



PORTFOLIO

한국기술교육대학교 컴퓨터공학부 양우민





목 차

▶ 자기소개

1. 프로그래머로서	3
2. 직원으로서	4

▶ 프로젝트

1. Get Weapons	6
2. Car Talk	11
3. Smart Lens Case	16

▶ 기타 프로그램

1. 멀티스레드 기반 채팅 프로그램	20
2. 온라인 3D 테트리스	20
3. Medicine Database Program	21
4. 기타 활동	21



▶ 자기소개

1. 프로그래머로서

프로그래머가 되기 위하여 어렸을 때부터 꿈꿔 왔던 것은 아니다. 나 역시 다른 이들과 마찬가지로 좋은 대학 진학을 목표로 한 자연계 고등학생이었다. 심지어 대학교와 학부 또한 성적에 맞추어 진학했는데 우연찮게 한국기술교육대학교 컴퓨터공학부에 들어왔다. 굳이 컴퓨터공학부를 선택한 이유라 하면 게임하는 것을 좋아했고 컴퓨터가 친근했기 때문이다. 여기까지 아무 생각 없는 진로 선택이었지만, 대학생활 4년을 끝으로 나 자신과 프로그래머 사이에 연결고리를 조금씩 찾게 되었다. 대표적인 연결 고리들은 다음과 같다.

1) 성실과 집착

남들도 흔히 갖고 있는 성실이다. 하지만 성실의 정도는 사람마다 매우 다르다. 여기에 집착이라는 요소를 더해서 내 성실의 정도를 표현하고자 한다. “나에게 주어진 어떤 일이든 다른 누구도 아닌 내 마음에 들 때까지 한다.” 이다.

이러한 성실과 집착은 프로그래밍에서도 아주 중요한 요소라 생각한다. 성실함 없이는 컴파일러가 내뱉는 빨간 줄들을 해결할 수 없으며, 집착 없이는 프로그램의 복잡한 알고리즘을 설계 및 해석할 수 없고 그것으로부터 나오는 문제를 해결할 수 없기 때문이다.

2) 인정과 열정

사람의 욕구는 금전욕, 식욕, 성욕, 인정욕 등이 있다. 나의 모든 욕구는 거의 인정욕일 만큼 남들에게 매우 인정받고 싶어 한다. 그래서 프로그램을 만들 때에도 특히 완성도를 굉장히 중요시한다. 프로그램의 완성도는 그 프로그래머에 대한 이미지로 각인되기 때문이다.

여기에 열정이란 기름을 부으면 어떻게 될까? 나는 프로그래밍을 하면서 내 프로그램을 보고 인정하는 남들의 모습을 매번 상상한다. 그것으로부터 나오는 기대감이 나를 더욱 채찍질하며, 그것이 현실로 됐을 때 얻는 만족감과 쾌감은 이루 말할 수 없다. 이것이 바로 열정이라 생각한다.



2. 조직원으로서

프로그래머로서 역량뿐만 아니라 조직원으로서 역량 또한 매우 중요하다. 내 조직원으로서 역량은 지금까지 경험해온 아르바이트, 군대, 인턴, 팀 활동을 토대로 분석한 결과, 크게 3가지가 있다.

1) 살신성인

나의 살신성인 역량은 앞서 설명한 인정욕에 근거함이 틀림없다. 인정이라는 것이 애초에 혼자서 이루어 질 수 없는 요소로, 조직에 있는 타인이 나를 평가하기 때문에 존재할 수 있는 것이라 생각한다. 그래서 조직으로부터 인정받기 위해 개인 역량을 최대한 발휘하여 조직에 많이 공헌하기 위한 노력을 한다. 이를 증명할 수 있는 근거로, 지금까지 경험해온 어느 조직에서든지 나와 계속 함께 일하는 것을 요구했었고 직위 또한 영향이 있었다. 호프집에서 아르바이트생으로 들어가 1년 중 6개월을 매니저로 근무했었다. 군에서 일반 병역을 마치고 전문하사로 1년을 더 근무했었다. 학교 졸업요건에 해당하는 인턴도 의무기간을 마친 후에도 약 2개월을 추가로 진행했었다. 현재 학교 앞 마트에서 근무하는데, 1년 6개월 차이로 아르바이트생으로 들어가 10개월 째 점장님 휴직 날만 대체근무를 진행하고 있다. 이러한 경험을 토대로 나의 조직원으로서 가장 큰 역량은 살신성인 정신이라는 것을 알 수 있다.

2) 리더쉽

리더쉽 역시 살신성인 정신이 깃들여 있다면 자연스레 표출되는 역량이라 생각한다. 여기선 주로 팀 활동 중점으로 설명하고자 한다.

지금까지 거의 모든 팀 활동에서 팀장의 역할을 수행해 왔다. 대부분 자진해서 그 역할을 수행했는데, 이것부터가 리더로서의 기본적인 자질이지 않을까 생각한다. 그리고 그 팀이 존재하는 이유와 궁극적으로 이루고자 하는 목적과 방향을 명확하게 한다. 또한, 목적을 이루기 위해 팀 스케줄을 계획하고 그대로 실천하고자 노력한다. 추가적으로, 각 팀원의 장점과 단점을 파악하고 그것에 걸맞는 임무를 배분하며, 그 결과를 꼼꼼히 짚고 넘어가는 스타일이다.

