



어린이를 위한 교육용 알고리즘 장난감 블록

Educational algorithm toy blocks for children

종합설계 설계서_

2014150018 서준원 (지도교수 서대영)
2012151002 권혁준 (지도교수 한익주)
2012151029 오우석 (지도교수 한익주)

Know Answer.

차례

1. 종합설계 개요
2. 관련 연구 및 사례
3. 시스템 수행 시나리오
4. 시스템 구성도
5. 시스템 모듈 상세 설계
6. 개발 환경 및 개발 방법
7. 데모 환경 설계
8. 업무 분담
9. 종합설계 수행일정
10. 필요기술 및 참고문헌
11. Q & A

지난 발표에서의 지적 사항

- ① 어느 정도 수준의 알고리즘 구현이 가능한지 명시가 필요하다.
- ② 콘텐츠는 어떤 콘텐츠인지 명시가 필요하다.
- ③ 콘텐츠 다양성을 설계해야 할 필요가 있다.

지적 사항에 대한 답변

- ① 5~7세의 아이들이 너무 어렵지 않게 느껴 흥미를 잃지 않도록 콘텐츠가 구성되어야 하므로 상하좌우 이동이나 회전 같은 간단한 수행의 알고리즘부터 비교적 어려운 반복 등을 이용하여 문제를 해결하도록 한다.
- ② 블록마다 고유의 기능이 담겨져 있어서 그 기능에 따라 콘텐츠의 스테이지에 맞게 블록을 올바른 순서로 두어 클리어하는 방식
- ③ 콘텐츠 다양성의 한계를 해소하고자 레벨 디자인을 통한 학습 곡선 완화

종합설계 개요

개발 배경

개발 목표

개발 효과

어째서 SW교육이 필요한가?

19세기 이전
농경사회



19 - 20세기

1, 2, 3차
산업혁명 시대



세상의 산업구조가
어떻게 변화되어가는가?

지금 해외에선
무슨일이 일어나고 있는가?



중국, 일본
연간 각 55,70시간 이상
SW 의무교육



영국

"2014년은 코드의 해"
초중등과정 SW 정규교육화
5세 이후 주당 1시간 이상 교육



미국

"하루 한시간씩 코딩하라"
코드교육 캠페인 품
초중등과정 SW 정규교육화

한국또한...

2018년 초중등 SW교육 의무화!

2016 초 5 - 6 대상 <SW 기초 소양교육> 개편

교육용 도구를 이용한 놀이중심의
알고리즘, 프로그래밍 체험 교육!

교육부

미래사회 경쟁력 강화 인식

"여러분의 아이들에게
SW교육이 필요할까요?"

학부모 61%
잘은 모르지만,
필요하다!

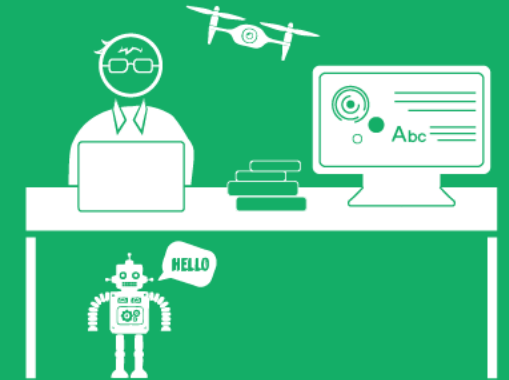
출처 : KBS 뉴스 기사 「라면콜이기도 알고리즘, 아이들 영어 말고 코드 교육」
NFC 통신 기사 「초등학부모 61%, SW교육 의무화...잘 모르지만 필요하다」
교육부 「2015년 초중고 교육과정 개정 보도자료」

21세기

4차 산업혁명_ 산업 융복합 시대

Digital Natives

기술자, 개발자, 디자이너 간의 장벽이 사라진다.



디터 제체 _ 메르세데스 벤츠 다임러 CEO

"이제 자동차는 가솔린이 아닌 소프트웨어로 달립니다."

버락 오바마 _ 미 전 대통령

"코딩은 당신의 미래일 뿐 아니라 조국이 미래이기도 합니다.
앱을 다운받아 놀지말고, 앱을 만들 수 있어야합니다."

마크 저커버그 _ 페이스북 CEO

"코드를 만들 줄 안다는 것은,
무언가를 만들어낼 수 있는 힘을 갖게 된다는 뜻이고,
아무도 당신을 막을 수 없다는 뜻입니다."

어린이가 알고리즘에 대해
쉽게 학습하도록 하려면
무엇이 필요할까!

종합설계 개요

개발 배경

개발 목표

개발 효과

—
어린이를 위한 소프트웨어 교육 장난감 !

소프트웨어 교육의 첫 수용자
차세대 디지털 세대의 주체

어린이

: 어린이가 소프트웨어 개념, 즉
알고리즘을 쉽게 익힐 수 있는 장난감을 만들자 !

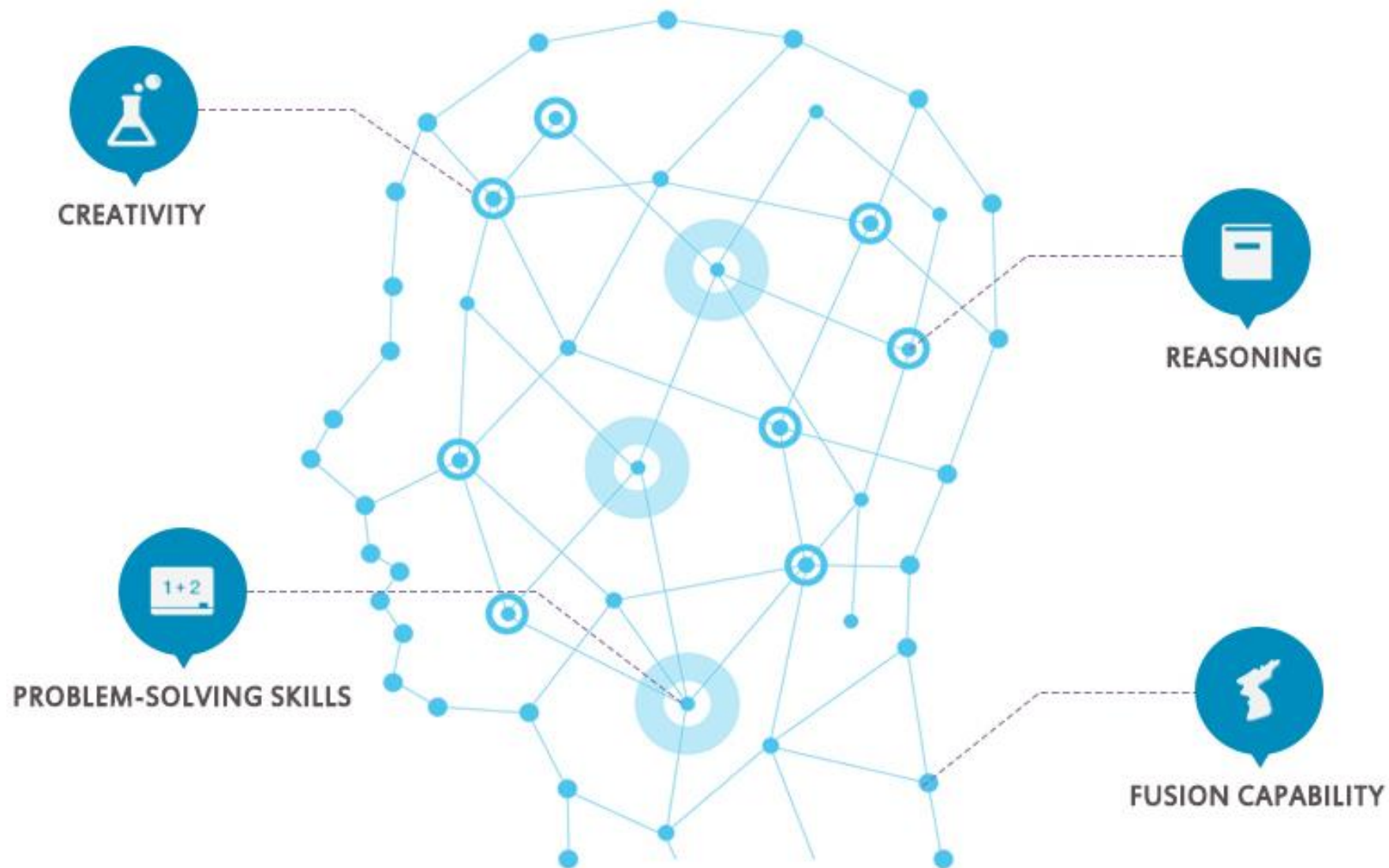


종합설계 개요

개발 배경

개발 목표

개발 효과



관련연구 및 사례

기존 사례

로봇형 제품

Tangible Programming

기존의
교육방식은
어떠한가?

