
Portfolio 2023

꿈임없이 배우는
프로그래머,
임호종입니다

01

Watchdog



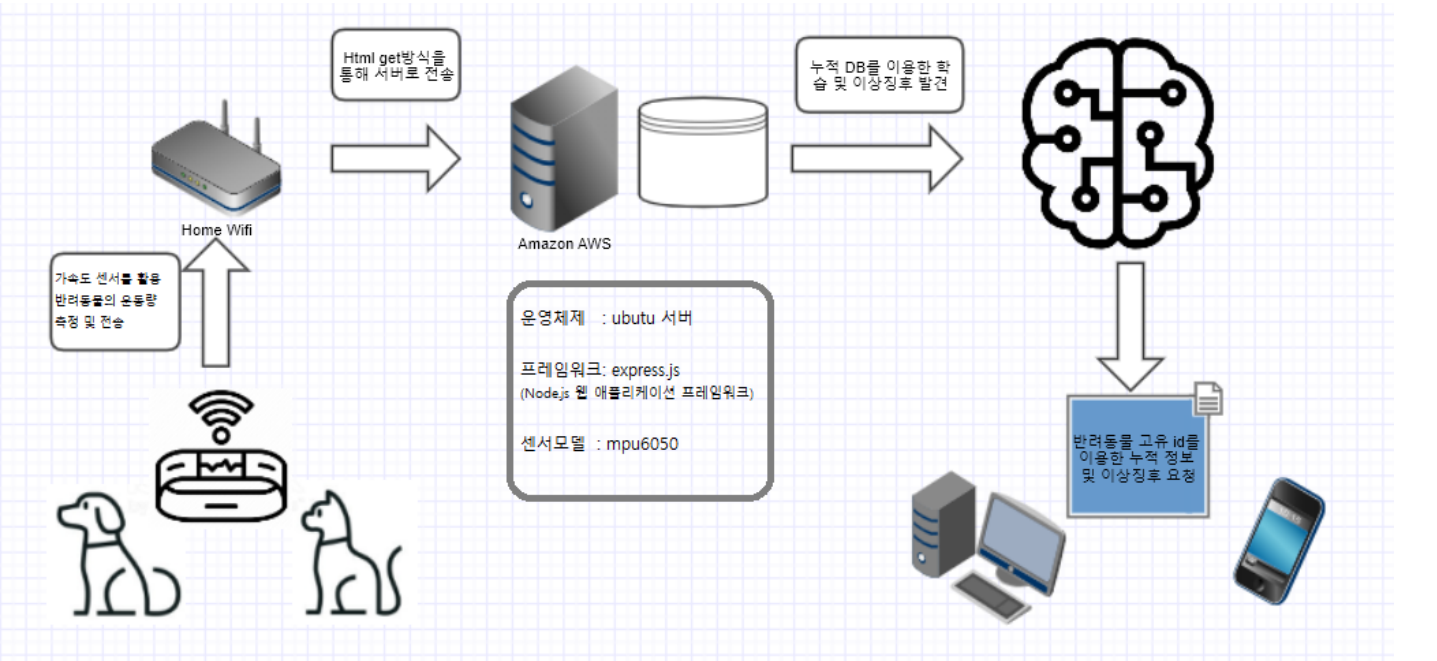
프로젝트 기간 : 2020.03 - 2020. 06

Watchdog (반려동물용 IoT 디바이스)

반려동물의 상태와 운동량을 알려주는 웨어러블 디바이스

IoT 디바이스를 개발하여 반려동물의 상황을 수집하고, 이에 대한 반려인들의 궁금증을 해결하는 것을 목표로 합니다. Esp32보드와 센서를 이용하여 데이터를 수집한 후, 클러스터링 알고리즘을 활용하여 수집된 데이터를 분석합니다. 디바이스는 와이파이나 스마트폰 핫스팟을 활용해 정보를 주고받습니다. 이러한 정보를 통해 반려동물을 더 잘 돌봄으로써 반려인들의 삶의 질을 높이기를 기대합니다.