팀 활동에서 가장 빛을 발했던 적이 “소프트웨어 공학”이라는 과목에서 이다. 팀 활동의 목적은 구조적 개발 방법론을 적용하여 팀에서 정한 주제를 설계 단

계까지 구현하는 것이었다. 프로젝트 계획 단계에서 문제 정의와 일정 계획, 작업 분해, 조직 계획 등을 통해 팀을 이끄는 것은 당연, 요구 분석에서 자료 흐름도와 구조적 설계에서 시스템 구조도 작성 시에도 팀원들의 많은 노력이 필요할 만큼 꼼꼼하게 진행하였었다. 팀원들이 임무에 대해 방향성을 잃거나 힘들어 할 때에도 올바른 방향으로 진행하도록 지도해 주었으며, 그들의 장점이 팀에 필수적임을 호소하여 그들이 팀에 더욱 헌신할 수 있도록 이끌었다. 이러한 팀원들 노력의 결실로, 엄격하고 독설로 유명한 해당 과목 교수님께 유일하게 긍정적인 평가를 받은 팀으로 남았고, 차후 다른 과목에서도 그 팀원들과 같이 팀 활동을 하게 되는 좋은 관계를 유지하고 있다.

3) 커뮤니케이션

지금까지 내용을 통해 리더로서 우직한 모습은 확인할 수 있었다. 이에 더불어 남을 배려하고 존중해주려는 성격이 있기 때문에, 그것에 걸 맞는 커뮤니케이션 역량에 대해 소개하려 한다. 사실, 팀원들과 커뮤니케이션 하는 것을 좋아한다. 특히, 프로그래밍적인 내용의 타당성, 합리성, 이상성을 팀원들과 같이 커뮤니케이션 하는 것을 매우 좋아한다. 심지어 취업 상담 때에도 상담자가 물어본 “하루 중에 제일 즐거운 때는 언제인가?” 라는 질문에, 내 여가 시간을 가질 때도 아닌, 팀 활동을 통해 팀원들과 커뮤니케이션을 나눌 때라고 답변했었다. 또한 인턴 근무할 때에도 2개월의 추가 근무를 제의받았는데, 이유 중 하나가 부대표님(1인 개발자) 사이에 커뮤니케이션이 좋아서였다고 대표님께 들은 바 있다. 부대표님 혼자 개발할 당시는 고되고 외로워 보였는데, 나와 함께 업무를 하면서 업무적인 소통으로 분위기가 활발해지고 즐거워 보였다고 한다. 이러한 커뮤니케이션의 즐거움은 상대를 설득시킬 때 얻는 쾌감과 상대 의견을 납득함으로써 동질감을 형성하는 것으로부터 생기는 것이라 생각한다. 이로써 커뮤니케이션 또한 직원으로서 나의 강력한 장점이라고 생각한다.



▶ 프로젝트

대학 생활 동안에 수행한 대표적인 프로젝트들을 소개하려고 한다.

1. Get Weapons

1) 프로젝트 설명

이 프로젝트는 2016년 “졸업설계1”, “졸업설계2” 과목의 졸업작품으로 진행되었다. 유니티 개발환경에서 제작되었으며, 안드로이드 플랫폼을 기반으로 한 1:1 FPS 게임이다.

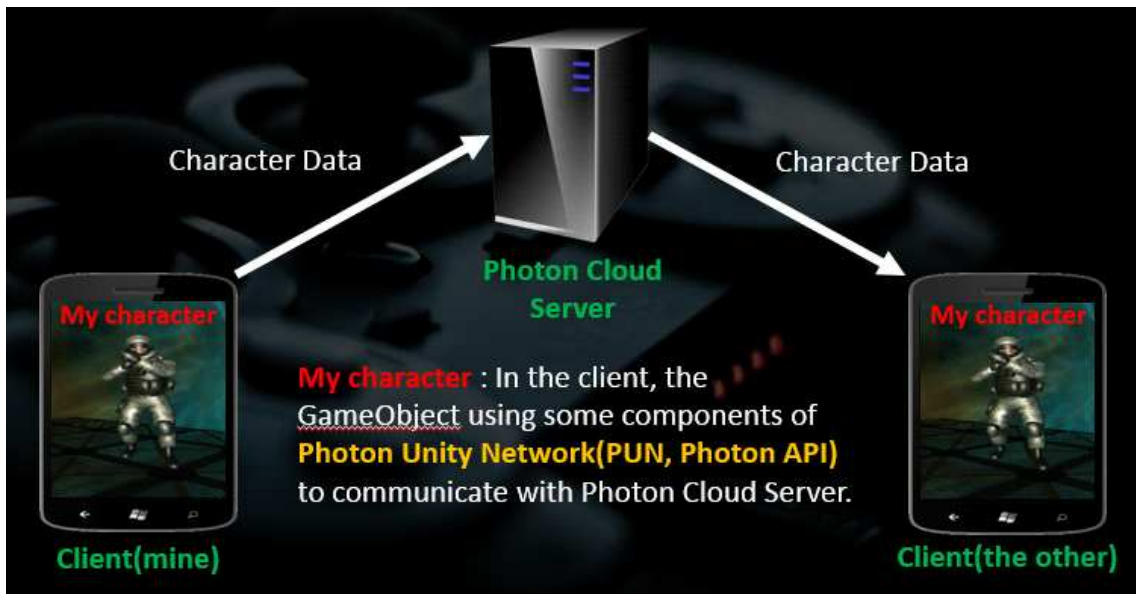
졸업작품으로서 이 프로젝트의 의의는 기존에 출시되어 있는 모바일 FPS 게임들이 사용자 인터페이스로 거의 터치스크린만을 사용한다는 점을 극복하고자 스마트폰의 여러 가지 센서들을 적용해 보는 것이다. 적용 방법은 아래와 같다.