FIRST

블록형 코딩 프로그램
스크래치 등장

SCRATCH



메사추세츠 공대 media labs 개발
2007년 개발, 2009년 국내 도입

현재까지 어린이 소프트웨어 교구로
가장 많이 쓰이고 있는 툴

THEN

블록형 코딩 프로그램
다양화

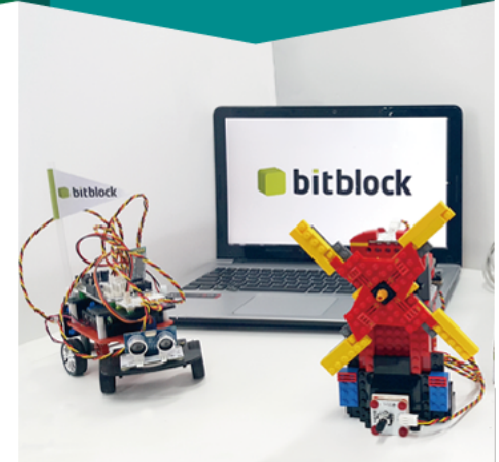


더 넓은 사용연령층을 위하여
더 쉬운 사용성을 가진
스크래치 주니어 출시

스크래치의 오픈소스를 필두로
국산 블록형 코딩 툴
엔트리 출시

AND NOW

코딩 교육용
로봇 장난감 등장



오픈소스 <아두이노>와
<블록형 코드>를 결합한
쉬운 로봇 제어 툴 등장

사운드, 이미지 중심의
디스플레이 교육을 넘어
아이들의 흥미를 돋구는 방향으로

관련연구 및 사례

유형 1_ 로봇형 제품

주로 한국 중소기업에서 최근 뜨겁게 출시되고 있는 로봇 형태의 코딩 교육 장난감.

자칫 딱딱할수 있는 알고리즘의 수학적 개념을 로봇 제어 개념과 접목시켜 친근하게 다가갈 수 있도록 한 제품들이 이목을 끌고있다

기존 사례

로봇형 제품

Tangible Programming



캐릭터를 입히는 로봇 DIO



하지만

- 1) 기존에 존재하던 아두이노 + 스크래치의 교구를 외형 디자인만 변화시켜 접근하고자 한 제품들이다!
- 2) 때문에 교육의 난이도, 접근성, 콘텐츠 등 UD 측면에서의 디자인이 고려되지 않고 있다!
- 3) 액추에이터 (구동부품)이 많아질수록 제품 단가가 상승한다!

관련연구 및 사례

기존 사례

유형 2_ Tangible Programming

최근 외국에서는 유아동층을 타겟팅하여
실제 블록을 이용한 **Tangible Programming** 개념이 확산되고 있다.

문제 해결의 순서 개념을 시각적으로 보여주는데 집중한다.

로봇형 제품

Tangible Programming

Fisher-Price Code-A-Pillar

Code와 Catapillar (애벌레) 의 합성어로,
애벌레의 몸체를 구성하는 각 블록에
방향 명령을 입력해놓으면
그 순서에 맞추어 움직이는 장난감



Google Project : Blocks

구글에서 개발중인 제품으로,
블록간의 다양한 조합을 통해
로봇을 움직이거나
회로 설계를 할 수 있도록 한다.

PRIMO Cubetto

상하좌우를 나타내는 블록을
판 위에 순서대로 배치하면
그 순서에 따라
작은 로봇이 움직인다

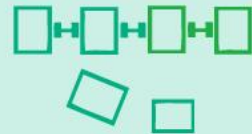
말을 손으로 움직이는 대신
알고리즘 판을 이용하여 원격조종하는,
보드게임에 로봇조종을 더한 장난감이다



OSMO Game : Coding

디스플레이와 피즐간의
상호작용을 이용하는
여러 게임 중 하나로
(Coding)을 선보였다.

컨트롤러가 아닌
블럭의 조합을 통해
캐릭터를 움직여 임무를 완수한다.



실제 블록의 조합 순서에 따라

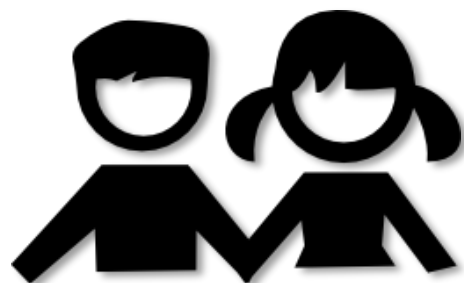


로봇이나 디스플레이 속 캐릭터를 움직이는 방식 !

하지만

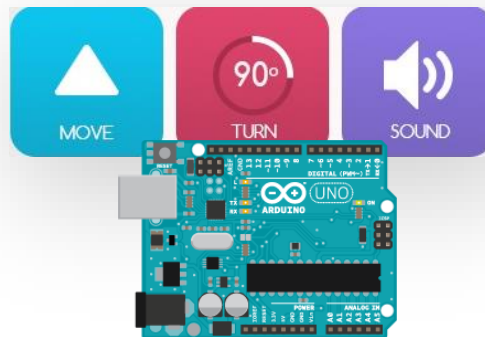
- 1) 프로그램을 짜고 제어하는데에는 훨씬 많은 수의 명령어가 필요한데, **블럭의 수는 제한되어있다 !**
- 2) 다수의 블럭을 꺼내고, 정리하는데에 **추가적인 어려움이 따른다 !**
- 3) 제공될 수 있는 콘텐츠에 **한계가 있다 !**

시스템 시나리오



5 ~ 7세의 아이

①
블록 조작



②
블록 배치
정보 전달

Web Browser

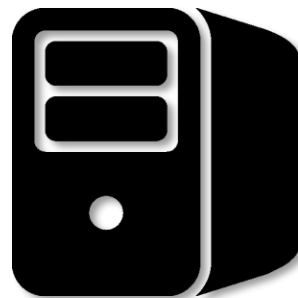


⑤
블록 배치
정보에 기반하여
임무 수행

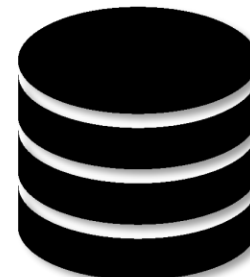
TASK

1. 블록들을 준비한다.
2. 메인블록의 전원을 켜 디바이스와 연동시킨다.
3. 앱을 실행시킨다.
4. 콘텐츠상에서 필요한 명령어에 따라 블록을 조합하여 명령어를 만들어낸다.
5. 메인 블록의 실행 버튼을 누를때마다 해당 정보가 디바이스로 전송된다.

