

# GPS + 칼만 필터 기반 헬스케어 게이미피케이션 플랫폼

---

연세대학교 미래\_김호중

## 칼만 필터 프로젝트

### 왜 하면 좋은가?

1. 헬스케어 도메인을 살리면서 차이점이 부각된다.

➤ 취업을 위해서는 명확한 차이점이 존재 해야 한다. 즉 뻔한 포트폴리오는 매력이 크게 없다.

뻔한 사람들과의 차별점이 될 수 있는 부분은 “무언가를 직접만드는” 사람이 되는 것이다.

2. 데이터를 직접 만들어 내고 다룰 수 있다.

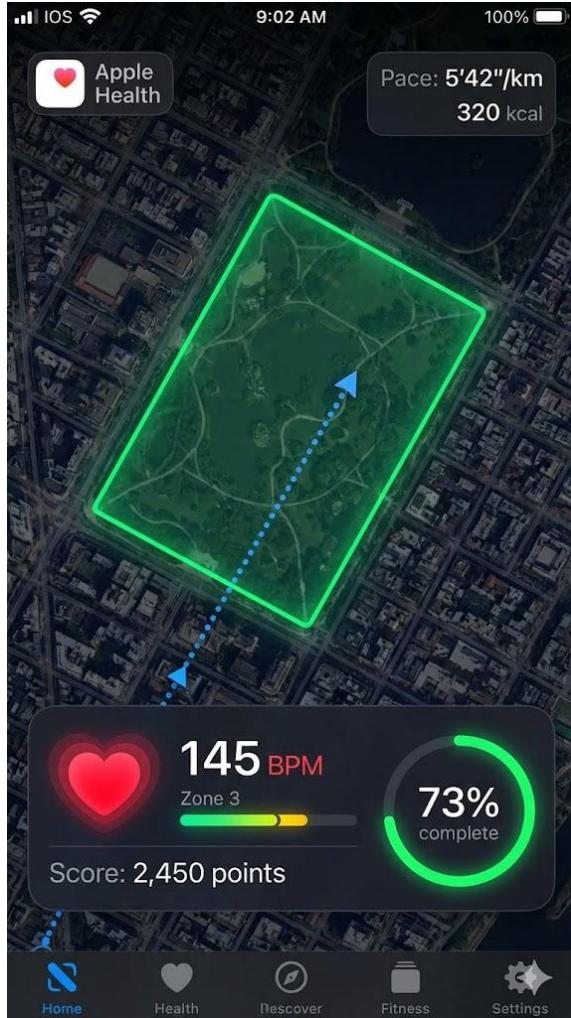
➤ 민감 정보로 취급되는 인간의 신체 데이터를 직접 수집하고 처리 가능하다.

기기의 제작, 소프트웨어 제작, 데이터 관리 및 분석 까지 이어지는 프로세스를 하나의 프로젝트에서 가능하게 한다.

3. 만들기가 어렵다.

얼마나 어려운지도 모르겠다. 처음 해본다. 수학 엄청 잘해야 한다. 가우시안기반 해석은 아직도 모르겠다.

## 프로젝트 예상 결과



### 영역 감싸기

이거 만들려면 GPS사야한다.

지도 위 특정 영역(폴리곤)을 운동으로 "정복"

예: 한강공원 산책로, 아파트 단지, 동네 블록

완주 판정: Point-in-Polygon 알고리즘 + 경로 커버리지 계산

### 심박 기반 동적 보상

이거 만들려면 심박수 센서 사야한다.