담당 역할	하드웨어 및 백엔드
기여도	40%
개발 환경	Arduino, Node.js, AWS EC2, Ubuntu
구현 사항	Wifi 통신을 이용한 가속도 센서 정보 수집 데이터 관리를 위한 mysql 서버 구축 고객용 홈페이지 제작



Watchdog (반려동물용 IoT 디바이스)

반려동물의 상태와 운동량을 알려주는 웨어러블 디바이스

1. Arudino 통신

```
55 sprintf(append_value, "%f%f%f%f%f", loop_count%10, accX, loop_count%10, accY, loop_count%10, accZ, loop_count%10, calorie);
56 strcat(url, append_value);
57 loop_count++;
58 if(loop_count%10 == 0){
59     HTTPClient http;
60     http.begin(url);
61     int httpCode = http.GET();
62     // httpCode will be negative on error
63     if(httpCode > 0) {
64         // HTTP header has been send and Server response header has been handled
65         USE_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);
66         // file found at server
67         if(httpCode == HTTP_CODE_OK) {
68             String payload = http.getString();
69             USE_SERIAL.println(payload);
70         }
71     } else {
72         USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
73     }
74     http.end();
```

2. 메인 서버

```
26 app.get('/data',function(req, res) {
27     var query = connection.query("insert into acc (id, accX, accY, accZ, cal) values (" + req.query.id + "," + req.query.accX + "," + req.query.accY + "," + req.query.accZ + "," + req.query.cal + ");");
28     if(err) throw err;
29 });
```

```
75 app.get('/watchdog', function (req, res) {
76     var html = fs.readFile('./graph.html', function (err, html) {
77         html = " " + html
78         console.log('read graph.html');
79         if (req.query.id == '') var qstr = 'select * from acc_10 where time >= date_sub(NOW(), interval 24 hour)';
80         else var qstr = 'select * from acc_10 where time >= date_sub(NOW(), interval 24 hour) and id = ' + req.query.id;
81         //else var qstr = 'select * from acc_100 where id = ' + req.query.id;
82         connection.query(qstr, function(err, rows, cols) {
83             if (err) throw err;
84             //need to be selected from database
85             //idlex = XXXXX; idley = YYYYY; idlez = ZZZZZ;
86             //idle accelerometer value
87             var data = "";
88             var comma = "";
89             for (var i=0; i< rows.length; i++) {
90                 r = rows[i];
91                 data += comma + "[new Date(" + r.time.getFullYear() + "," + r.time.getMonth() + "," + r.time.getDate() + "," + r.time.getHours() + "," + r.time.getMinutes() + "," + r.time.getSeconds() + "); " + r.accX + "," + r.accY + "," + r.accZ + "," + r.cal + "], ";
92                 comma = ",";
93             }
94             res.send(html + data);
95         });
96     });
97 });
```

