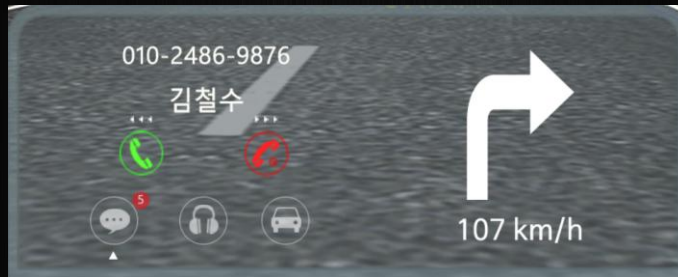
## BSA (Blind Spot Assist)



BSA 메뉴를 선택하면 현재 차 주위 **측.후방에 다른 차량이 있는지 감지하고 경고 이미지** 출력  
 BSA 메뉴를 선택하지 않더라도 **측.후방에 다른 차량이 가깝게 접근하면 경고 이미지** 출력



## Call



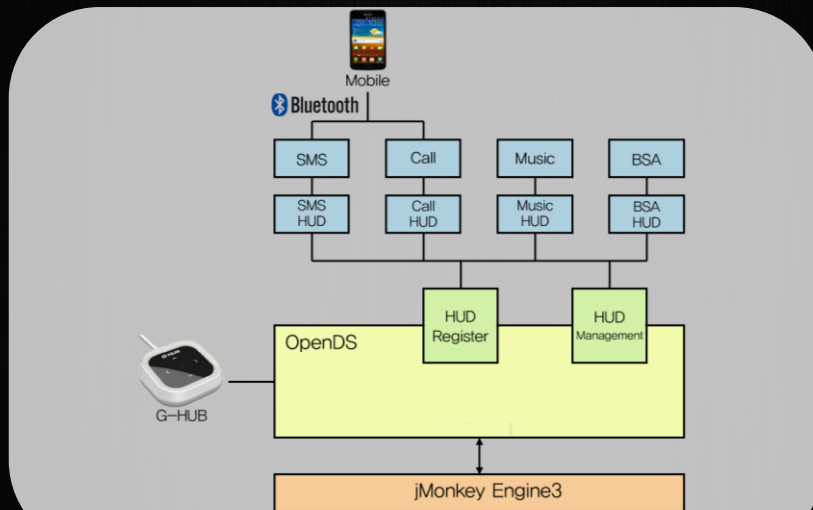
주행 중 전화가 오면 **통화 수신 화면**을 띄워줌  
 모션인식으로 수신, 거절 선택이 가능  
 통화를 수신하면 통화 중 화면으로 전환