- (1) 가속도 : 캐릭터 이동
- (2) 자이로스코프 : 재장전
- (3) 근접 : 줌 인/아웃
- (4) 터치스크린 : 캐릭터 회전
- (5) 왼쪽 버튼 : 앞기/서기
- (6) 오른쪽 버튼 : 발사

위의 센서들을 사용하려면 유니티에서 기본적으로 제공하는 API를 사용하면 된다. 하지만 근접 센서는 API로 제공되지 않기 때문에 유니티 플러그인을 통해서 native로 접근하는 export한 자바 라이브러리 소스가 필요하다. 그 소스는 안드로이드 스튜디오에서 작성하였으며, 상대방부터 피격을 받았을 때 바이브레이션을 발동시키는 바이브레이터 관련한 부분 또한 포함하고 있다.

게임 컨셉은 1:1 대전으로부터 각 라운드마다 승자에게 무기와 랭크 점수가 지급되는 것이다. 무기는 기본 무기보다 강력하며 랜덤으로 그 종류가 결정된다. 획득한 무기로 다음 라운드에서 이길 확률을 높일 수 있으며, 결국 자신의 랭크를 높일 수 있다. 라운드에서 이길 수 있는 방법은 상대를 사살하거나 보급 상자를 먼저 얻는 유저가 승리하게 된다. 그리하여 게임 이름이 “Get Weapons” 인 것이다.

2) 시스템 구성도



클라이언트 플랫폼으로 안드로이드를 사용한다고 언급했었다. 위에 그림에서 확인할 수 있듯이, 서버는 유니티에서 가장 많이 사용하는 포톤 클라우드 서버를 활용하였다. 포톤 클라우드 서버는 동시 접속자 20명까지는 무료로 제공이 되며, 유니티 개발환경에서 “PUN(Photon Unity Network)” 라는 API를 사용해서 접근할 수 있다.

포톤 클라우드 서버를 사용한 동기화 방식은 간단하다. 위에 그림과 같이 왼쪽 클라이언트의 캐릭터 데이터를 오른쪽 클라이언트에 있는 왼쪽 클라이언트 캐릭터로 동기화하는 방식이다. 동기화 종류는 크게 “PhotonSerializeView” 라는 초당 여러번 동기화하는 방법과 “RPC(Remote Procedure Calls)” 라는 상대 클라이언트의 함수를 호출하는 방법이 있다. RPC는 특정 시점에 동기화하고 싶은 데이터를 동기화하기에 유용한 방법이다. 동기화할 데이터는 캐릭터 위치와 회전, 상태 등이 있으며, FPS 특성상 실시간을 보장해야 하므로 PhotonSerializeView를 적용했다. 그리고 게임의 승패 여부를 RPC를 통하여 동기화하도록 구현하였다.



다음은 이 프로젝트에 적용한 구체적인 동기화 기법이다. 앞에서 설명한 동기화 방식은 포톤 클라우드 서버를 사용하기 위한 동기화의 기본적인 메커니즘이라면, 이번은 FPS 게임으로서 적용한 동기화 메커니즘이라 할 수 있겠다. 캐릭터가 취할 수 있는 상태는 발사, 재장전, 줌 인/아웃, 앞기/서기, 피격 등이 있다. 여기서 제일 중요하게 동기화할 요소는 발사와 피격, 그리고 승패이다. 아래와 같은 순서로 동기화를 진행한다.

- (1) 내 클라이언트에서 내 캐릭터가 발사를 한다. 만약 상대 캐릭터가 피격을 입으면 데미지는 입지 않고 피격 위치만 1초 동안 기억하도록 한다.
- (2) 상대 클라이언트에서 내 캐릭터도 발사를 한다. 상대 플레이어 입장에서는 자기 캐릭터가 피격을 입는 상황이 발생한다.
- (3) 다시 말해, 내 클라이언트에서 내 캐릭터가 피격을 받으면 체력이 피격 받은 위치에 따라 감소하게 되고 0 이하가 되면 죽는다.
- (4) 내 클라이언트에서 내 캐릭터가 피격을 받거나 죽는 것은 캐릭터의 상태가 되며, 그 상태를 상대 클라이언트에 있는 내 캐릭터의 상태로 적용한다.
- (5) 1번 단계에서 피격 위치를 기억하고 있는 캐릭터의 상태가 피격 상태가 되면 피격 애니메이션과 피격 위치에서 출혈이 발동한다.
- (6) 내 캐릭터가 죽었을 경우 자신이 패배한 것으로 결정되며 RPC를 통해 상대 클라이언트에게 상대가 승리했다는 것을 전달한다.
- (7) 보급 상자를 먼저 얻는 경우도 RPC를 사용하여 상대의 패배로 결정한다.
- (8) 게임 도중에 게임을 나가게 되면 패배로 결정되며, 상대방의 접속이 끊겼을 경우 승리하는 것으로 결정한다.



끝으로 부가적인 시스템 구성 요소들에 대해 설명하고자 한다. 앞서 설명한 내용 중에 랭크와 획득한 무기에 대한 저장 방법이다. 서버는 연구실에 있는 내 개인 PC의 LAMP(Linux, Apache, Mysql(MariaDB), PHP)로 구현하였다. 클라이언트에서 서버로 HTTP 프로토콜 방식으로 통신하며, 데이터베이스에서 테이블은 총 3개가 있다. 사용자 ID와 랭크 점수를 저장하는 테이블이 있고, 사용자에게 따른 무기와 그 수량, 코스튬과 그 수량에 대한 테이블 2개가 있다. 물론 이 2개의 테이블이 처음 설명한 테이블의 사용자 ID를 외래키로 참조하여 무결성을 유지하도록 했고, 추가 사용자 생성 시에 트리거를 사용해 기본 의상과 무기가 주어지도록 구현하였다. 더 나아가 게임 알고리즘 상, 롤백해야 할 상황이 필요했지만 시간적 여유가 없어서 그 부분을 다른 방법으로 대체하였다.