Server

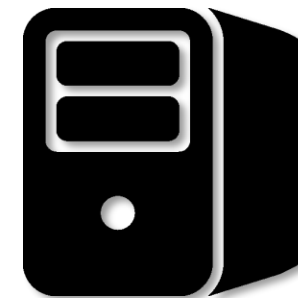


DB



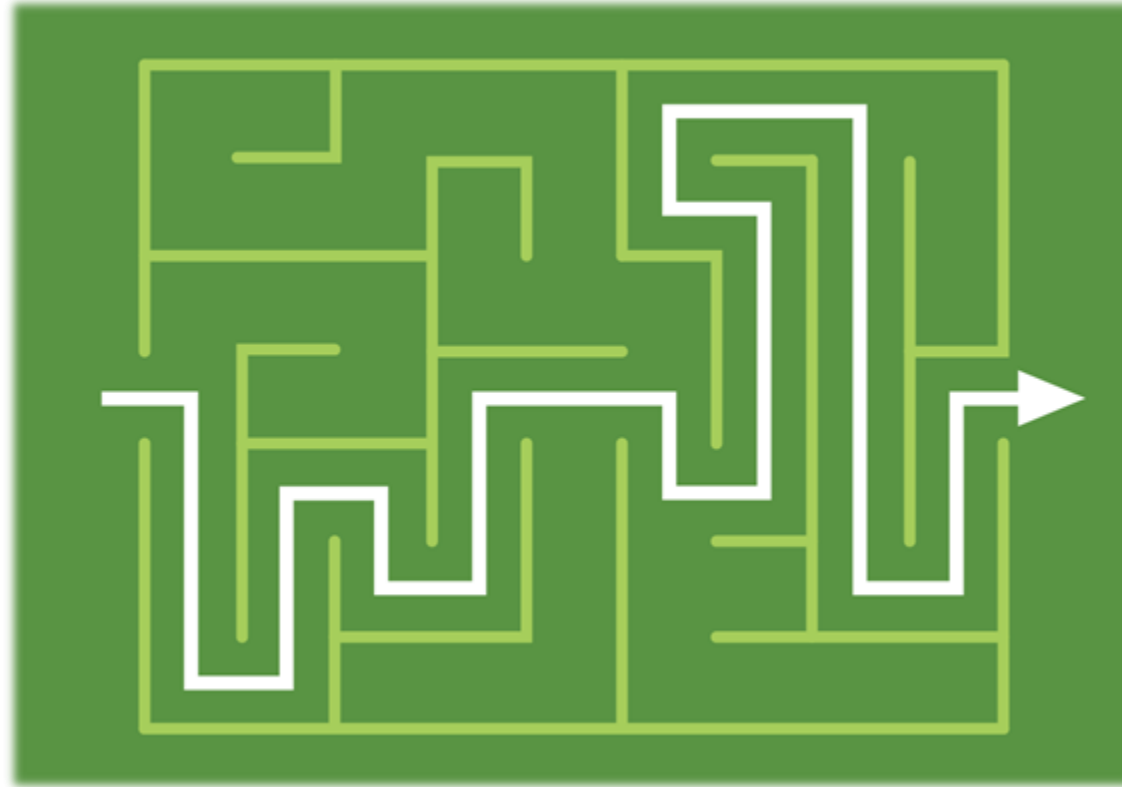
④
실시간
DB 변화
감지

Client



③
DB 저장
및 불러오기

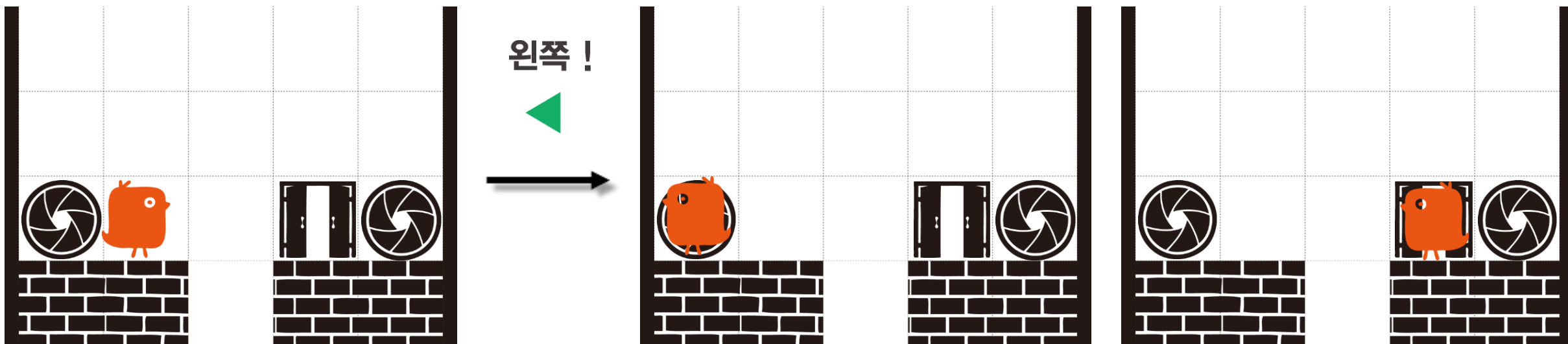
시스템 시나리오



기본 개념은 미로!

시스템 시나리오

블록마다 고유 기능이 담겨있어 배치된 순서의 블록에 따라서 해당 스테이지를 클리어하는 방식



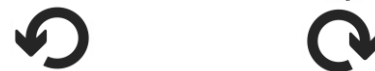
어린이 수준에 맞춘
알고리즘 구현!!

컨텐츠 다양성의 한계를 해소하고자 레벨 디자인을 통한 학습곡선 완화

오른쪽 ! 점프 ! 왼쪽 !



왼쪽 회전 ! 오른쪽 회전 !



컨텐츠

전진하기

후진하기

좌회전

우회전

반복시작

반복시작(2)

반복끝

아이템줍기

01

튜토리얼

게임 진행방식 소개

02

미로찾기

사용블럭 : 전후+회전

03

미로찾기

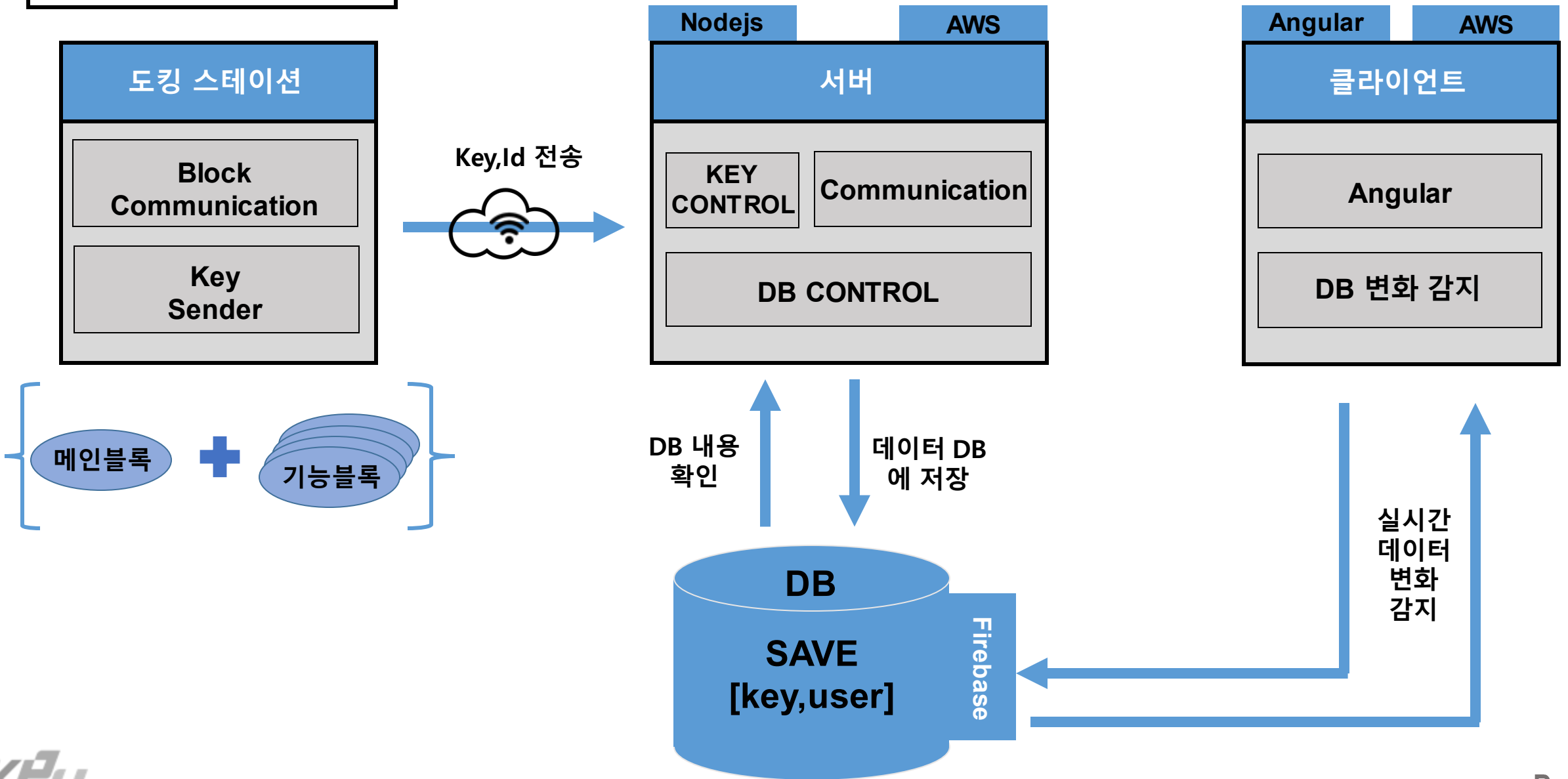
사용블럭 : 전후+회전+반복

04

길찾으며 행동하기_미로찾기(HARD)

사용블럭 : 전후+회전+반복+행동 (줍기)

시스템 구성도



기능

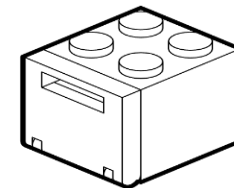
기능 블록들 (슬레이브)의 계산 또는 고정된 데이터를 I2C 통신을 이용하여 **메인 블록**(마스터)에 데이터 전송한다.
 각 기능 블록으로부터 키값을 도킹 스테이션의 지정된 순서대로 받는다.(마스터)
 순서대로 받은 값을 구분자 K 로 두어 하나의 문자열로 저장(마스터)

다루는 정보

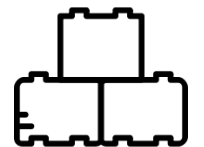
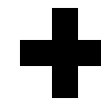
각 기능 블록의 시리얼 고유번호
 (예 : 100,200,300,.....1200,1300)
 각 기능 블록의 고유이름
 (예 : FWRD, BWRD, TUNR, TUNL, LOOP, GETT, LPED)

