

부루마블
PARK GYEONGHO

라인컴퓨터아트학원
박경호

TITLE
INDEX



01

SCHEDULE

page 03 - 04

02

DESIGN

page 05 - 07

03

FOLDER STRUCTURE

page 08 - 09

04

DESCRIPTION

page 10 - 17

05

CODE

page 18 - 23

06

REVIEW

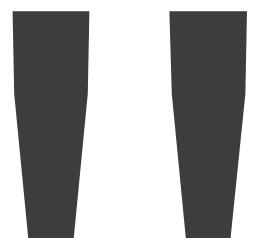
page 24 - 25

Page. 2



SCHEDULE

PARK GYEONGHO



TITLE

SCHEDULE

10.31

DRAWING 작업
필요 기능 및 구조 정리

11.01

각 도시별 정보 JSON 작업
PUG, SASS 레이아웃 작업

11.02

JS CLASS 구조화 및 작업

11.03

JS CODE 세부 작업
NODE.JS, SQL 연동

11.04

테스트 및 버그 수정 작업

11.05

PPT 자료 정리





DESIGN

PARK GYEONGHO





TITLE DRAWING



- 좌, 우측에 각 유저 정보 기입
- 승, 패 순위표 DB 데이터 기입
- 게임판 3D 속성으로 기울어져 보이게 표현
- 가운데 주사위 굴러가는 모션 표현



TITLE DRAWING

- 게임 상황에 맞는 여러 팝업창 레이아웃

도시 이름 X

토지 빌라 빌딩 호텔

구매하기 100.0만

통행료 > 100.0만

관광지 이름

그림 단계별 통행료

구매하기 100.0만

닫기

통행료 > 100.0만

000 님이 무인도에 갇히셨습니다.
3 차례 동안 이동을 할 수 없습니다.

20만원 지출 닫기

도시 인수

서울 인수할까요?

구매하기 100.0만

닫기

인수 후 남는 마블 > 100.0만

캐릭터 선택

닉네임을 입력해주세요.

