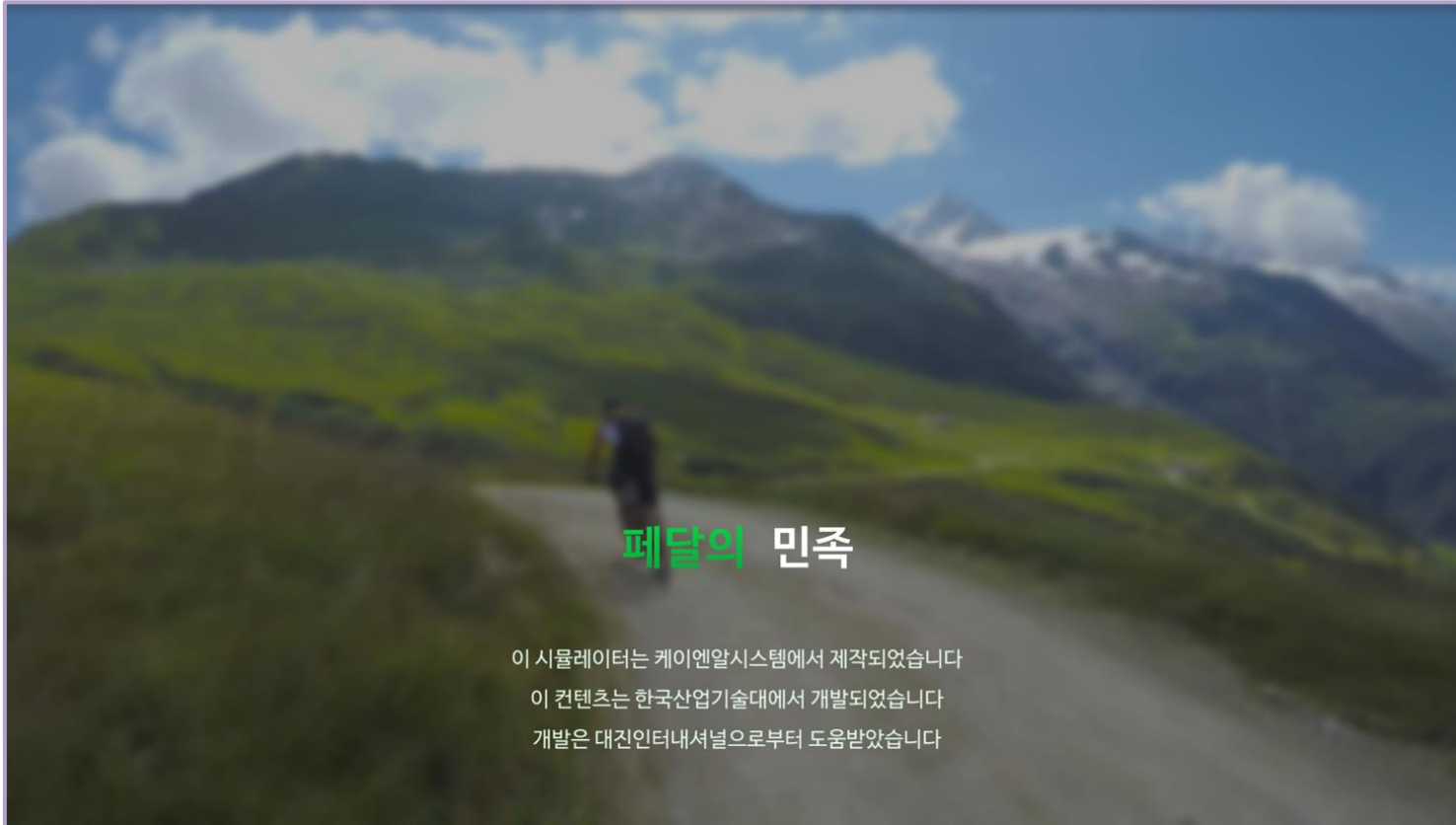


# CURSUS 상세 설명

게임 개발자 김대호

# CURSUS



## 가상 MTB 자전거 시뮬레이터

실제 MTB 자전거를 거치하여 가상의 MTB 자전거를 주행 할 수 있는 콘텐츠 입니다.  
매장에서 실제 자전거를 거치하여 각 자전거의 성능 확인 및 비교, 장비 조정을 할 수 있도록 고안되어 제작하였습니다.