메인블록에 들어오는 정보

도킹 스테이션에 꽂혀있는 각 기능블록의 고유이름



도킹스테이션



기능블록

수행 흐름

메인블록에서 각 기능블록으로 부터 반복문을 이용하여 데이터 전송받기
데이터는 각 기능 블록들의 고정주소와 고정이름을 순서있게 문자열에 저장

수행시점

메인 블록에 수행 이벤트 발생 시

블록 간 통신방법

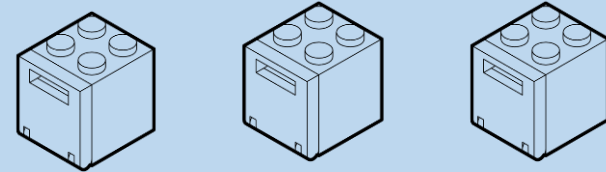
Inter Integrated Circuit (I2C)
1:N 방식으로 통신

이용 알고리즘

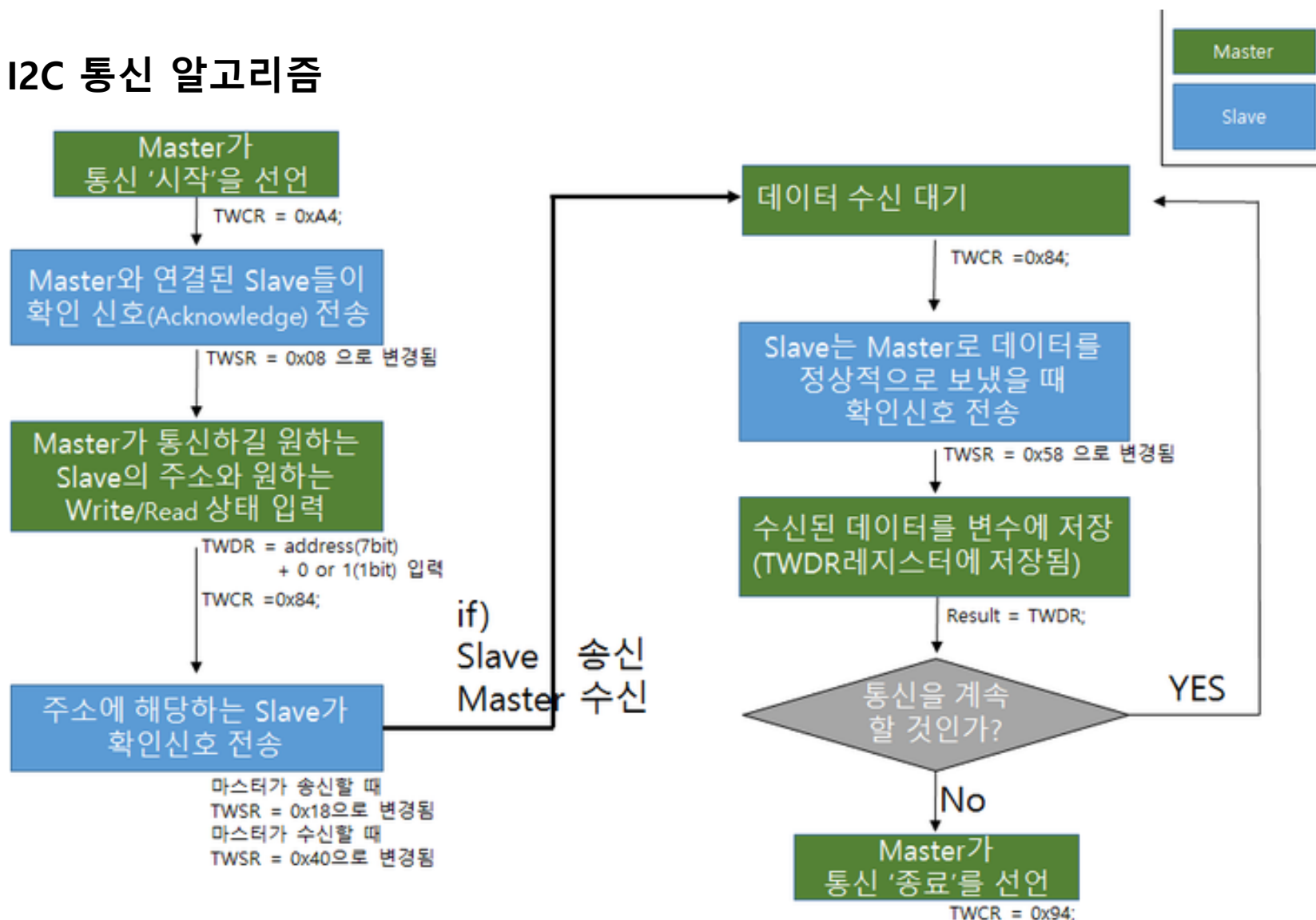
I2C 수신 알고리즘(블록간 통신)

실행 결과 값

String CMD;
-> 각 기능 블록 데이터 집합 배열



I2C 통신 알고리즘



기능

ESP 8266-1 아두이노 와이파이 모듈을 이용하여 와이파이 신호를 잡고, 서버에 결과데이터를 전송한다.

전송 시 사용되는 통신규약

hypertext transfer protocol
HTTP

전송 시 사용되는 전송방식

Get method

다루는 정보

String CMD; → Block Communication 을 통해 나온 블록값 결합 결과값
CMD 는 K 구분자로 구분되어있는 형태

전송 URL

<http://localhost/InputKeyPage?keyvalue=데이터값>

전송 명령어

아두이노 명령어 AT+CIPSTART 이용 [Serial.println(CMD)]

기능

도킹스테이션으로부터 날라온 데이터를 가공하며, 가공 시 블록의 순서 또는 논리의 오류가 있을 경우 처리한다.
(데이터 가공 : K 구분에 따른 분리, 반복문 블록 처리)

선수 작업

도킹스테이션으로부터 날라온 데이터를 `SeparateKeyValues()` 를 이용하여 K 로 구분되어있는 값을 배열에 차례대로 저장한다.

후 작업

저장된 배열 값을 이용하여 블록의 논리성이 있는지 판단한다. 논리성에 판단여부에 따라 오류값 입력

다루는 데이터

`String rcvData`: 아직 구분자로 구분이 안된 데이터
예) `data1Kdata2Kdata3Kdata4.....` 인 문자열 형식(K구분자로 구분)
`String separatedKeys[]`: 구분자로 구분된 데이터

수행 시점

도킹스테이션으로부터 데이터를 바로 받아오자마자 실행
서버 접속 이벤트가 발생할 시 실행

함수명	SeparateKeyValues()
형식	function SeparateKeyValues(string rcvData, 'K')
리턴값	String[] separatedKeys
설명	들어온 데이터값을 K으로 구분하여 separatedKeys 에 저장한다. 함수 내 사용되는 함수로 splite 가 있다.

함수명	CheckKeyValues()
형식	function CheckKeyValues(string[] separatedKeys)
리턴값	ture false
설명	분리된 키값들의 반복문 블록에 따른 논리적 순서가 맞는지 체크한다. 체크 해서 정상 일 경우 DB 컨트롤러에서 정상처리, 오류일시 DB 컨트롤러에서 오류처리

기능

도킹 스테이션으로 부터 HTTP GET 방식으로 데이터 수신하여 KeyController에 전달
데이터베이스 와 서버간 통신 설정

수행 시점

아두이노에서 데이터를 서버로 송신한 다음 수행

흐름

1. 도킹 스테이션으로 부터 webpage를 호출 시에 대한 서버 연결
2. 데이터베이스로 데이터를 전송하기 위해 데이터베이스 연결

다루는 정보

도킹 스테이션으로 부터 받은 Key 값



기능

데이터베이스에 접속한다.
도킹스테이션으로부터 요청 받아 가공된 데이터를 파이어베이스 데이터베이스에 업데이트한다.

수행 시점

도킹스테이션으로부터 전달된 데이터가 가공 된 후 수행

다루는 정보

separatedKeys 배열의 데이터를 데이터베이스에 올리기 위해서 다룬다.