3) 한계점 및 개선 사항

부족한 개발 시간 때문에 충분한 테스트가 이루어지지 않았다. 특히 모바일에서 네트워크 상태에 따른 대처를 거의 할 수 없었고, 개인 서버에 도메인 신청을 하지 않았기 때문에 LTE 망으로는 게임이 불가능하다. 또한, 캐릭터 상태로 직에 대해서 충분한 테스트 과정을 거치지 않았기 때문에 의도치 않은 캐릭터의 상태가 간혹 발견되기도 하였다. 마지막으로, 이 프로젝트에 적용한 동기화 기법은 클라이언트 중심 동기화 기법이기 때문에 서버 중심 동기화 기법보다 동기화가 엄격하지 못할 것이라는 개인적인 생각이 있다.

4) 발전 방향

발전 방향으로서는 적용한 여러 센서들의 감도를 사용자가 직접 설정할 수 있는 환경을 제공하는 것이 있다. 그리고 각 무기의 특징을 데미지 외에도 연사속도나 총알 개수 등을 다양화 하는 것도 있다. 또한, 근접 센서를 보다 사용하기 쉽게 사용자 인터페이스적인 고려를 통해 적합한 알고리즘을 개발하는 것이다. 마지막으로, 지금은 게임 동기화를 서버가 아닌 클라이언트 수준에서 결정하도록 만든 시스템이다. 포톤 클라우드 서버의 역할은 단지 클라이언트가 동기화하고 싶어 하는 데이터만 넘겨주는 형식인데, 내가 직접 서버 플랫폼을 구현한다면 현장에서 근무할 때 많은 도움이 되지 않을까 생각한다. 보통 실제 게임들은 동기화할 중요한 요소들을 서버가 판단해서 각 클라이언트로 동기화 해주는 것으로 알고 있다. TCP/IP(실제로는 UDP) 방법을 사용하여 소켓 통신으로 클라



이연트들을 관리하고 동기화할 데이터들을 멀티스레드를 사용하여 처리 및 송·수신한다면 충분히 가능해 보이기도 하다.

5) 개발 기간

연구내용 \ 월	1월	2월	3월	4월	5월	6월	7월	8월
기존 제품 조사 및 분석								
주제 선정								
UI, 오브젝트 디자인, 효과								
서버/클라이언트								
test/보완								

(2016년)