02

Strengton



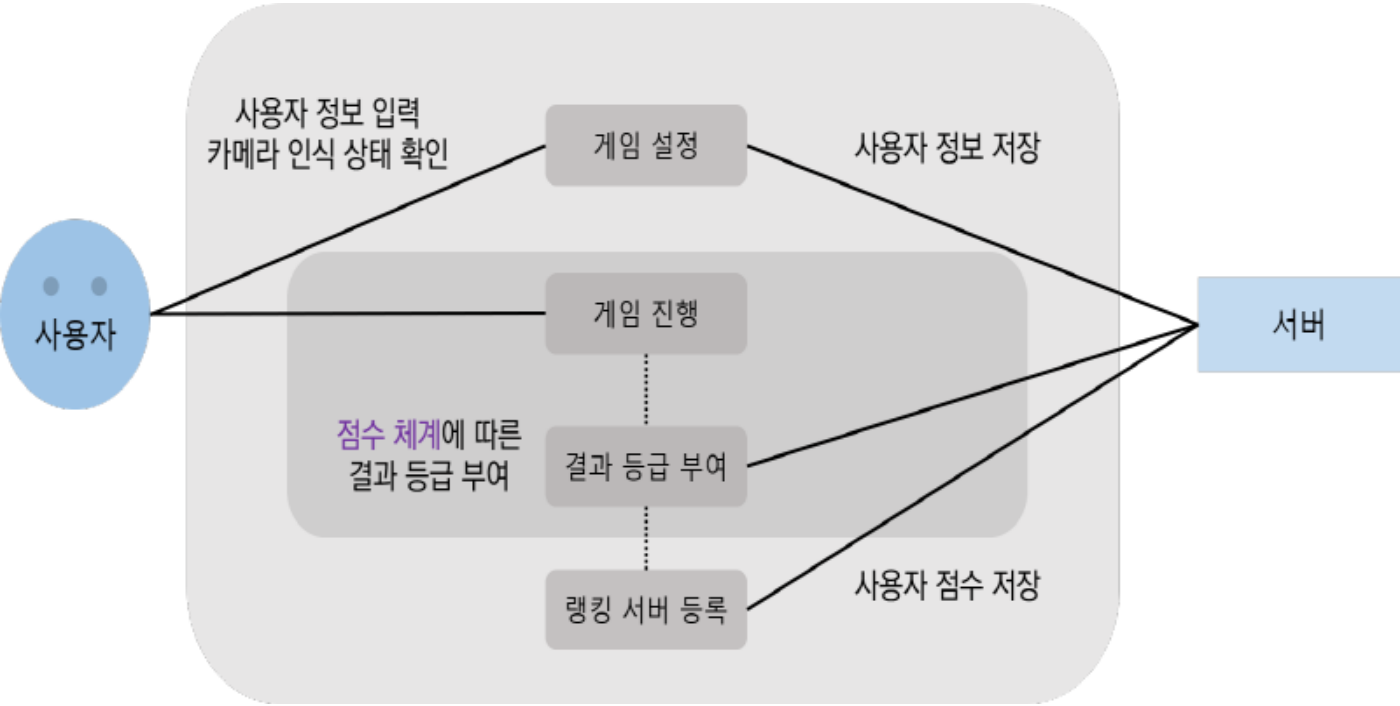
<https://youtu.be/cpm6jk2YmNs>

Strengton (웹캠을 활용한 코어 근육 단련 게임)

Skeleton + Stregth. 모션캡처를 통한 컨트롤로 단련

건강과 재미를 동시에 즐길 수 있는 게임으로, 코어 근육을 강화하는 자세들을 게임 안에서 구현하여 사용자가 따라할 수 있도록 제공합니다.

담당 역할	게임 구현
기 여 도	40%
개발 환경	Unity, Node.js, AWS EC2, Ubuntu
구현 사항	실시간 이미지의 자세 인식 핵심 부위 좌표의 차이를 통한 점수화 서버를 이용한 랭킹 시스템



Strengton (웹캠을 활용한 코어 근육 단련 게임)

Skeleton + Stregth. 모션캡처를 통한 컨트롤로 단련

1. 핵심 좌표 계산

```
public void PoseUpdate()
{
    // calculate movement range of z-coordinate from height
    var t1 = Vector3.Distance(jointPoints[PositionIndex.head.Int()].Pos3D, jointPoints[PositionIndex.neck.Int()].Pos3D);
    var t2 = Vector3.Distance(jointPoints[PositionIndex.neck.Int()].Pos3D, jointPoints[PositionIndex.spine.Int()].Pos3D);
    var pm = (jointPoints[PositionIndex.rThighBend.Int()].Pos3D + jointPoints[PositionIndex.lThighBend.Int()].Pos3D) / 2f;
    var t3 = Vector3.Distance(jointPoints[PositionIndex.spine.Int()].Pos3D, pm);
    var t4r = Vector3.Distance(jointPoints[PositionIndex.rThighBend.Int()].Pos3D, jointPoints[PositionIndex.rShin.Int()].Pos3D);
    var t4l = Vector3.Distance(jointPoints[PositionIndex.lThighBend.Int()].Pos3D, jointPoints[PositionIndex.lShin.Int()].Pos3D);
    var t4 = (t4r + t4l) / 2f;
    var t5r = Vector3.Distance(jointPoints[PositionIndex.rShin.Int()].Pos3D, jointPoints[PositionIndex.rFoot.Int()].Pos3D);
    var t5l = Vector3.Distance(jointPoints[PositionIndex.lShin.Int()].Pos3D, jointPoints[PositionIndex.lFoot.Int()].Pos3D);
    var t5 = (t5r + t5l) / 2f;
    var t = t1 + t2 + t3 + t4 + t5;
```