로그인

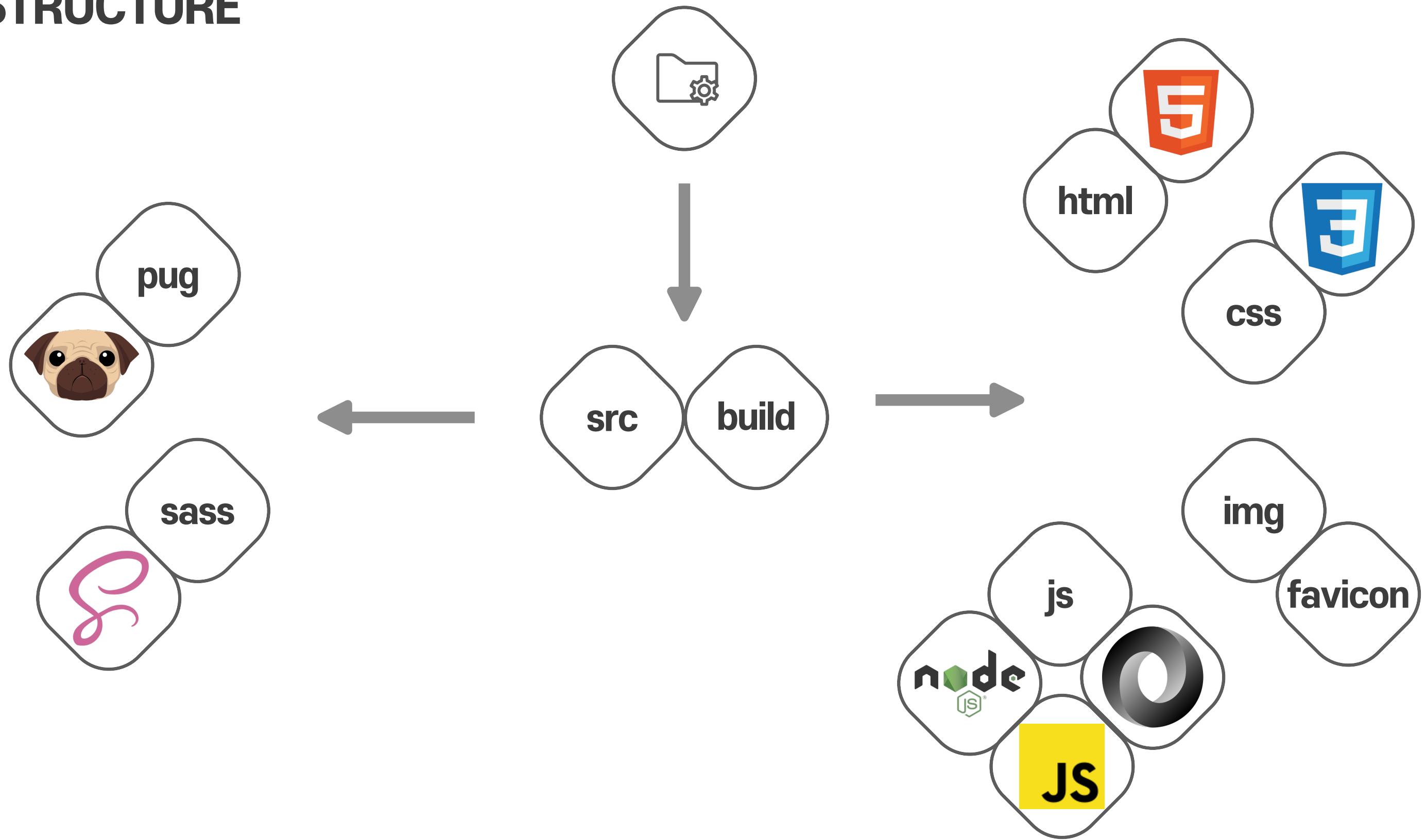


FOLDER STRUCTURE

PARK GYEONGHO

TITLE

FOLDER STRUCTURE





DESCRIPTION

PARK GYEONGHO



TITLE

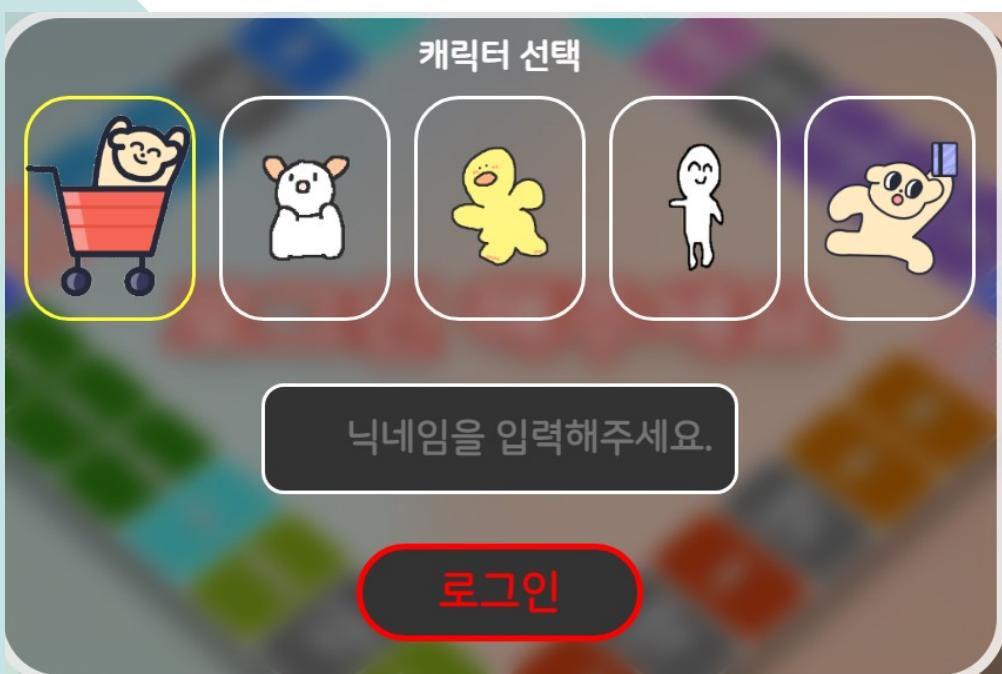
DESCRIPTION



로그인

유저 2명이 각각 원하는 캐릭터와 닉네임을 선택할 수 있다.

2명이 로그인 되면 주사위가 활성화된다.