## 참여 기간

2015.12 ~ 2016.10 (11개월)

## 개발 콘텐츠

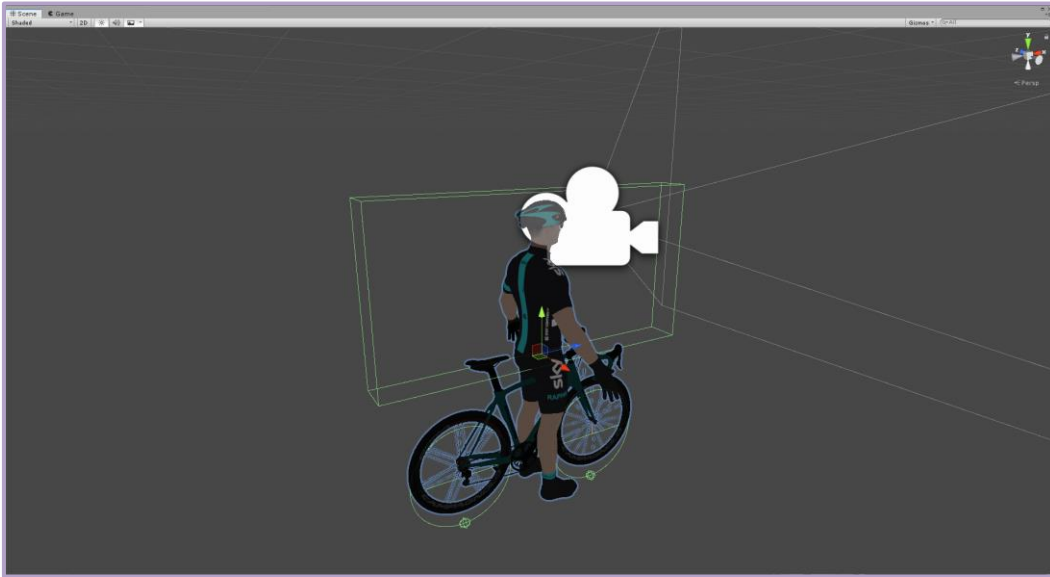
가상 자전거 구현  
맵 선택 구현  
인게임 UI 기능 구현  
3D , 2D 리소스 적용  
주행 기록 구현  
시뮬레이터 통신 구현