DB 초기화

firebase.initializeApp() 를 이용하여 DB URL 초기화
firebase.database() 를 이용하여 DB 객체 초기화

DB 수정

DB객체값.update() 로 DB값 업데이트
DB개게값.set() 로 DB값 초기화

함수명**checkId()****형식****function checkId(var id)****리턴값****string _id;****설명****아두이노의 아이디 값을 체크한다.****함수명****getKey()****형식****function getKey()****리턴값****string _key;****설명****키값을 데이터베이스에서 얻어온다.**

함수명**getStation()****형식****function getStation()****리턴값****String result****설명****현재 시나리오를 알려준다.****함수명****getSolution()****형식****function getSolution(station,step)****리턴값****String result****설명****현재 시나리오의 단계에서의 문제답안을 얻어온다.**

함수명**updateStep()****형식****function updateStep(station, id)****리턴값****void****설명****해당하는 레벨단계를 수정한다.****함수명****updateKey()****형식****function updateKey(key)****리턴값****void****설명****데이터베이스의 키값만 수정해준다.**

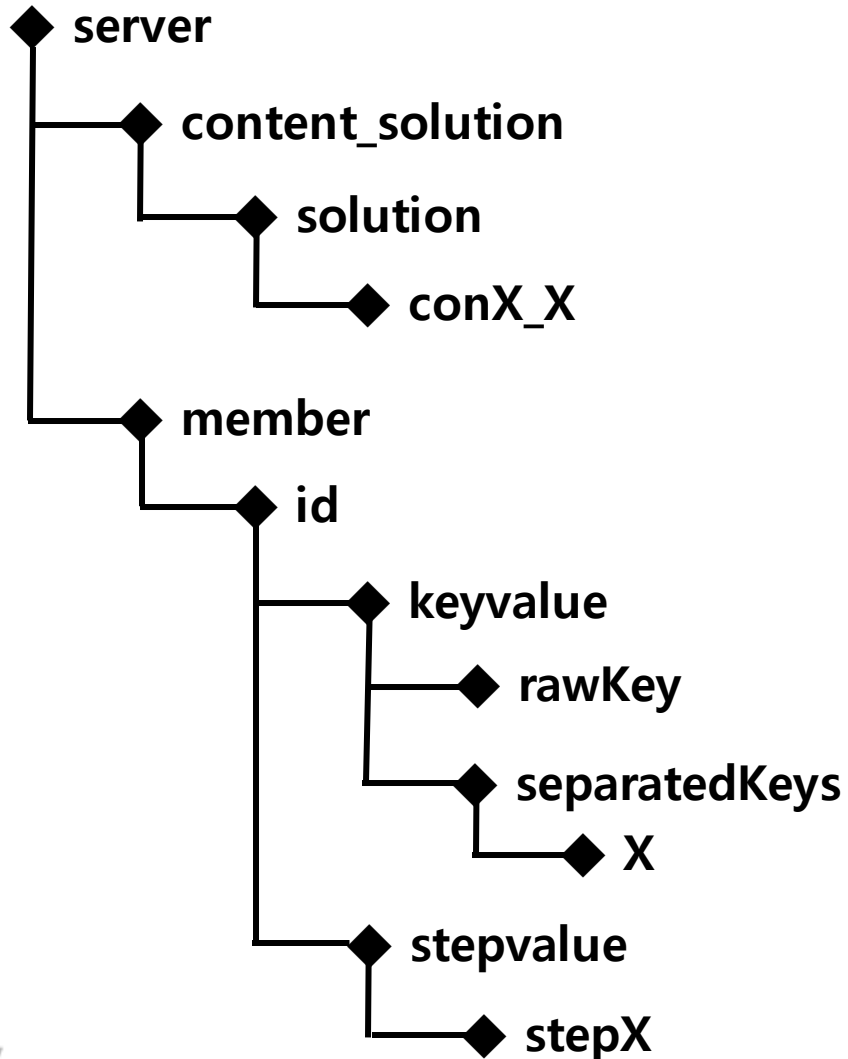
함수명**setKey()****형식****function setKey(key)****리턴값****void****설명****key값을 초기화한다.****함수명****getStep()****형식****function getStep(id,station)****리턴값****int result****설명****해당하는 레벨 단계를 리턴한다.**

함수명**errorKey()****형식****function errorKey()****리턴값****void****설명****판별되어 온 값에서 오류라 판단하면 정상단계 비실행, 대신 위 함수 실행**

함수명	list
형식	<code>constructor(af: AngularFire) { this.items = af.database.list('url 주소');</code>
설명	생성자에서 데이터베이스 목록을 list함수에 url주소를 인자로 주면 해당 데이터베이스 정보를 가지고 있는 FirebaseListObservable 객체를 반환한다. 이를 통하여 데이터를 받을 수 있다.

함수명	push
형식	<code>this.items.push("집어 넣을 데이터");</code>
설명	실시간 데이터베이스의 push id들을 이용하여, 리스트에 새로운 목록을 만듭니다

함수명**update****형식****this.items.update("키 값", "데이터");****설명****해당 키의 데이터를 전달인자로 보내는 데이터로 갱신합니다.****함수명****remove****형식****This.items.remove() 또는 this.items.remove("키 값")****설명****전달인자로 아무것도 없으면 모든 데이터를 삭제하지만,
전달인자로 키 값을 보내면 해당 키에 해당하는 데이터만 삭제 합니다.**



DB

- firebase DB 서비스이용
- 주소
<https://m2mproject-d7ae6.firebaseio.com/>

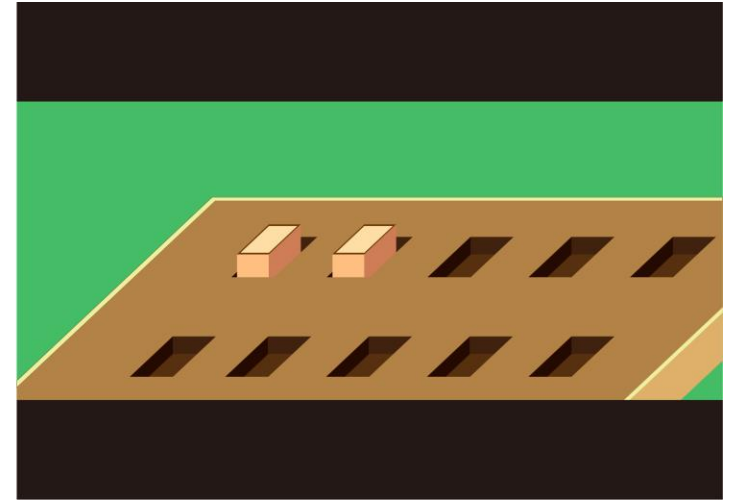
DB form

크게 2개로 구분
문제정답 결과 / 회원정보

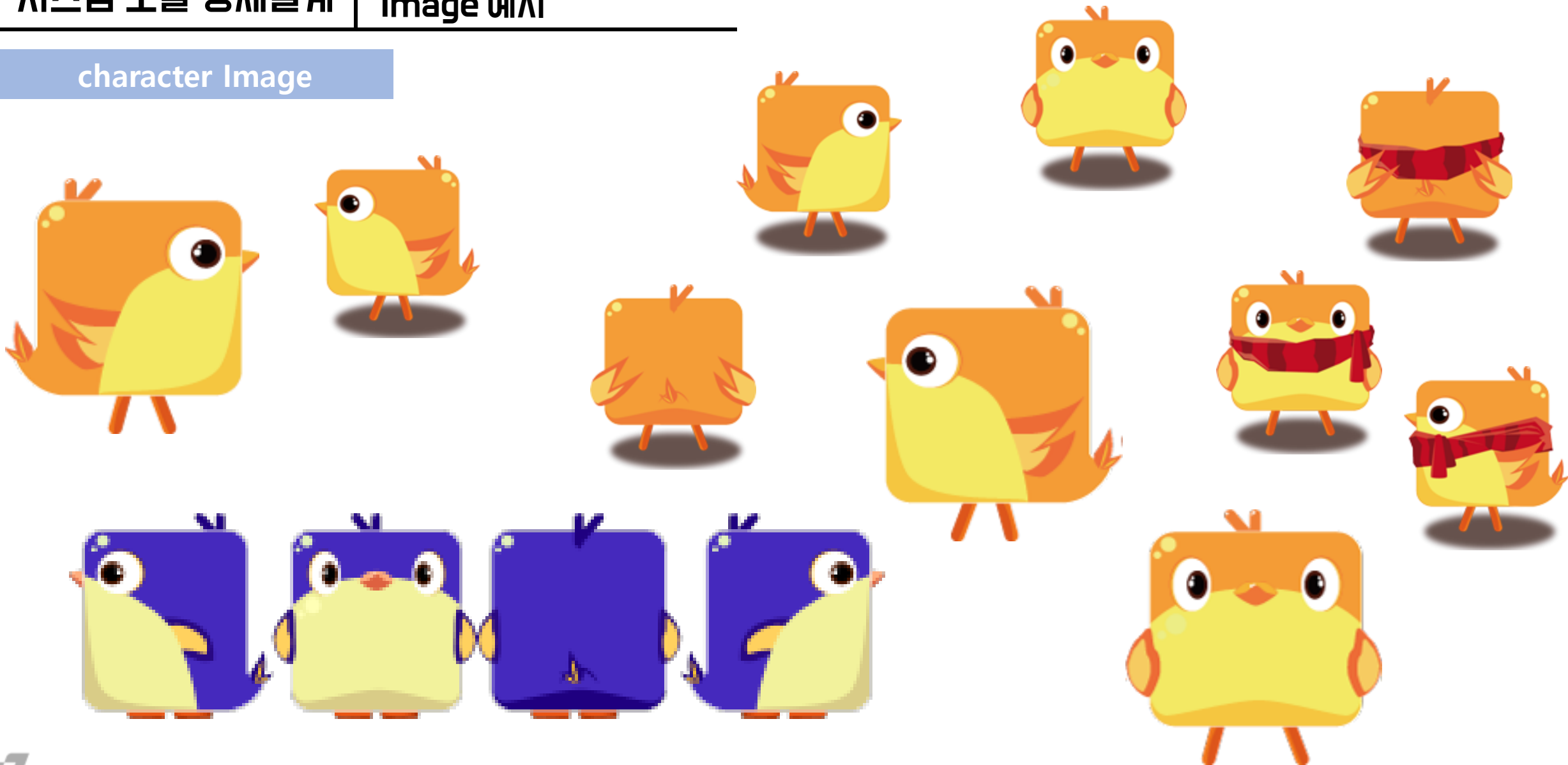
각 회원은

- 각각 키 값에 대한 정보를 가지고있고, 키 값은 원본 키 값과 순서에 따른 분리된 키 값으로 저장된다.
- 자신의 문제 단계를 저장하고있다.