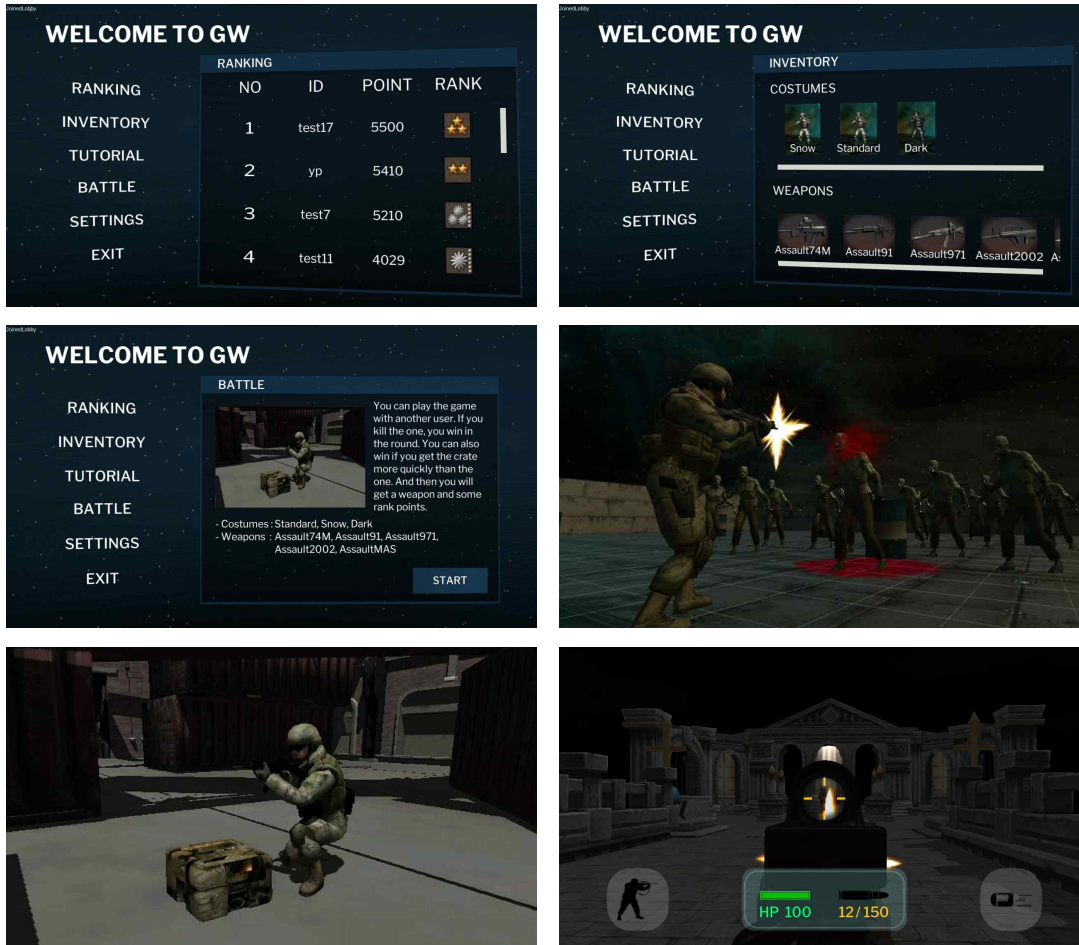
6) 역할 분담

직책	팀원	역할
팀장	양우민	팀 운영, 게임 기획, UI, 서버/클라이언트, 데이터베이스
팀원	팀원1	맵 제작, 충돌 제어
팀원	팀원2	튜토리얼 모드, UI

7) 기타

- 개발 환경 : Unity 5.x, Android Studio, Visual Studio 2015, vi 편집기
- 개발 언어 : C#, Java, PHP5
- 기타 : MariaDB(DBMS), NEXUS 6 사용

8) 구현 결과



– 소스 코드

해당 repository는 유니티 프로젝트의 Assets 폴더로 디렉토리를 설정하였다.
Github에서 팀원들이 각자 Scene과 스크립트 파일을 구분하여 협업하였다.

C# : <https://github.com/YangWoomin/Assets>

Java & PHP :

https://github.com/YangWoomin/Portfolio/tree/master/Get_Weapons

– 시연 영상

<https://www.youtube.com/watch?v=0Kbl6GyQW1U&feature=youtu.be>

2. Car Talk

1) 프로젝트 설명

이 프로젝트는 2016년 1학기 “임베디드 응용 및 실습” 과목의 팀(Term) 프로젝트로 진행되었다. 프로젝트 취지는 컴퓨터공학부 4학년 수준에 맞게 지금까지 배워온 것들을 최대한 활용하는 것과 임베디드로서의 리눅스 커널 모드에 접근해 보는 것이다. 프로젝트 최소 스펙(Specification) 요구 사항은 라즈베리파이가 다른 디바이스(서버나 다른 플랫폼들)로부터 수신한 문자열을 도트 매트릭스(Dot Matrix)로 출력하는 것이다. 다른 플랫폼과의 통신은 이미 구현이 되어 있는 소프트웨어 플랫폼인 웹 서버(Apache, Node.js 등) 등을 사용하지 않고 직접 송수신 모듈을 구현해야 한다. 또한, 라즈베리파이에서 수신한 문자열을 출력할 때, 이미 제작되어 있는 라이브러리(WiringPi 등)를 단지 사용하는 것만이 아닌, 실제 물리 메모리까지(GPIO 핀 주소 영역) 접근하는 과정을 설명할 수 있어야 하거나, 직접 디바이스 드라이버 모듈을 직접 구현하면 된다. 마지막으로 자신이 생각해낸 아이디어를 프로젝트에 적용하여 독창성을 보여주는 것이다.

위의 최소 요구 사항과 아이디어를 접목한 것이 “Car Talk” 이라는 프로젝트 명으로, 자동차 후방 유리에 도트매트릭스를 장착하고 스마트폰과 연동하여 뒤에 있는 차량에게 메시지를 전달하는 프로젝트로 진행하였다. 위급한 상황이나 자신이 알고자 하는 상황을 빠르게 전달하는 것이 목적이다. 아래 사진은 이 아이디어의 영감을 얻게 된 표본이다.



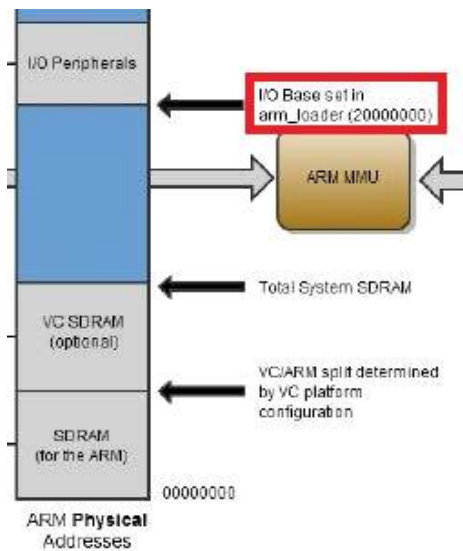
2) 시스템 구성도



시스템 구성도는 위의 사진과 같다. 사용자는 스마트폰을 통하여 서버에 접속하고 자신이 기존에 저장한 메시지를 받아 올 수 있다. 서버는 LAMP(Linux, Apache, Mysql(MariaDB), PHP)를 사용한다. (이 부분은 최소 스펙 요구 사항이 아닌 부가적인(Optional) 부분에 해당된다.) 사용자는 메시지를 추가하거나 삭제할 수 있다. 또한, Daum의 STT(Speech To Text) Open API를 사용하기 때문에 사용자 음성을 문자열로 변환도 가능하다. 이러한 문자열을 스마트폰에서 라즈베리파이로 전송시킬 수 있는데, 전송 방법은 블루투스로 하였다. 블루투스로 선택한 이유는 차량 내에서 스마트폰이 인터넷 연결을 위해 Ethernet을 사용할 수도 있고, 블루투스가 보다 그 용도에 적합하다고 생각했기 때문이다.