## 개발 방식

- Unity 
- Ant+
- UDP/IP
- Json

# 개발 컨텐츠

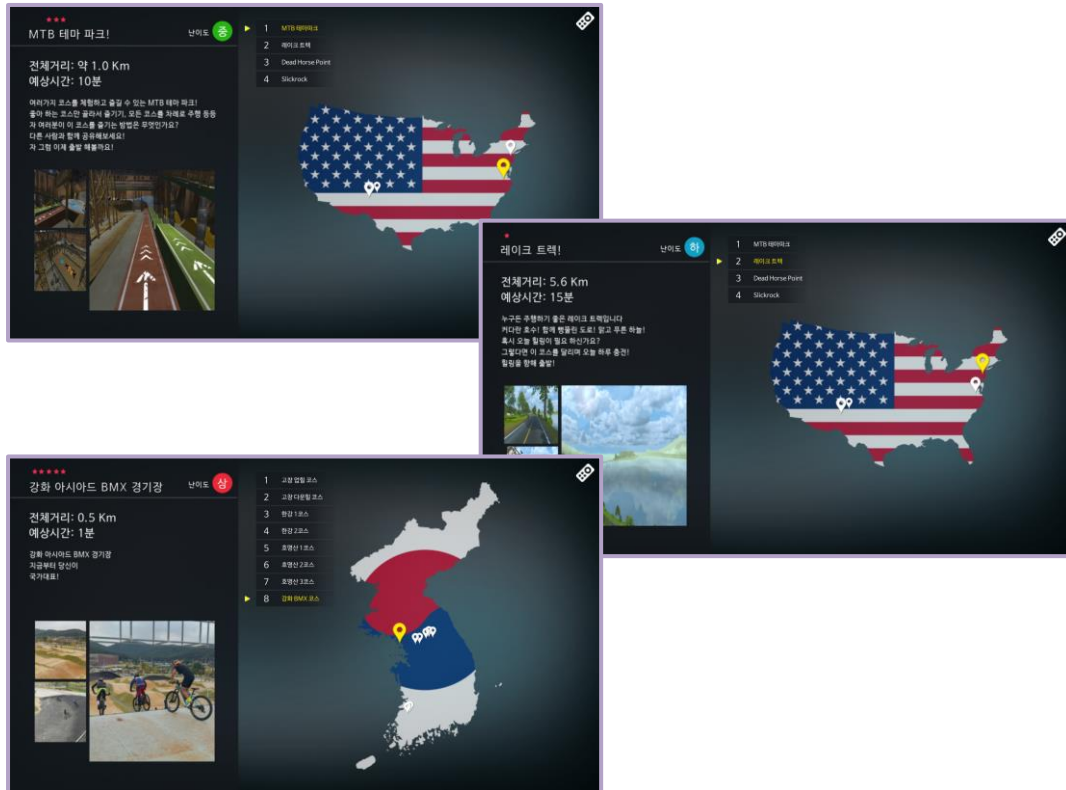
## 가상 자전거 구현



- Box collider, Wheel collider, Rigidbody를 이용해 가상 자전거를 구성함
- 여러 타입의 입력 방식을 구현함
  - 방향조작: 키보드 및 레이스 휠, 시뮬레이터
  - 이동: 키보드, ANT+, 시뮬레이터
- 후면 Wheel collider의 motorTorque, Rigidbody의 velocity 를 이용하여 가상 자전거의 속력을 컨트롤 함
- 전면 Wheel collider의 y축으로 방향을 컨트롤 함
- 전후면 Wheel collider만으로는 중심을 잡지 못하기 때문에 Rigidbody의 무게 중심을 Wheel collider와 지면의 접점보다 낮게 설정하여 무게중심을 설정

# 개발 콘텐츠

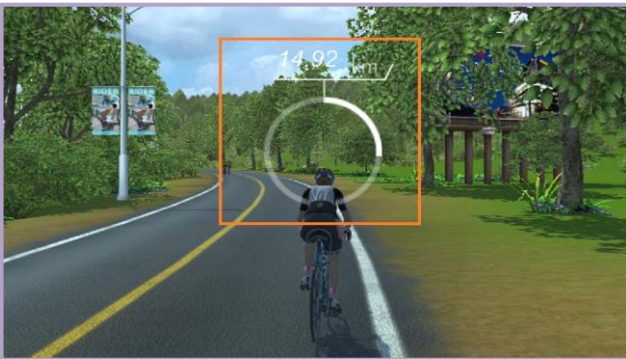
## 맵 선택 구현



- 맵 표기 및 목록 오브젝트는 오브젝트 풀링으로 구현
- 실사 아웃도어 트랙
  - 한국, 프랑스, 미국
- 가상 트랙
  - 인도어 MTB 트랙, 아웃도어 호수 트래킹 트랙

# 개발 컨텐츠

## 인게임 UI 기능 구현



- UGUI로 UI 구현
- Anchor Preset으로 UI의 Pivot을 수정
- UI 에서도 속도감을 주기 위해 Player의 Velocity에 따라 좌우 UI의 회전각을 다르게 하고 상단 UI속도 게이지 역시 변화를 줌
- AI player UI는 World Space로 구성하고 LookAt 함수로 Player를 바라보게 하며 Player와 거리에 따라 on/off 폴링형식으로 구현함

# 개발 콘텐츠

## 3D, 2D 리소스 적용



- 3D 리소스 및 텍스처는 외주로 제작함
- 당시 ScaleFactor 정의 없이 진행하여 사이즈 차이가 발생함
- 2D 리소스(UI)의 경우 회사 그래픽직원이 포토샵을 이용해 제작
- Player, Map, AI Player등 3D 리소스 적용
- UI 및 2D 오브젝트 리소스 적용
- 실제 촬영한 실사 동영상을 AVPro 서드파티를 사용하여 맵으로 적용