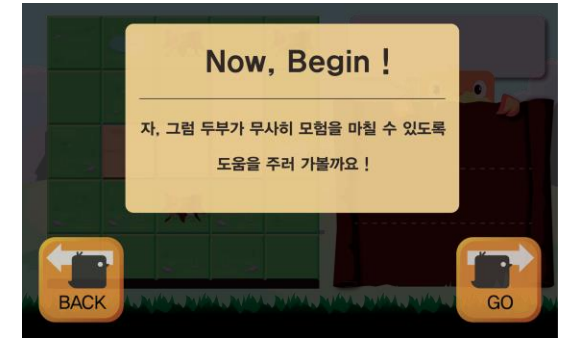
Image 예시



character Image



story



Server

- 기본포트 80 이용
- 기본URL http://localhost/

- **/**
처음 인트로 화면 표시
- **/InputKeyinitpage**
데이터베이스 초기화하는 페이지
- **/InputKeypage**
아두이노로부터 데이터값 받고 DB에 저장하는 역할

Client

- 기본포트 4200 이용
- 기본URL http://localhost:4200/

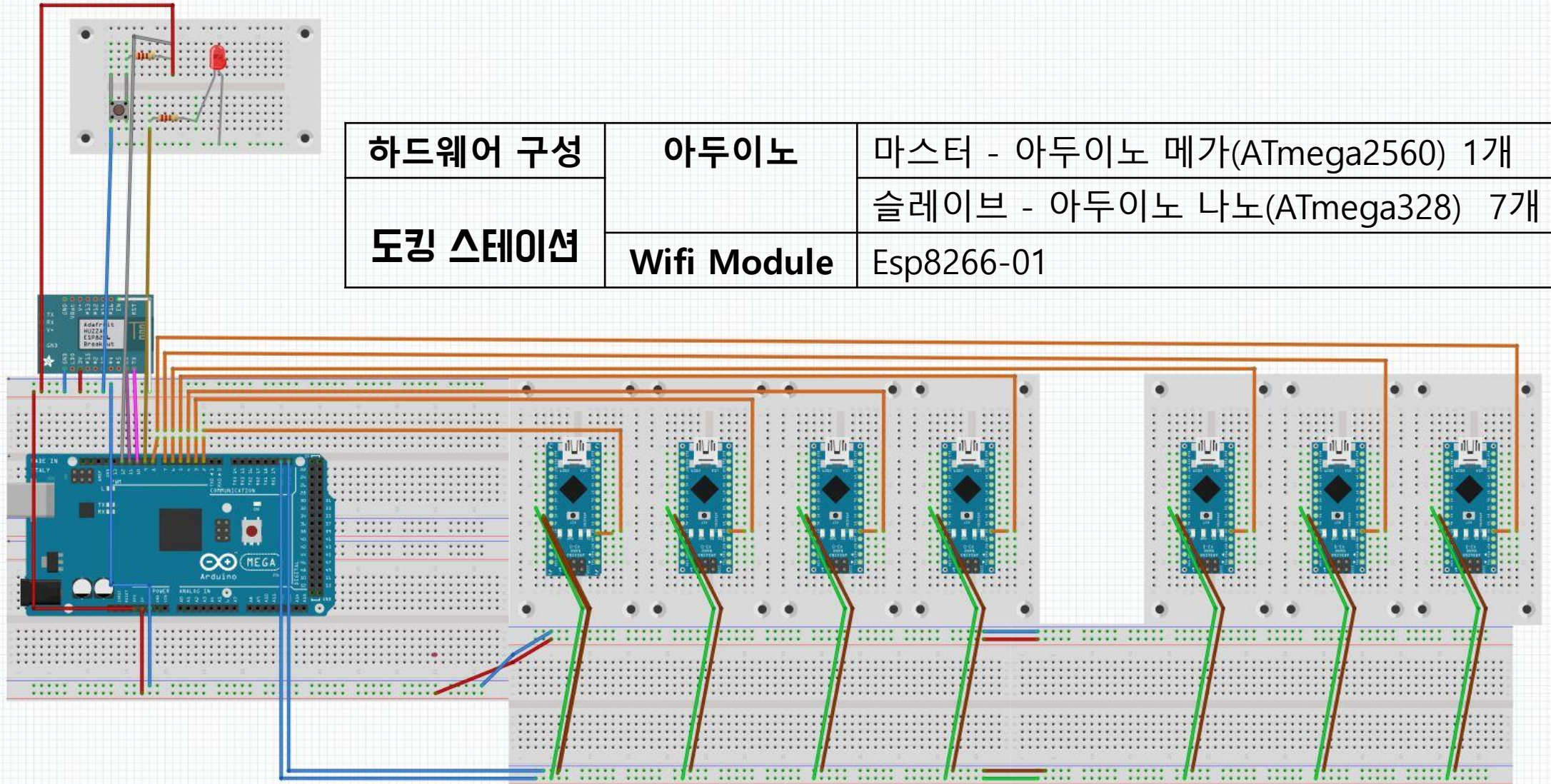
- **/**
처음 인트로 화면 표시
- **/game1**
게임 첫번째 페이지
- **/game2**
게임 두번째 페이지
- **/game3**
게임 세번째 페이지
- **/game4**
게임 네번째 페이지