스마트폰이 블루투스를 통해 문자열을 송신할 때에 멀티스레드를 사용하도록 구현하였다. 안드로이드는 메인 스레드가 UI 쪽에 잡혀 있기 때문에 통신 모듈을 구현하기 위해서는 다른 스레드를 사용해야 한다. 라즈베리파이에서는 문자열을 수신하는 모듈을 파이썬으로 작성하였으며, 멀티프로세스로 동작하도록 구현하였다. 1개의 프로세스는 스마트폰으로부터 문자열을 수신해야 하고, 다른 1개 프로세스는 저수준 함수를 통해 디바이스 드라이버 모듈에 문자열을 넘

겨줘야하기 때문이다.



여기까지가 사용자 인터페이스 부분과 리눅스의 유저 모드 부분에 해당한다. 커널 모드 부분에 해당하는 디바이스 드라이버 모듈은 먼저 커널에 적재되어야 한다. 모듈이 커널에 적재되면 리눅스 가상 메모리에서 물리 메모리(ARM physical address) I/O peripherals 영역을 접근하는 부분을 사용할 수 있도록 ioremap 함수를 통해 커널로부터 권한을 얻어야 한다. 그 이후, 유저 모드로부터 저수준 함수를 통해 수신한 문자열을 도트매트릭스로 출력시킬 수 있는 비트로 변환한 뒤에, GPIO 레지스터에 접근하여 그 비트들을 HIGH/LOW 시켜 사용자가 알아볼 수 있는 문자열로 출력시킨다.

3) 한계점 및 개선 사항

라즈베리파이에서 스마트폰으로부터 메시지를 수신할 때 사용자 인증을 위한 과정이 있는데, 이 때 라즈베리파이에서 서버로 접근하는 부분이 Ethernet의 HTTP 프로토콜을 사용한다. 그래서 라즈베리파이가 인터넷에 연결되어 있지 않으면 문자열을 출력시킬 수 없는 상태이다. 이러한 보안적인 문제와 관련해서 ethernet을 사용하지 않고도 사용자 인증이 가능한 방법이 필요하다.

4) 발전 방향

라즈베리파이에서 항상 사용자의 문자열 메시지만 출력하는 것만이 아닌 날씨

정보나 광고도 출력하는 것으로 상업적으로도 보다 좋은 효과가 나올 것이라고 예상한다.

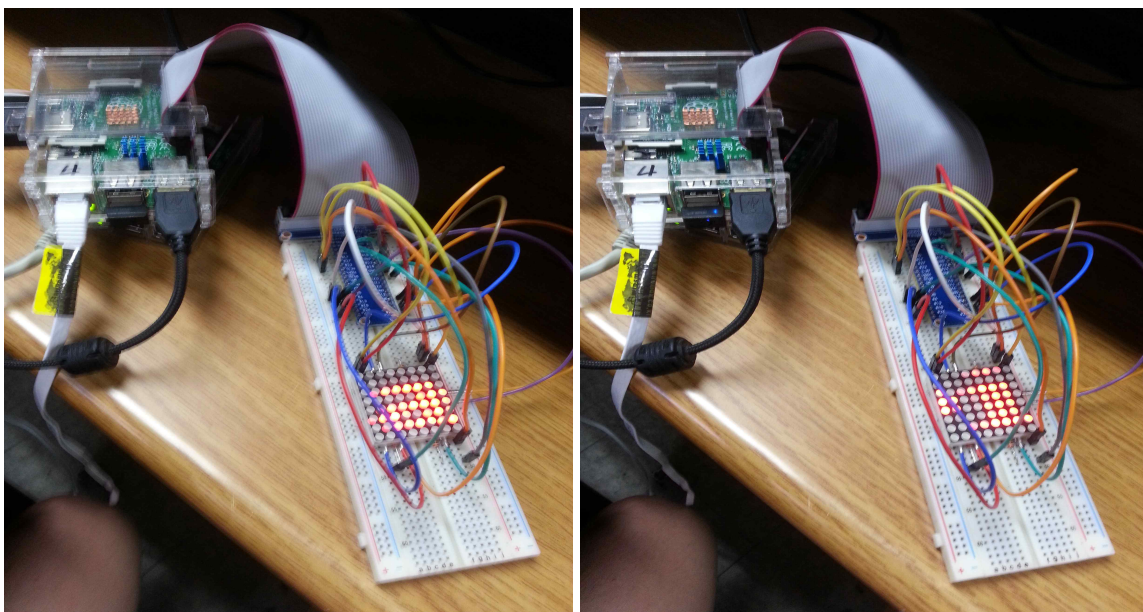
5) 역할 분담

개인 프로젝트

6) 기타

- 개발 환경 : Visual Studio 2015, Android Studio, vi 편집기
- 개발 언어 : C, Java, PHP5, Python
- 기타 : MariaDB(DBMS), 삼성 갤럭시 노트2 사용