같은 거리라도 운동 강도에 따라 점수 차등

Zone 1-2 (걷기): 기본 점수  $\times 1.0$

Zone 3-4 (조깅/달리기): 점수  $\times 1.5 \sim \times 2.0$

### 실시간 피드백 & 게이미피케이션

이걸 공원이라 하고

점선 처럼 휘감아서 점수를 얻는다

현재 심박존, 누적 점수, 예상 칼로리 실시간 표시

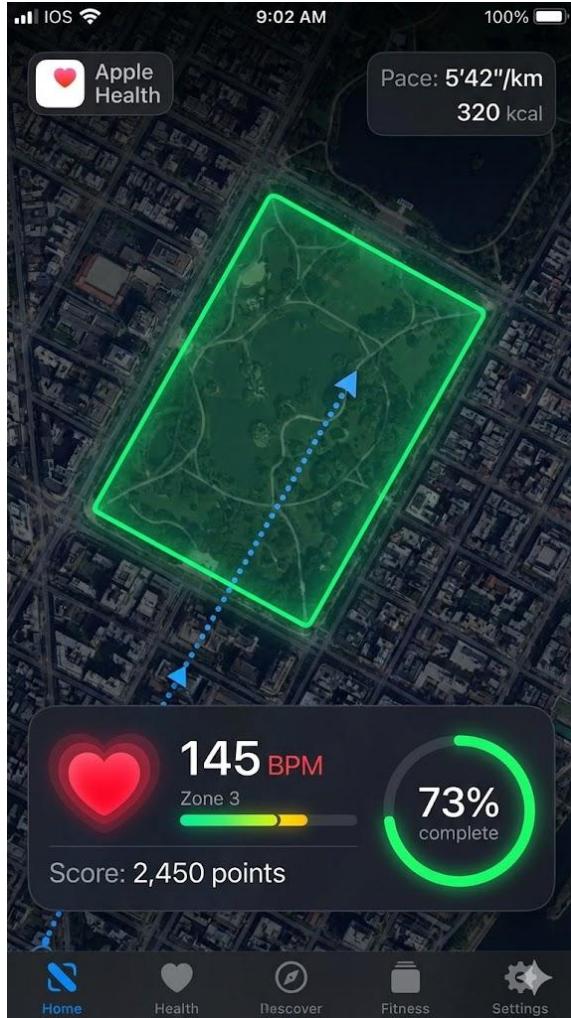
영역 완주까지 남은 거리/시간 안내

일간/주간 리더보드, 친구와 경쟁

이게 좀 고민인데 데이터수집과 처리에 집중할지 or 트랙잭션 만들지 골라야한다.



## 프로젝트 예상 결과



이걸 공원이라 하고  
점선 처럼 휘감아서 점수를 얻는다

## 영역 감싸기- 칼만필터

이걸 그냥하면 정확도가 구리다 그래서 칼만필터라는 것을 쓴다.  
대충 수학 기반 추정이다.

$$x = [p_x, p_y, v_x, v_y, \theta]^T$$

위에가 상태벡터 라는 얘인데 얘를 사용해서 상태를 추정한다.

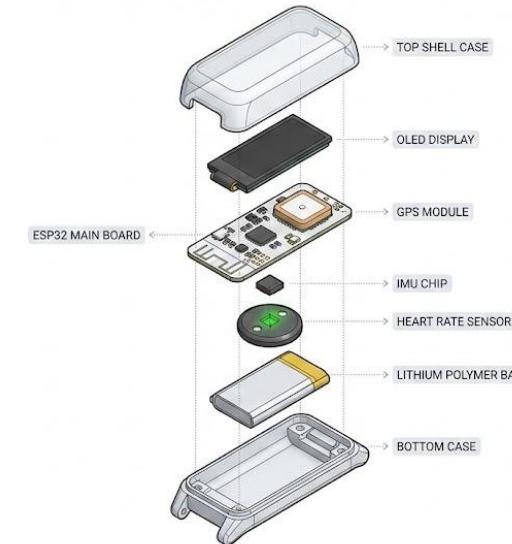
$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_k)$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1})$$

좀 쉽게 풀어서 설명하면  
현재 나의 상태(t)는 이전상태(t-1)와 이전상태에서의 동작으로  
추정 가능하다라는 말이다.