싱글페이지로 운영

시스템 모듈 상세설계

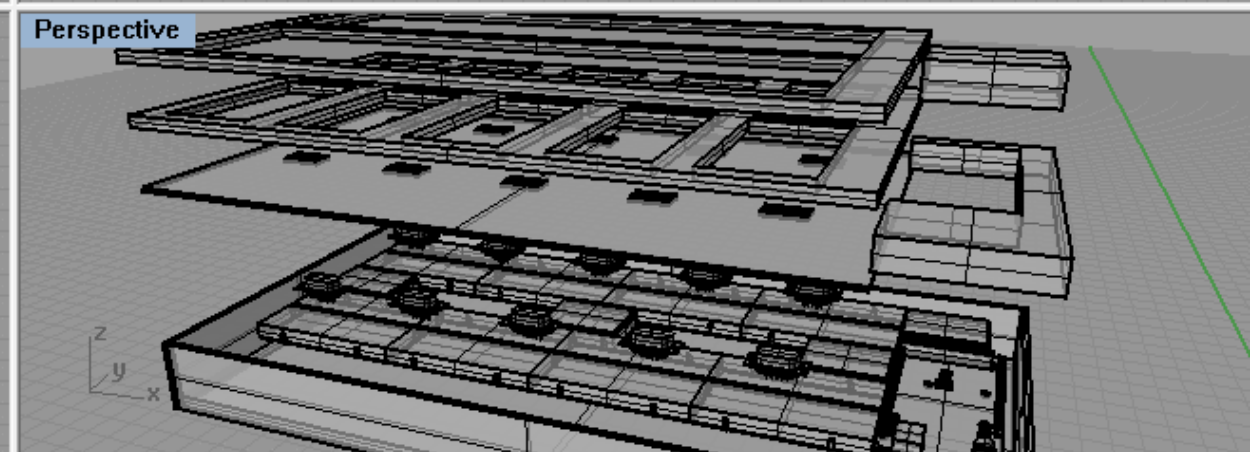
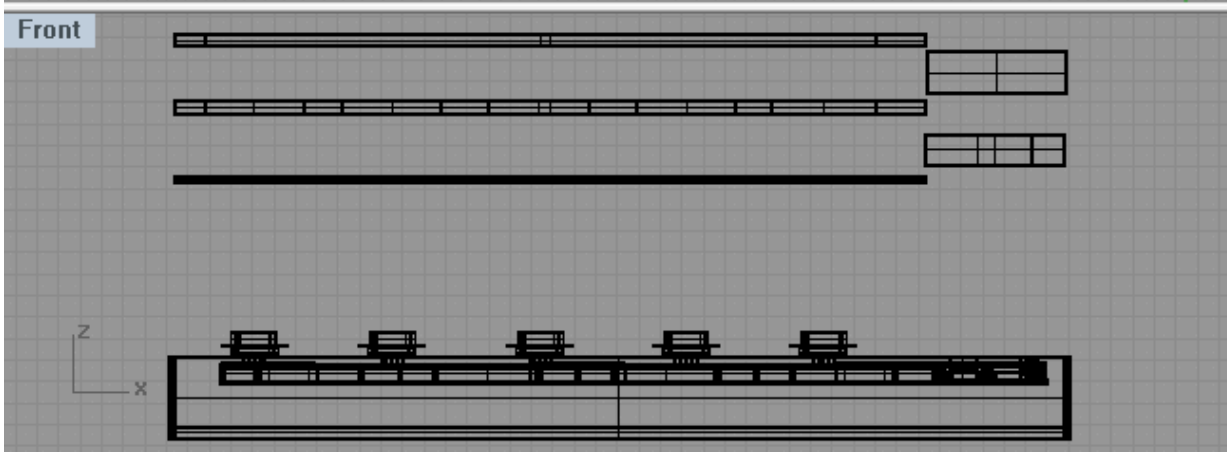
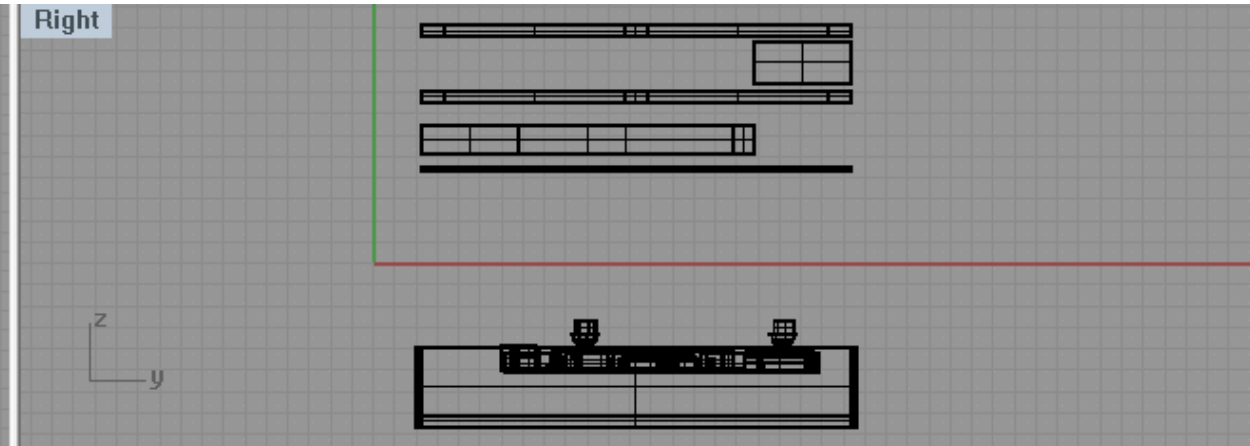
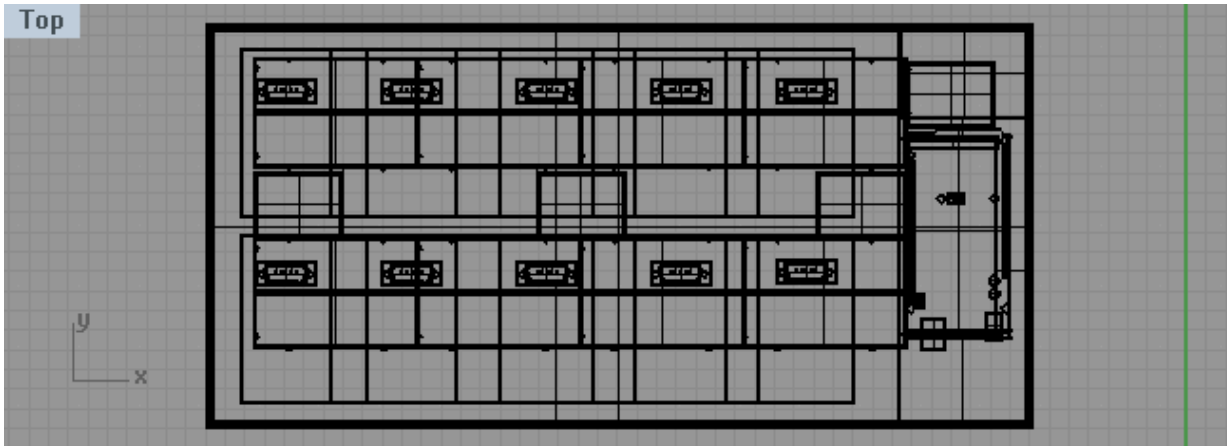
예상 도킹스테이션 회로도(HW)

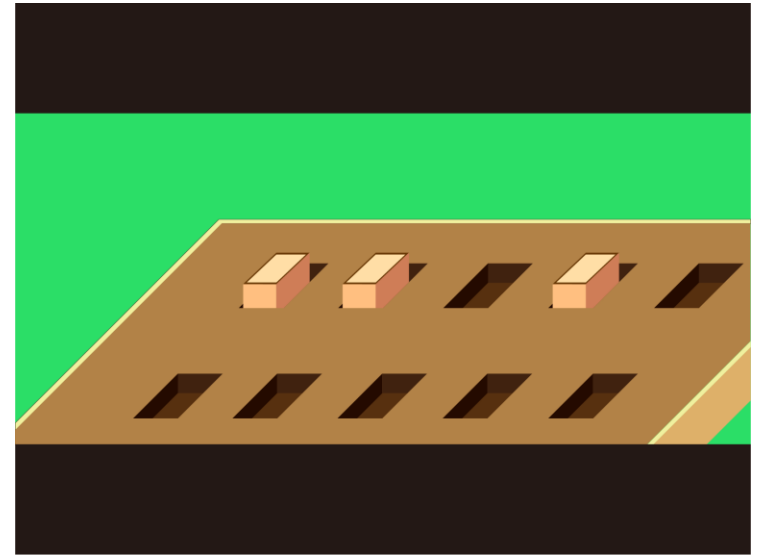
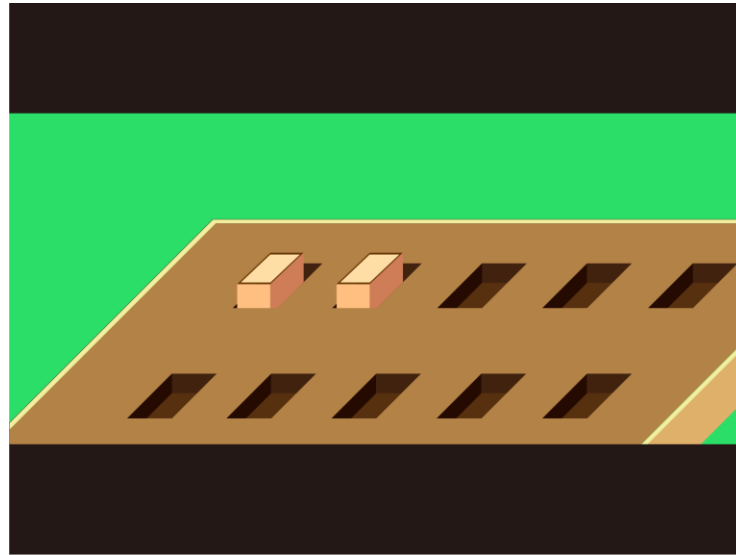
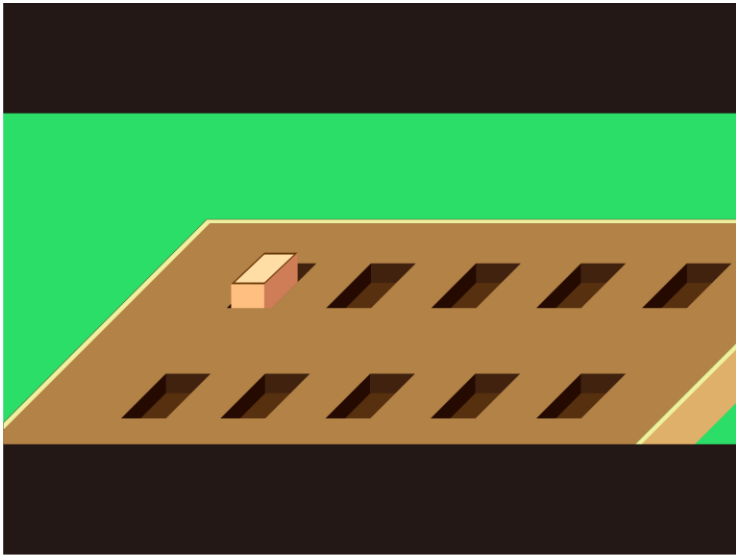
하드웨어 구성	아두이노	마스터 - 아두이노 메가(ATmega2560) 1개
도킹 스테이션	Wifi Module	슬레이브 - 아두이노 나노(ATmega328) 7개
		Esp8266-01



시스템 모듈 상세설계

예상 도킹스테이션 도면(HW)