7) 구현 결과



- 소스 코드

https://github.com/YangWoomin/Portfolio/tree/master/Car_Talk

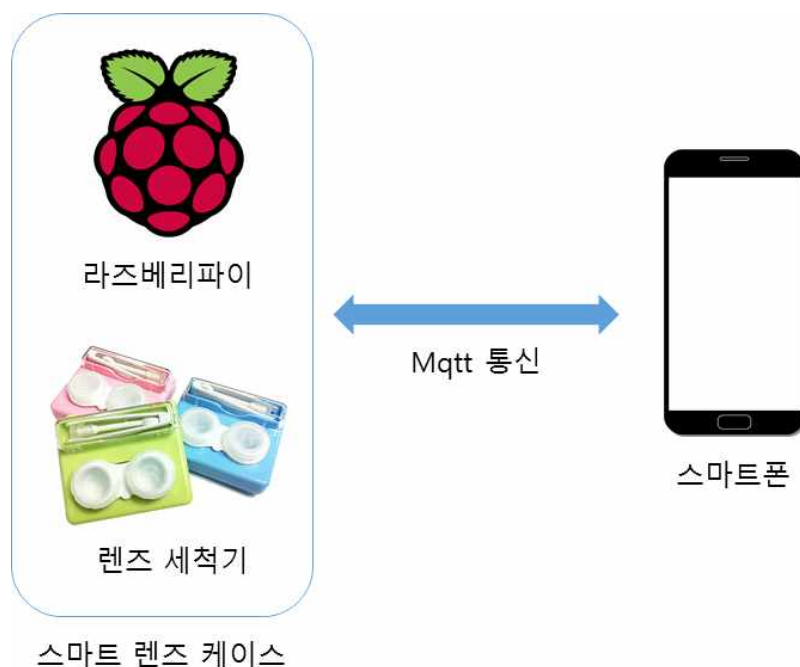
3. Smart Lens Case

1) 프로젝트 설명

이 프로젝트는 2016년 1학기 “IoT 개론 및 실습”이라는 과목에서 진행한 프로젝트이다. 렌즈 착용에 관련한 알림을 모바일로 받을 수 있는 시스템이다. 프로젝트 목적은 렌즈를 착용하는 바쁜 현대인들에게 하루 권장 착용시간, 유통기한, 위생을 신경 쓸 수 있도록 관리해주는 어플리케이션 개발이다. 알림 종류는 다음과 같다.

- (1) 하루 권장 착용시간 알림 : 보통 렌즈의 하루 권장 착용시간은 최대 8시간이다. 사용자의 렌즈 착용시간이 8시간이 초과되면 알림을 준다.
- (2) 유통기한 알림 : 렌즈는 1회용 렌즈부터 최대 2년까지 다양한 렌즈의 종류가 있다. 사용자가 어플리케이션에 새로 산 렌즈를 이름과 유통기한을 등록하면 유통기한 만기 시에 알림을 받을 수 있다.
- (3) 세척 알림 : 렌즈의 권장 세척 주기는 2~3일이다. 구성품인 렌즈 세척기로부터 렌즈의 세척 누락을 확인하고 세척 필요시기에 알림을 받을 수 있다.
- (4) 취침 전 알림 : 어플리케이션에 취침시간을 입력해 놓으면, 렌즈를 착용하고 있는 중이라면 취침시간에 렌즈 해제 알림을 받을 수 있다.

2) 시스템 구성도





- (1) 라즈베리파이 : 라즈베리파이(라즈비안 OS)는 스마트폰과 Mqtt 통신을 위해 Ethernet(Internet)으로 연결되어 있다. 그리고 렌즈 세척기에 달린 조그마한 렌즈 케이스의 유무를 판별하기 위해서 초음파 센서를 스마트 렌즈 케이스 천장에 부착하여 사용한다. 또한, 렌즈 세척기에 부착된 초음파 진동 센서를 동작시키기 위한 버튼이 라즈베리파이 GPIO 핀에 연결되어 있어서 세척을 수행하거나 버튼 눌림 여부를 확인할 수 있다.
- (2) 렌즈 세척기 : 렌즈 세척기는 부착된 버튼으로 초음파 진동 센서를 동작시켜 렌즈를 세척할 수 있다. 또한 렌즈를 휴대하기 간편하도록 조그마한 렌즈 케이스가 탈부착이 가능하게 되어 있는 형태이다. 이번 프로젝트에서는 총 3개의 렌즈 세척기를 배치해 두었다. 렌즈 세척기들은 스마트 렌즈 케이스 내부에 고정되어 있다.
- (3) 스마트폰 : 라즈베리파이와 Ethernet(Internet)으로 렌즈 관리를 위한 통신을 하며, 사용자 인터페이스 역할을 수행한다. 어플리케이션으로 렌즈의 유통기한과 세척 시기를 확인할 수 있고, 앞서 설명한 4가지 알림을 어플리케이션 내의 서비스를 통해 받을 수 있다.

먼저, Mqtt 사용에 대해 간략히 설명하고자 한다. Mqtt는 IoT 분야에서 사용할 수 있는 경량의 Publish/Subscribe 메시징 프로토콜이다. 라즈베리파이에서 초음파 센서와 렌즈 세척기에 부착된 버튼 인식 GPIO 핀에 각 Publish 프로세스를 동작시킨다. 그리고 라즈베리파이에서 스마트폰으로 데이터를 전달해 줄 Broker(Server)를 동작시킨다. 한편 스마트폰에서는 위의 Publish되는 요소들을 읽어 들이기 위하여 Subscribe 프로세스를 동작시킨다. 각 두 개의 Publish에서 렌즈 세척기에 부착된 조그마한 렌즈 케이스의 유무와 렌즈 세척기 버튼 동작 여부를 판별하여 Publish한다. Publish된 데이터들은 Broker로 전달되고, Broker는 해당 데이터에 대해 Subscribe하는 Client에게 데이터를 전달한다. Subscribe하는 클라이언트인 스마트폰은 데이터를 Broker로부터 수신하여 적절한 조치(타이머 시작 등)를 수행한다.