2. 오차 점수화

```
for (int i = 0; i < 28; i++)
{
    warriorbalance[PoseIndex, i] += center_difference;
    difference = Vector3.Distance(warriorbalance[PoseIndex, i], jointPoints[i].Pos3D);
    joint_difference += difference;
    if (i >= PositionIndex.rShldrBend.Int() && i <= PositionIndex.rMid1.Int())//5
        r_arm += difference;
    if (i >= PositionIndex.lShldrBend.Int() && i <= PositionIndex.lMid1.Int())//5
        l_arm += difference;
    if (i >= PositionIndex.lEar.Int() && i <= PositionIndex.Nose.Int())//5
        face += difference;
    if (i >= PositionIndex.rThighBend.Int() && i <= PositionIndex.rToe.Int())//4
        r_leg += difference;
    if (i >= PositionIndex.lThighBend.Int() && i <= PositionIndex.lToe.Int())//4
        l_leg += difference;
    if (i >= PositionIndex.abdomenUpper.Int() && i <= PositionIndex.spine.Int())//5
        body += difference;
}
joint_difference = 10000 - joint_difference;
MaxScore = MaxScore < joint_difference ? joint_difference : MaxScore;
```

03

개인 프로젝트

PC 던파 100Lv 조합기

세트 점수 기반 최적 경우의 수 순회

새로운 세트 조합 시스템과 다양한 신규 아이템이 추가되어 최종 데미지 계산이 복잡해진 상황에서, 11종류의 데미지 증가, 크리티컬 데미지 증가, 스킬 공격력 증가 등의 스탯을 계산해 최종 데미지를 쉽게 파악할 수 있도록 도와주는 Python프로그램입니다.

https://github.com/noranbanana/dnf_100lv_set/blob/master/100lv.py

서버 ...