더 알고 싶다면 히든 마르코프 모델과 베이지안 필터  
칼만필터를 공부해야 한다. 나도 잘 모른다. 어렵다 매우

## 웨어러블 기기



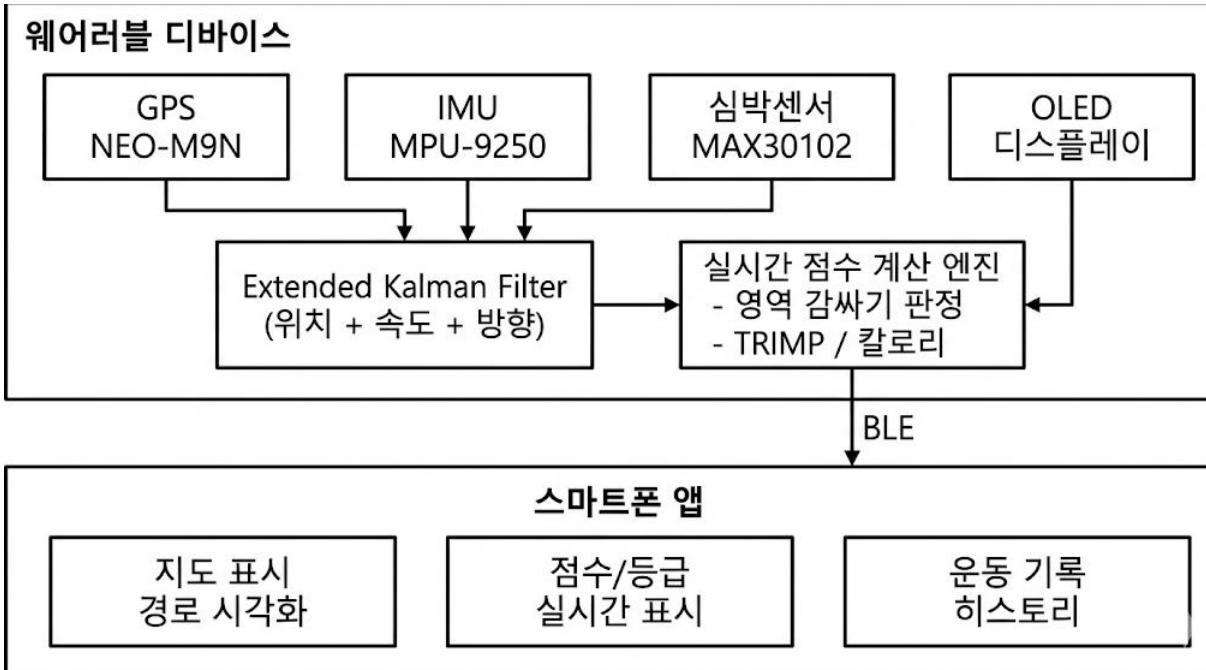
우리가 선택할 수 있는 선택지가 대충 4개가 있다.

1. 기성품 웨어러블 + API 연동
  2. 스마트폰 온리, 내장 gps랑 가속도 카메라 써서
  3. 최소 하드웨어 + 시뮬레이터
  4. 하드웨어 덕지덕지
- 4. 하드웨어 덕지덕지 이게 가장 이상적이지만 돈도 들고 하드웨어 파트에서 얼마나 시간 쓸을지 모른다.  
GPS만 하면 5만원 안쪽 심장박동과 이것저것 하면 15만쪽으로 구매 가능하다  
이걸 하고자 하는 이유는 센서데이터 직접 획득과 칼만필터 적용을 위해서

매우 이상적인 예상도

현실은 무언가 전자기기와 덕지덕지된 센서를 예상한다.

## 어플 대충



## 이런것들 필요하다

구분	컴포넌트 모델	주요 역할	주요 사양 및 특징
메인 프로세서 (MCU)	ESP32-S3	시스템 제어 및 통신	<ul style="list-style-type: none"> <li>Dual Core 240MHz</li> <li>BLE 5.0 / Wi-Fi 내장</li> <li>FreeRTOS 지원</li> </ul>
GPS 모듈	u-blox NEO-M9N	위치 추적	<ul style="list-style-type: none"> <li>1.5m CEP 정확도</li> <li>GNSS 멀티밴드 수신</li> <li>25mW 저전력</li> </ul>
IMU 센서	MPU-9250	모션/자세 감지	<ul style="list-style-type: none"> <li>9축 (가속도+자이로+지자기)</li> <li>200Hz 고속 샘플링</li> </ul>
심박 센서	MAX30102	생체 신호 측정	<ul style="list-style-type: none"> <li>PPG 방식 (광학식)</li> <li>SpO2(산소포화도) 측정</li> </ul>
배터리	LiPo 500mAh	전원 공급	<ul style="list-style-type: none"> <li>3.7V 리튬 폴리머</li> <li>약 8시간 연속 사용</li> </ul>

기능 블록	상세 내용	데이터 흐름
센서 데이터 수집	GPS[위치], IMU(자세), HR(심박) 데이터 동기화 및 수집	Sensors → ESP32 (I2C/UART)
신호 처리 (Filter)	칼만 필터 (Kalman Filter) 적용	센서 노이즈 제거 및 위치/자세 추정 오차 보정
연산 (Logic)	스코어 계산 (Score Calc)	수집된 데이터를 기반으로 사용자 활동 점수화
통신 (Stack)	BLE 5.0 Stack	가공된 데이터를 스마트폰으로 전송 (저전력)

대충 이런것들 필요하다 느낌으로 받아들여 주세요.