여기서 개인적인 역량이 발휘되었던 내용은 스마트폰에서 여러 가지 알림에 대한 기능들을 하나의 추상 클래스로 공통적인 부분을 정의하고, 각 세부 기능을 그 클래스로부터 상속된 클래스들로 정의하여 구현했다는 점이다. 그 공통적인 부분이 입력받은 시간 이후에 AlarmManager를 사용하여 알림을 받도록



하고, 그 알림을 입력받은 알림 내용과 함께 Notification을 사용하여 실제 스마트폰에서 푸시알림이 이루어지도록 하는 부분이다. 이러한 부분은 차후 추가될 수 있는 비슷한 유형의 기능에 대하여 손쉽게 작업이 가능할 것이라는 점에서 유용할 것이라 생각한다.

3) 한계점 및 개선 사항

라즈베리파이에 도메인 주소를 적용하지 않아서 스마트폰에서 LTE 망으로 통신이 불가능하다는 것이 있다. 또한, 렌즈 세척기에 부착된 조그마한 렌즈 케이스를 빼 가는 것으로 사용자가 렌즈 착용하는 것으로 인식하기 때문에, 실제로 사용자가 착용하는 시간에 대한 정확도가 떨어진다는 점이 있다. 마지막으로 사용자가 사용하는 총 렌즈 수에 대해 스마트폰과 스마트 렌즈 케이스 사이에 동기화를 시스템적으로가 아닌 사용자가 맞춰줘야 한다는 점이 있다. 즉, 사용자가 새로 산 렌즈를 스마트 렌즈 케이스에 추가하면 스마트폰에서도 사용자가 직접 추가해야 한다는 점이다. 렌즈를 제거할 경우도 마찬가지이다.

4) 발전 방향

만약 시장에 진출하게 된다면, 렌즈 판매사와 제휴하여 렌즈를 굳이 오프라인으로 구매하는 것이 아닌, 온라인으로 구매하도록 지원하는 것도 괜찮다고 생각한다.

5) 역할 분담

직책	팀원	역할
팀장	양우민	Subscribe 구현(어플리케이션)
팀원	팀원1	Publish 구현(라즈베리파이)
팀원	팀원2	부품 조달, 발표 및 시연
팀원	팀원3	부품 조달, 보고서 및 ppt 작성



6) 기타

- 개발 환경 : Android Studio, vi 편집기
- 개발 언어 : Java, Python
- 기타 : 삼성 갤럭시 노트2 사용

7) 구현 결과

- 소스 코드(모바일 부분)

https://github.com/YangWoomin/Portfolio/tree/master/Smart_lens_Case

- 시연 영상

https://www.youtube.com/watch?v=tow8_QtbIWU

https://www.youtube.com/watch?v=4CQZW3Me_dM

▶ 기타 프로그램

프로젝트들에 비해 비교적 규모가 작은 프로그램들에 대한 설명이다.

1. 멀티스레드 기반 채팅 프로그램

이 프로그램은 2016년 1학기 “네트워크 프로그래밍”이라는 과목에서 구현한 프로그램이다. 기존에 나와 있는 아주 기본적인 채팅 프로그램에 채팅방이라는 요소를 더한 것이 특징이다. 서버는 유닉스를 사용한 멀티스레드 기반으로 연결된 클라이언트 1개당 1개의 스레드로 관리하는 역할을 수행한다. 클라이언트도 메시지 송신부와 수신부를 스레드 단위로 역할을 수행하도록 구현하였다.

1) 개발 환경 및 언어

- 리눅스(우분투) 16.04 LTS, vi 편집기
- C언어

2) 구현 결과

- 소스 코드

https://github.com/YangWoomin/Portfolio/tree/master/Chatting_program_by_multi_thread

- 시연 영상

<https://www.youtube.com/watch?v=6Vu56Nu6Orl&feature=youtu.be>

2. 온라인 3D 테트리스

이 프로그램은 2015년 2학기 “컴퓨터 그래픽스 및 실습”이라는 과목에서 구현한 프로그램이다. OpenGL 라이브러리를 사용하였으며 온라인 3D 테트리스이다. 1:1 대전 형식이며, 게임으로서 완성된 프로그램은 아니고 테트리스로서 기본 조작과 화면 동기화 정도까지만 구현하였다. 통신 프로토콜은 TCP를 사용했으며, 윈도우에서 개발하였다.

1) 개발 환경 및 언어

- 윈도우 8, Visual Studio 2015



- C언어
- 2) 구현 결과
- 소스 코드

https://github.com/YangWoomin/Portfolio/tree/master/Online_3D_Tetris

- 시연 영상

<https://www.youtube.com/watch?v=2vrGsw59Nmo&feature=youtu.be>

3. Medicine Database Program

이 프로그램은 2015년 2학기 “데이터베이스 프로그래밍”이라는 과목에서 구현한 프로그램이다. MSSQL로 데이터베이스를 구축하였으며, Netbeans로 데이터베이스를 관리하는 응용프로그램을 구현하였다. 프로그램 목적은 사용자에게 약의 정보, 약국 정보, 약 제조사 정보, 약품 인증기관 정보를 제공하기 위한 프로그램이다. 위의 데이터베이스 구축을 위해서 요구 분석, 개념적 설계, 논리적 설계, 물리적 설계, 데이터베이스 구현의 단계를 수행하였다.

- 1) 개발 환경 및 언어
- 윈도우 8, MSSQL 2008, Netbeans
- Java

- 2) 구현 결과
- 소스 코드

https://github.com/YangWoomin/Portfolio/tree/master/Database_Programming

4. 기타 활동

2016년 1학기부터 교내 알고리즘 스터디그룹 활동으로 Baekjoon Online Judge 사이트에서 문제풀이를 진행하고 있다.

- 사이트

<https://www.acmicpc.net/user/cicyowoo>