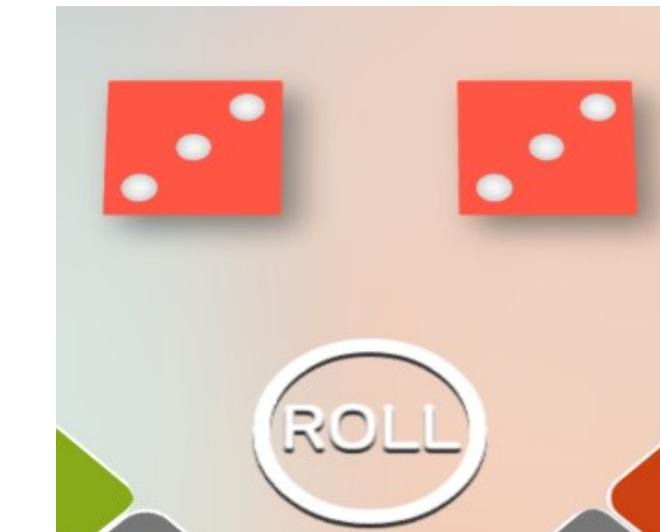
순위

유저 닉네임의 승, 패의 횟수에 따른 순위를 확인할 수 있다.

순위표				
순위	닉네임	승	패	
1	TEST2	3	2	
2	TEST3	2	0	
3	www	1	0	
4	TEST4	1	1	
5	TEST1	1	4	
6	3333	0	1	

주사위

2개의 주사위의 나온 수만큼 이동하며 같은 수가 나오면 기회가 추가된다.





출발

게임을 시작하는 위치이며
한 바퀴를 돌면 30만 원 지급



무인도

3년 동안 이동할 수 없다.
20만 원 지출 or 주사위 숫자가
같을 경우 탈출 가능



올림픽

자신이 보유하고 있는 도시에
올림픽을 개최할 수 있다.
X2 ~ X5까지 중첩 가능
다른 도시에 개최 시 초기화



세계여행

세계여행을 제외한 모든 곳으로
이동할 수 있다.

TITLE

DESCRIPTION



CITY

표기된 곳은 "도시"이며 땅, 빌라, 빌딩, 호텔을 지을 수 있다
각각 "도시"마다 건물별 금액, 통행료가 다르며 상대편이
자신의 "도시"에 방문 시 통행료를 지불해야 하며
상대편은 해당 "도시"를 구매 비용의 2배로 인수할 수 있다.

도시 인수

제네바 인수할까요?

주의: 건설비용의 2배지불

15

76.0만

취소

인수 후 남는 마늘

-48.0만

카이로



구매 6.0만

통행료 ▶ 1.0만



Page. 13

TITLE

DESCRIPTION



ISLAND

한 유저가 "섬 관광지" 3개 중 하나를 구매 시 깃발,
두 개를 구매 시 파라솔, 세 개를 구매 시 방갈로로 변경된다.
변경 시 해당 유저가 보유한 모든 "섬 관광지"가 변경된다.
다른 유저가 인수할 수 없으며 구매비는 하나당 10만 원이다.