Arduino Mega328

마이크로 컨트롤러	ATmega328
작동 전압	5V
정격 전압	7~12V
허용 전압	6~20V
디지털 입출력핀	14개 (이중 6개는 PWM을 지원함)
아날로그 입력핀	6개
DC 입출력핀의 출력 전류	20 mA
DC 3.3v일 때 출력 전류	50 mA
플래시 메모리	32 KB (이중 0.5 KB는 부트로더가 사용)
SRAM	2 KB
EEPROM	1 KB
클럭 속도	16 MHz



Arduino Mega2560

마이크로 컨트롤러	ATmega2560
작동 전압	5V
정격 전압	7~12V
허용 전압	6~20V
디지털 입출력핀	54개 (이중 15개는 PWM을 지원함)
아날로그 입력핀	16개
DC 입출력핀의 출력 전류	40 mA
DC 3.3v일 때 출력 전류	50 mA
플래시 메모리	256 KB (이중 8 KB는 부트로더가 사용)
SRAM	8 KB
EEPROM	4 KB
클럭 속도	16 MHz



ESP8266-01

S802.11 b/g/n

Wi-Fi Direct(P2P), soft-Ap

TCP/IP 프로토콜 내장

SDIO 2.0, SPI, UART

32핀 QFN 패키지

RF 스위치, balun, 24dBm PA, DCXO, PMU 내장

RISC 프로세서, 메모리, 외부 메모리 인터페이스 내장

MAC/baseband 프로세서 내장

고기능성 오디오 어플리케이션과 인터페이스를 위한 I2S

모든 내부 전원 공급을 위한 low-dropout linear regulator 내장

적절한 spurious-free clock generation 아키텍처

WEP, TKIP, AES, WAPI engine 내장

개발 환경

졸업작품 GitHub 주소

- https://github.com/kwonmory/KPU_KnowAnswer.git

팀원별 GitHub ID

팀장 : 권혁준

ID : kwonmory

팀원 : 서준원

ID : junwon1994

팀원 : 오우석

ID : woosuck-oh



Home

kwonmory edited this page just now · 1 revision

한국산업기술대학교
KOREA POLYTECHNIC UNIVERSITY

KOREA POLYTECHNIC UNIVERSITY

COMPUTER ENGINEERING

Graduation Work GITHUB

Member

- Seo junwon (12)
- Kwon Hyukjun (12)

Start date 2016-12-01

Report

- Date of Proposals & Plan - 16.12.22
- The Proposal release date - 16.12.26

개발 현황

▶ 진행된 부분

- 1) 하드웨어] 도킹스테이션 Block Communication
- 2) 하드웨어] 블록하드웨어에서 서버로 데이터를 순서있게 전송하는 기능
- 3) 하드웨어] 각 블록의 기능 추가하는 부분
- 4) 서버] 블록의 순서 논리력 판단 체크하는 기능(반복문)
- 5) 서버] 파이어베이스를 이용한 실시간 데이터베이스 연동
- 6) 서버] 블록으로 부터 들어온 데이터 올바르게 순차로 실시간 데이터베이스에 저장하는 기능
- 7) 클라이언트] 웹페이지 구성(html, css, javascript)
- 8) 클라이언트] 실시간 데이터베이스 연동에 따른 컨텐츠진행 기능
- 9) 클라이언트] 데이터베이스 값의 변경 데이터에 따른 애니메이션 작동
- 10) 디자인] 캐릭터, 컨텐츠, 하드웨어 목업, 블록,

▶ 진행할 부분

- 1) 하드웨어] 기존 문제점에 따라 블록 배치방식을 새로운 방식으로 진행예정
- 2) 클라이언트] 좀 더 디테일 한 컨텐츠 부분추가
- 3) 클라이언트] 좀 더 많은 블록 기능(ex : 아이템 줍기, 몬스터 죽이기 등)
- 4) 클라이언트] 반응형 디자인

개발 방법



Hardware

아두이노를 이용하여 서버에 필요한 데이터값 전송하는 하드웨어 구현
ESP8266 모듈을 통해 데이터값을 통신
ARDUINO IDE 1.8.1 로 펌웨어 실시



Server

Amazon Web Service의 EC2를 이용
Node.js Express 서버 구축



Database

FireBase 의 NoSQL 사용



Web Application (Client)

Angular, Node.js, Express, Typescript 으로 개발
웹 기반으로 어플리케이션을 개발함으로써 모든 플랫폼에서 이용 가능하게 함



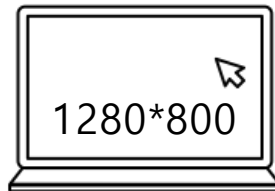
도킹 스테이션 1대



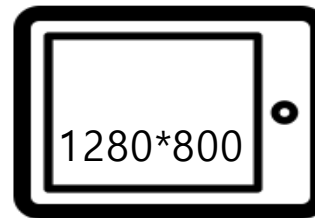
순서 블록 00개



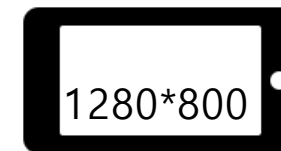
와이파이



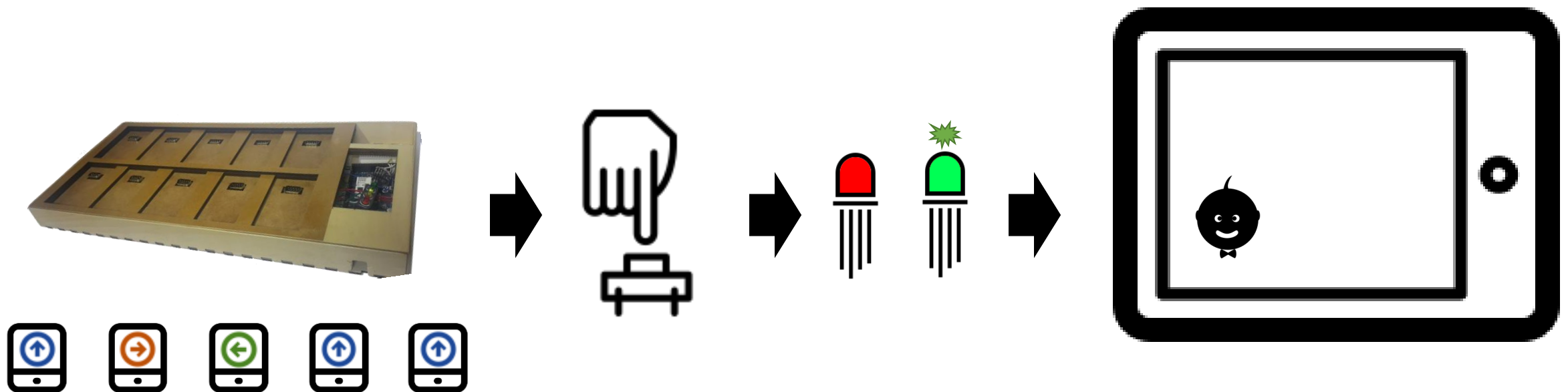
OR



OR



디스플레이 1대(노트북,태블릿,핸드폰 중 택1)





업무 분담

	자료수집	설계	구현	테스트
서준원	ESP8266 Typescript Node.js 웹 구조(UX,UI 포함) 어린이대상 콘텐츠	Web 구조설계 웹서버 구축 콘텐츠 로직	콘텐츠 구현 목업 디자인 출력	데이터 정확성 테스트 최종 테스트
권혁준	ESP8266 Typescript Node.js AWS (서버 & DB) 선행기술조사	ESP8266 설계 Web 통신부분 설계 콘텐츠 로직	콘텐츠 구현 목업 디자인 조립	단위 테스트 콘텐츠 테스트
오우석	X	DB 설계 콘텐츠 로직 Web 통신부분 설계	콘텐츠 구현 목업 디자인 조립	단위 테스트 콘텐츠 테스트
공동	콘텐츠 개발,유지보수			

종합설계 수행일정

추진사항	12월	1월	2월	3월	4월	5월	6월	7~9월
조사 및 역할 배정	■ ■ ■							
요구사항 정의 및 분석		■ ■ ■						
시스템 설계 및 상세설계			■ ■ ■	■ ■ ■	■ ■ ■			
구현 및 프로토타입 제작				■ ■ ■	■ ■ ■			
통합 테스트					■ ■ ■	■ ■ ■		
산업기술대전 준비					■ ■ ■	■ ■ ■		
문서화 및 발표						■ ■ ■	■ ■ ■	
최종보고서 작성								■ ■ ■

필요기술 및 참고문헌

필요기술	참고문헌
Node.js	https://nodejs.org/ko/docs/
Angular	http://angular.io/
Express	http://www.expressjs.com
Arduino	www.Arduino.cc
Typescript	https://www.typescriptlang.org/docs/tutorial.html
jQuery	www.jquery.com
AWS (Amazon Web Service)	https://aws.amazon.com/ko/getting-started/?nc2=h_l2_cc

Q & A

별첨자료

지도학생	지도교수	서명란
2014150018 서준원	서대영교수님	
2012151002 권혁준	한익주교수님	
2012151029 오우석		