# 개발 컨텐츠

## 주행 기록 구현

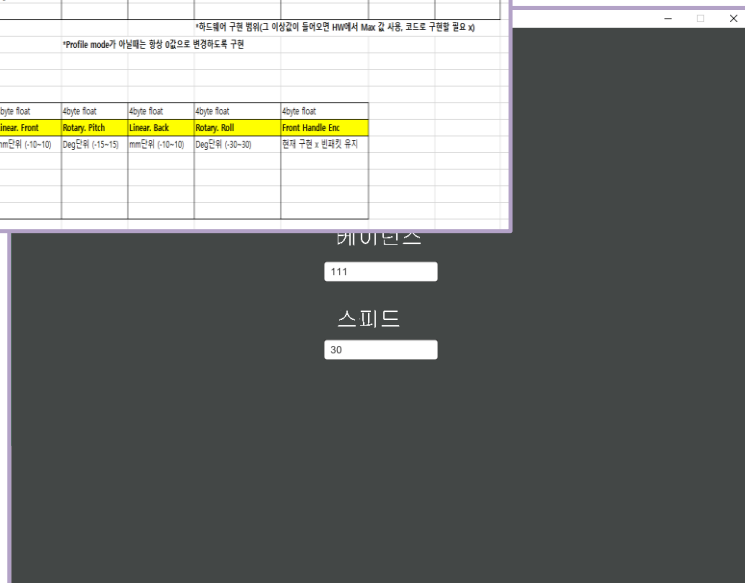
순위	이름	최고속도	평균속도	주행거리
1	이성욱 7741	10.10 Km/h	55.67 Km/h	0.01 Km
2	text 1234	42.00 Km/h	32.45 Km/h	0.16 Km
3	이대현 9954	24.46 Km/h	23.17 Km/h	2.03 Km
4	신정원 7203	13.74 Km/h	8.62 Km/h	1.45 Km
5	가나다 4321	0.00 Km/h	0.00 Km/h	0.00 Km
6	가나다 1234	0.00 Km/h	0.00 Km/h	0.00 Km

- 주행 기록은 Json 형식으로 기록 및 관리하며 로컬에 저장함
- 기록을 Map별로 구분되며 이름과 번호, 최고 속도, 평균속도, 주행거리를 하나의 Json객체로 하여 기록함
- 결과 창에서 주행 결화를 기록할 수 있으며 최고속도, 평균속도, 주행거리 별로 정렬 할 수 있음

# 개발 콘텐츠

## 시뮬레이터 통신 구현

1	A	B	C	D	E	F	G	H	I	J	K	L	M
2	신기대 Unity Send Packet												
3	size	1byte	1byte	1byte	1byte	1byte	1byte	1byte	1byte	4byte float	4byte float	4byte float	4byte float
4	Name	header	index	length	State	LCU mode CMD	Vibration Pattern	Profile NO.	Profile mode	Linear Front	Rotary Pitch	Linear Back	Rotary Roll(각도)
5	Default Value	0x03	0x31	0x1A(26)	0	RST	0	0	0	0	0	0	0
6	Kind	0x03	0x31	0	0(none)	RST(대기)	0(none)	*0(None), 255	none(0)	*mm단위 (-10~10)	Deg단위 (-15~15)	mm단위 (-10~10)	Deg단위 (-30~30)
7					1(game)	OR(작동)	1(asphalt)		play(1)				
8					2(profile)	INT(준비과정)	2(wood)		pause(2)				
9					3(performance)	FIN(종료과정)	3(ground)		stop(3)				
10							4(gravel)						
11													
12													
13													
14													
15	*Profile mode가 아닐때는 항상 0값으로 변경하도록 구현												
16	신기대 Unity Server to Unity Read Packet												
17	size	1byte	1byte	1byte	1byte	1byte	4byte float	4byte float	4byte float	4byte float	4byte float		
18	Name	header	index	length	LCU mode Read	Profile Load	Linear Front	Rotary Pitch	Linear Back	Rotary Roll	Front Handle Exc		
19	Kind	0x03	0x11	0x1A(26)	RST(대기)	0	mm단위 (-10~10)	Deg단위 (-15~15)	mm단위 (-10~10)	Deg단위 (-30~30)	현재 구동 x 원패킷 유지		
20					RST(준비 완료)	0							
21					OR(작동중)	loading(1)							
22					INT(준비 진행중)	loadDone(2)							
23					FIN(종료 진행중)								
24													
25													



- UDP/IP 통해 장치들과 통신함
- 시뮬레이터 엔지니어와 패킷을 규정하고 UDP/IP 로 데이터 통신함
- 입력 장치 통신 전용 Thread를 만들어 receive로 인한 무한 대기상태 회피함
- 시뮬레이터와 PC간 통신 모듈과 호환 되도록 Ant+ 센서도 오픈소스 자료를 UDP패킷 서버로 수정하여 적용함
- 시뮬레이터, Ant+ 센서 없이 테스트 할 수 있는 환경을 구성하기 위해 가상 시뮬레이터 프로그램을 제작하고 루프백으로 기존 통신 모듈과 연결하여 테스트 진행함
- 주행장치로서 Ant+ 센서와 자전거 시뮬레이터 중 선택할 수 있음