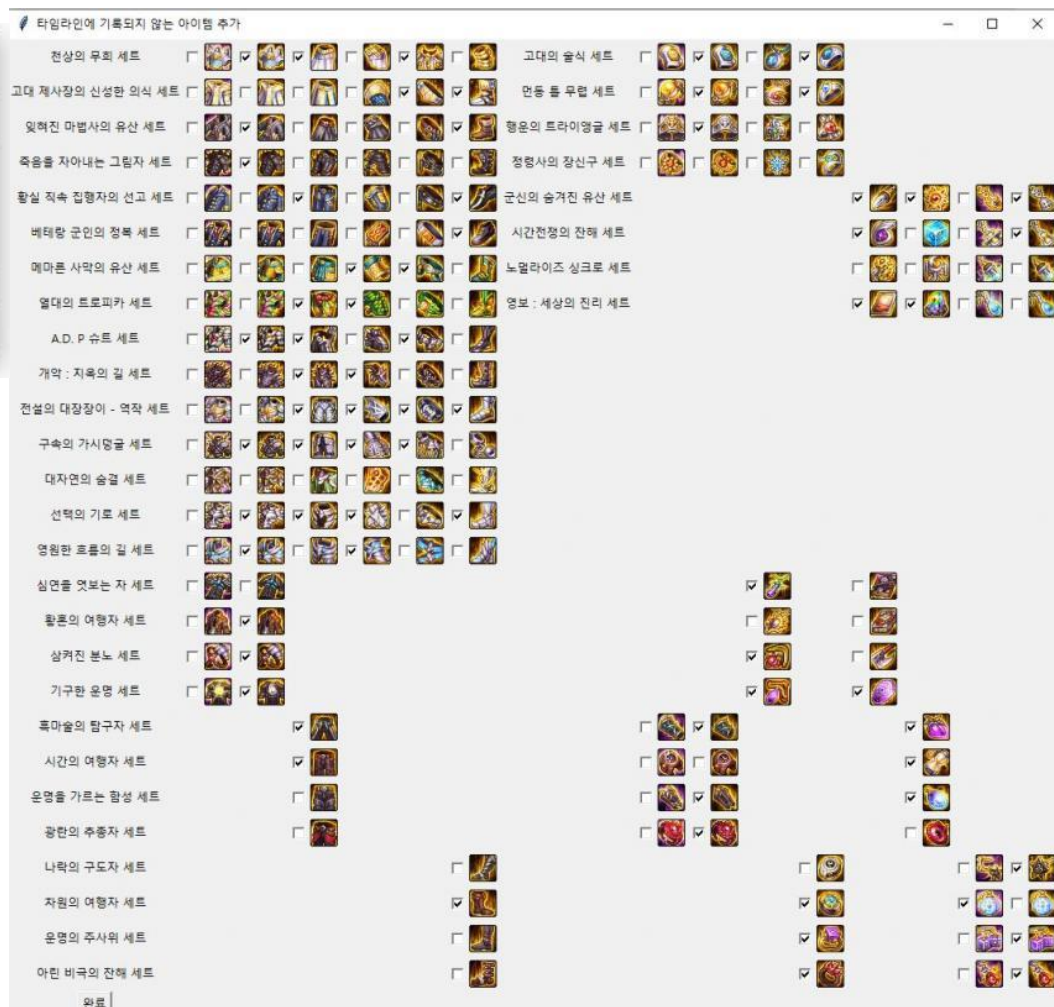
닉네임

무기

get

OpenAPI를 활용,
획득한 아이템을 체크하고

초월, 정가 등을 고려하여
추가적으로 선택할 수 있게
만들었습니다.



```
85 now = datetime.datetime.now()
86 nowDatetime = now.strftime('%Y%m%d')
87 for cd in timeline_code:
88     dungeon_url = "https://api.neople.co.kr/df/servers/" + server_name + "/characters/" + char_id + "/timeline?limit=100&code=" + cd + "&startDate=2"
89     dungeon_response = requests.get(dungeon_url)
90     dungeon_result = json.loads(dungeon_response.text)
91     for epic_drop in dungeon_result['timeline']['rows']:
92         item_list.append(epic_drop['data']['itemName'])
93     if dungeon_result['timeline']['next']:
94         while True:
95             dungeon_url = "https://api.neople.co.kr/df/servers/" + server_name + "/characters/" + char_id + "/timeline?" + "next=" + \
96                 dungeon_result['timeline']['next'] + "&apikey=NwrkuMSx3YoOmFLLM5y1RdQTyPsX0vtH"
97             dungeon_response = requests.get(dungeon_url)
98             dungeon_result = json.loads(dungeon_response.text)
99             for epic_drop in dungeon_result['timeline']['rows']:
100                 item_list.append(epic_drop['data']['itemName'])
101             if not dungeon_result['timeline']['next']:
102                 break
103
```

PC 던파 100Lv 조합기

세트 점수 기반 최적 경우의 수 순회

1. 아이템 옵션 수치화

```
15 set_list = {'현상의 무회 세트': {'setItemId': '72b9d0625ac8f77c16e4fb3a329235a5', 'setItems': [{'slotId': 'JACKET', 'slotName': '상의', 'itemId':  
16 option_dict = {'고대 제사장의 로브': [0, 0, 0, 0, 12.0, 10, 10, 0, 0.0, 0.0, 0], '대제사장의 예복': [0, 0, 10, 0, 12.0, 21, 20, 0, 0.0, 0.0, 0], '  
17 set_option_dict = {'고대 제사장의 신성한 의식 세트2': [21.0, 0, 0, 0, 0, 0, 8.0, 0, 0, 0.0, 0], '고대 제사장의 신성한 의식 세트3': [21.0, 0, 21.0, 0, 0, 0, 0, 0, 0, 0.0, 0]}
```

2. 세트 구성

```
for items in candidate[candi_level]:  
    tmp = {}  
    for part in parts:  
        tmp[part] = ''  
  
    if len(items) == 2:  
        tmp['score'] = 2.0  
    if len(items) == 3:  
        #if len(set_list[candi_level])  
        #    tmp['score'] = 5.0  
        #else:  
        #    tmp['score'] = 4.0  
        tmp['score'] = 5.0  
    if len(items) == 5:  
        tmp['score'] = 7.0
```

```
410 for complete_lists in sets_list:  
411     if complete_lists['score'] > best_score:  
412         best_score = complete_lists['score']  
413         secondary = best_sets[:]  
414         best_sets = [complete_lists]  
415     if complete_lists['score'] == best_score:  
416         if not complete_lists in best_sets:  
417             best_sets.append(complete_lists)  
418
```