## Call mapping

	Key	Motion
통화 수신	F2	
통화 거절	F3	
통화 종료	N	

### Ⅲ. 시스템 구조도

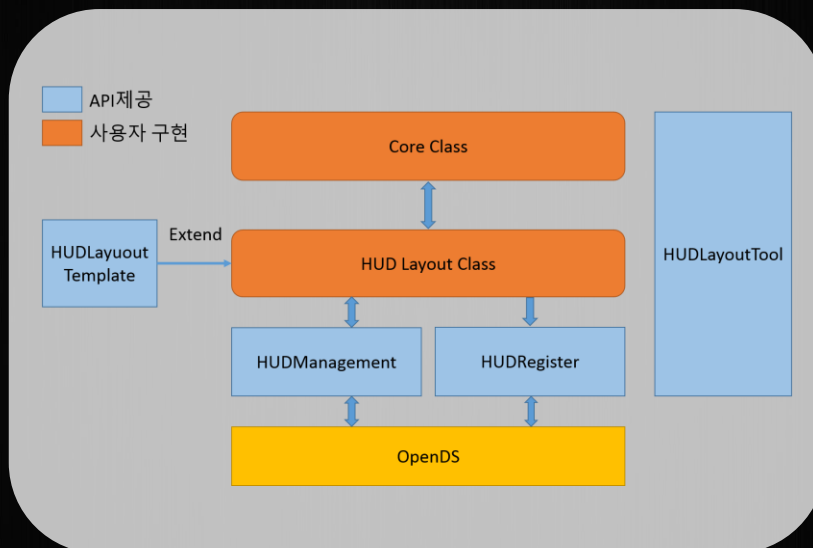
#### 시스템 구조도



※ Navigation 기능은 HUDManagement에 기본으로 내장되어 있다

## IV .HUD API

### HUDAPI 구조

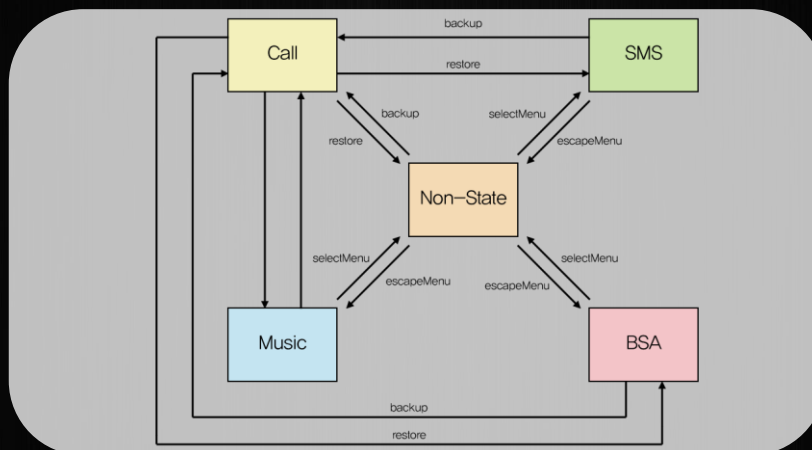


## HUD API method

Method명	설명
backupHUD(int changeState)	이전 State를 저장하고 입력된 State로 변경
restoreHUD()	현재 State를 종료하고 이전 State로 복구
getState()	현재 실행중인 State를 반환
escapeMenu()	메뉴 선택 모드(Non-State)로 이동
disableMenu(int state)	입력된 state의 메뉴 아이콘을 제외하고 모두 비활성화 (-1일 경우 전체 비활성화)
enableMenu()	모든 메뉴를 활성화
regist()	HUDManagement에 HUD 등록
setMenuIcon(Picture pic_en, Picture pic_dis, int state)	메뉴 아이콘을 메뉴바에 등록
attach(Node subGui)	HUD 패널에 입력받은 Layout을 삽입
detach(Node subGui)	HUD 패널에 입력받은 Layout을 제거

이 API는 **HUDManagement class**에 존재하는데 이 클래스의 핵심 기능은 **HUD Layout 별로 State를 할당하고 이를 관리하는 것**이다  
또한 HUD의 메뉴 부분을 간단하게 등록하고 관리할 수 있는 기능을 제공한다

## HUD State



위의 Call을 제외하고는 모두 메뉴에 등록하는 함수이므로 메뉴를 선택하면 HUD 패널에 디자인한 HUD Layout이 올라간다. 하지만 Call의 경우 외부에서 이벤트가 발생하면 바로 실행되므로 자체 구현한 Update 함수에서 backup과 restore method를 호출하고 state를 직접 변경해야 한다.

## V .HUD Layout Tool

### HUD Layout Tool



←↑↓→키를 통해 **Element** 위치를 설정하고  
F8/F9키로 **offset**을 설정할 수 있는 **Layout Tool**을 개발

## 구현배경 및 개선점

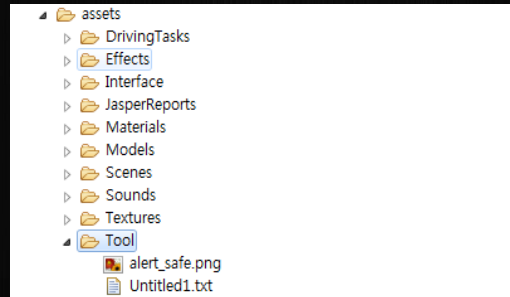
**배경** : HUD에 필요한 element들의 위치 및 크기를 시뮬레이터 화면에 맞게 조정하는 일의 번거로움을 해결하고, OpenDS의 개발자에게 도움을 주기 위해 구현

**개선점** : 평소에는 일일이 수치를 타이핑으로 기입하고 시뮬레이터를 실행해가며, 실행 시 화면에서 element들의 배치 및 크기가 적당한지를 테스트를 하였지만, 이 툴을 통해서 element를 시뮬레이터 실행화면에서 방향키 조작으로 위치 및 크기를 조정하고, 위치 및 크기 값을 콘솔 창으로 출력하게 하여 번거로움을 해결

## 단축키 정의

단축키	단축키 기능
Q	HUDLayoutTool 온 / 오프 기능
F7	Element Type의 전환 (text ↔ picture)
F8	Mode 전환 (Move ↔ Size)
F9 / F10	Offset 증가 / 감소
F10 / F11	이전 / 다음 element 선택
F12	현재 선택된 element의 좌표 및 크기를 콘솔 창에 출력
Delete	현재 선택된 element를 제거

## Tool 사용법



1) /Assets/Tool/ 디렉토리에 element 소스인 .txt 파일과 .png 파일을 복사



2) 시뮬레이터를 실행하고 Q키를 눌러 HudLayoutTool 모드로 전환



3) 방향 키를 이용하여 원하는 위치로 element를 이동  
(F9/F10 키로 움직임의 단위를 줄이고 늘릴 수 있음)



4) F7을 눌러 Mode를 Size Mode로 전환하고 방향키로 크기를 늘리거나 줄임  
(Move mode면 Size mode로 Size mode면 Move mode로 전환)



```

Simulator [Java Application] C:\Program Files\Java\jre1.8.0_25\bin\javaw.exe (2015. 5. 25. 오후 1:50:13)
SettingsControllerServer started at port 5678
12

Camera Position: (-250.25597, 3.8619335, 271.81274)
Camera Rotation: (0.014069882, 0.86607593, -0.0033039672, -0.4997036)
Camera Direction: (-0.8656554, 0.008338568, -0.5005708)
cam.setLocation(new Vector3f(-250.25597f, 3.8619335f, 271.81274f));
cam.setRotation(new Quaternion(0.014069882f, 0.86607593f, -0.0033039672f, -0.4997036f));
alert_safe.png: pos(346, 184) size(437, 187)

```

5) 선택한 원소의 위치 및 크기를 얻기 위해 F12키를 누름  
(콘솔 창에 출력)



6) F7 키를 누르면 element Type을 text로 전환 할 수 있음  
(조작 방식은 그림과 같음)



7) Delete key를 누르면 화면에서 원소를 제거할 수 있음

## VI. 기대효과

## 기대 효과

- OpenDS에서 기존에 없던 **HUD API를 제공**하므로 개발자들이 **자신만의 HUD 기능을 구현**할 수 있다.
- OpenDS에 HUD 기능이 추가됨에 따라 **운전정보를 화면에 출력**할 수 있기 때문에 OpenDS개발자들의 **편의성과 흥미를 증진** 시킬 수 있다.
- 현실에서 HUD를 구현할 때보다 **시간적, 경제적 노력이 절감**된다.

Thank you

**감사합니다!**