알리랑 국내랑 가격도 봐야하고 호환이 되는건지 조립이 어려운지 납땜 필요한지 등 알아봐야할 것들이 좀 많아서

## 데이터 엔지니어링

토픽명 (Topic)	파티션 키	메시지 구성 (Payload)	처리량 (Throughput)	보존 기간	특징
raw.gps.events	user_id	사용자 ID, 위/경도, 정확도, Timestamp	~1 msg/sec	7일	사용자별 순서 보장 필수
raw.hr.events	user_id	사용자 ID, BPM, RR Interval, Timestamp	~1 msg/sec	7일	심박 변이도 분석용 데이터 포함
raw imu.events	user_id	사용자 ID, 가속도[3], 자이로[3], 지자기[3]	~50 msg/sec	3일	고빈도 데이터, 용량 최적화를 위해 보존 기간 단축

인프라와 데이터 파이프라인은 여기까지 구성한 이후의 문제

수집할 수 있는 데이터와 그에 따라 발생되는 데이터 처리되는 데이터가 굉장히 많다. 물론 질은 보장 못한다.

수집된 데이터에 대한 DB구성과 트랜잭션 처리 등등 할 수 있는 것들과 해야 하는 것이 많다.

위는 카프카 예시 정도

## 베이지안 필터

$$p(x_t | z_{1:t}, u_{1:t}) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) \cdot p(x_t | z_{1:t-1}, u_{1:t})}{p(z_t | z_{1:t-1}, u_{1:t})}$$

$u_{1:t}$ : 처음부터 시간 t까지의 모든 제어 입력

$$\Rightarrow p(z_t | x_t, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)$$

위 식은 마르코프 가정에 의하여 다음과 같은 형태로 변환할 수 있다.

현재의 관측값  $z_t$ 는 오직 현재의 상태  $x_t$ 에만 의존

$$bel(x_t) = \eta \cdot p(z_t | x_t, z_{1:t-1}, u_{1:t}) \cdot p(x_t | z_{1:t-1}, u_{1:t})$$

분모 = 노말라이저(n)로 정리하면 위와 같이 표현된다.

$$\Rightarrow bel(x_t) = \eta \cdot p(z_t | x_t) \underbrace{p(x_t | z_{1:t-1}, u_{1:t})}_{\text{Prediction Term}}$$

마르코프 가정에 의해 한번 더 정리한 꼴

$$p(x_{t-1} | z_{1:t-1}, u_{1:t})$$

$u_t$ 는 시각 t에 가한 제어이므로, 과거인 시각 t-1의 상태  $x_{t-1}$ 에는 영향을 줄 수 없음

$$\Rightarrow p(x_{t-1} | z_{1:t-1}, u_{1:t-1})$$

$$\Rightarrow = bel(x_{t-1}) \quad \text{위과 같이 정리가능}$$

## 베이지안 필터

$$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | x_{t-1}) bel(x_{t-1}) \Rightarrow$$

$$bel(x_t) = \eta_t p(z_t | x_t) \overline{bel}(x_t)$$

```

1: Algorithm Bayes_filter(bel(x_{t-1}), u_t, z_t):
2:   for all x_t do
3:      $\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx$ 
4:     bel(x_t) =  $\eta p(z_t | x_t) \overline{bel}(x_t)$ 
5:   endfor
6:   return bel(x_t)

```

```

1: Algorithm Discrete_Bayes_filter({p_{k,t-1}}, u_t, z_t):
2:   for all k do
3:      $\bar{p}_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1}$ 
4:      $p_{k,t} = \eta p(z_t | X_t = x_k) \bar{p}_{k,t}$ 
5:   endfor
6:   return {p_{k,t}}

```

알고리즘으로 구현 하면 된다.