3. 최종 데미지 비교

```
482 elemental = (1.05 + (default[10]*0.0045))  
483 add_dmg = default[2]/100.0 + default[3]*elemental/100.0 + 1  
484 final_score *= elemental * add_dmg * (default[8]/100.0)  
485 for i in range(0, 10):  
486     if i in [0, 1, 4, 5, 6, 7, 9]:  
487         final_score *= (default[i]+100.0)/100.0  
488
```



채팅 및 KEY:VALUE 소켓 프로그램

C 소켓 활용 네트워킹

멀티스레드와 소켓을 활용한 간단한 네트워크 프로그램입니다.

Server

```
// mutex
pthread_mutex_lock(&mtx);
for (int i = 0; i < MAX_CLIENT_CNT; i++)
{
    if (client_sock[i] != 0)
    {
        write(client_sock[i], buf, strlen(buf));
    }
}
pthread_mutex_unlock(&mtx);
```

Client

```
// chat_cli.c
sprintf(buf, "%s %s", name, msg);
write(sock, buf, strlen(buf));
```

클라이언트에서 보낸 메시지를

접속한 모든 유저에게 전송합니다.

```
yong@hostname:~/바탕화면/hojong/0411$ ./chat_ser.out
Wait for next client...
INFO :: Connect new Client (ID : 0, IP : 127.0.0.1)
INFO :: Connect new Client (ID : 1, IP : 127.0.0.1)

yong@hostname: ~/바탕화면/hojong/0411
yong@hostname:~/바탕화면/hojong/0411$ ./chat_cli.out im
hello
[im] hello
[hojong] hi

yong@hostname:~/바탕화면/hojong/0411$ ./chat_cli.out hojong
[im] hello
hi
[hojong] hi
```

채팅 및 KEY:VALUE 소켓 프로그램

C 소켓 활용 네트워킹

```
char filepath[1024];
sprintf(filepath, "data/%s", key);
char reply[300];
if (flag == 1){
    if(access(filepath, F_OK) == 0){
        sprintf(reply, "replacing key '%s'\nsaved '%s'", key, value);
    }
    else{
        sprintf(reply, "creating key '%s'\nsaved '%s'", key, value);
    }

    FILE *fp = fopen(filepath, "w");
    if (fp == NULL){
        sprintf(value, "파일명을 다시 확인해주세요");
    }
    else{
        fwrite(value, strlen(value), 1, fp);
        fclose(fp);
    }
}
else{
    FILE *fp = fopen(filepath, "r");
    if (fp == NULL){
        sprintf(reply, "no key found");
    }
    else{
        fread(value, 99, 1, fp);
        fclose(fp);
        sprintf(reply, "reading key '%s'\nvalue: %s", key, value);
    }
}

write(client_sock[id], reply, strlen(reply));
pthread_mutex_unlock(&mutex);
```

전송받는 KEY:VALUE 혹은 KEY를
상황에 맞게 저장, 수정, 읽으면서
통신합니다

Server

```
ubuntu@ip-172-31-9-217:~/ppt_server$ ./server
Wait for next client ...
INFO :: Connect new Client (ID : 0, IP : 43.130.55.130)
INFO :: Connect new Client (ID : 1, IP : 43.130.55.130)
INFO :: Disconnect with client.. BYE
INFO :: Disconnect with client.. BYE
```

Client A

```
ubuntu@ip-172-31-9-217:~/ppt_server$ ./client
>>connect 43.130.55.130 8080
>>save BOSS:13055
creating key 'BOSS'
saved '13055'
>>read BOSS
reading key 'BOSS'
value: 13055
>>save BOSS:23502
replacing key 'BOSS'
saved '23502'
>>read BOSS
reading key 'BOSS'
value: 23502
>>
```

Client B

```
ubuntu@ip-172-31-9-217:~/ppt_server$ ./client
>>connect 43.130.55.130 8080
>>read BOSS
reading key 'BOSS'
value: 23502
>>
```