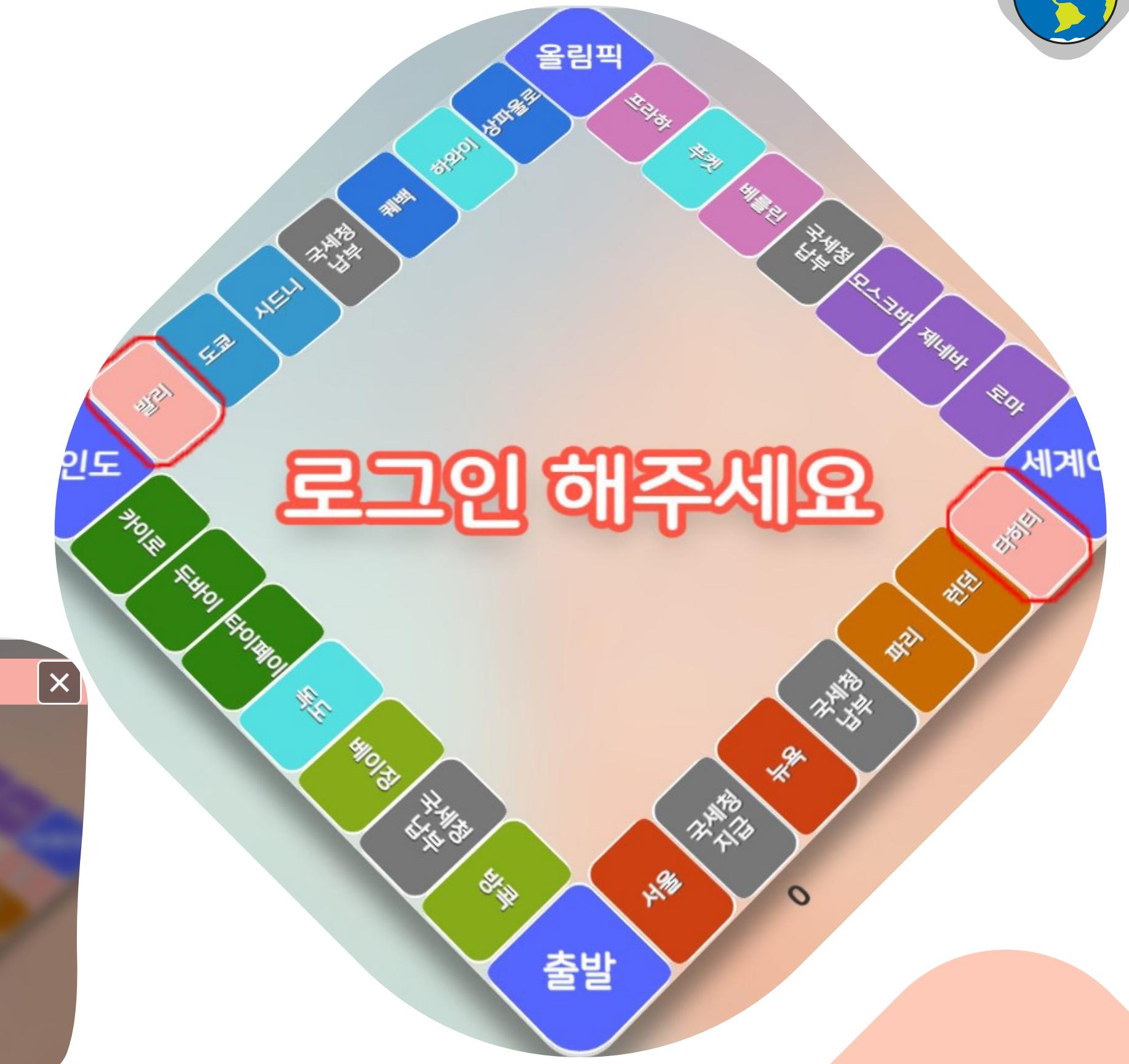
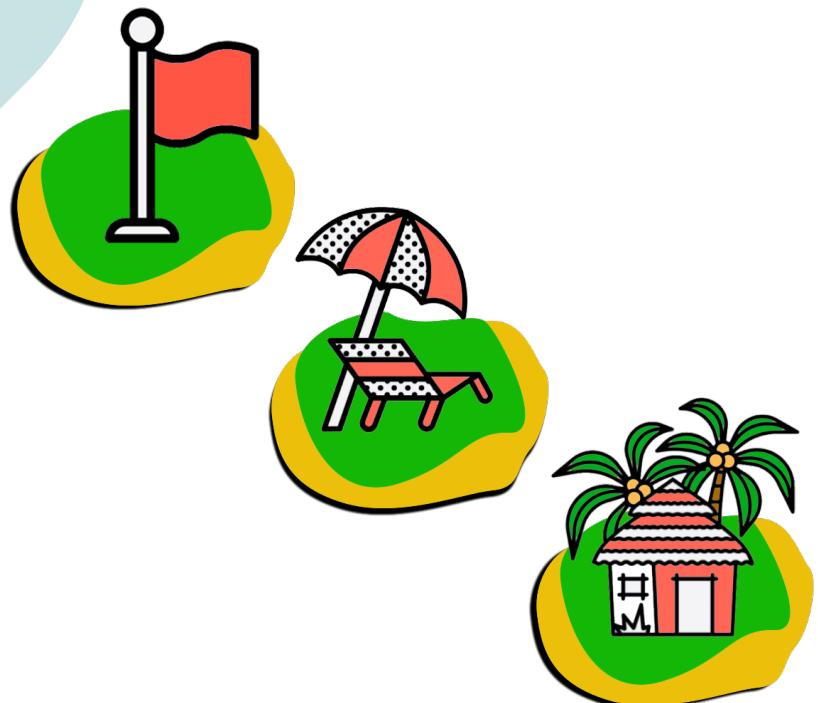
TITLE

DESCRIPTION



BEACH

"해변 관광지"는 단 한번만 10만 원으로 구매 가능하며
소유자, 상대편 중 아무나 방문하면 한 단계씩 상승한다.





TITLE

DESCRIPTION

TAX

"국세청 납부"에 도착하면 20만 원의 세금이 차감되며

"국세청 지급"에 도착하면 그동안 쌓인 세금을 한 번에 받는다.

보너스가 지급 되었습니다.

+ 20만원

달기

세금이 납부 되었습니다.

- 20만원

닫기



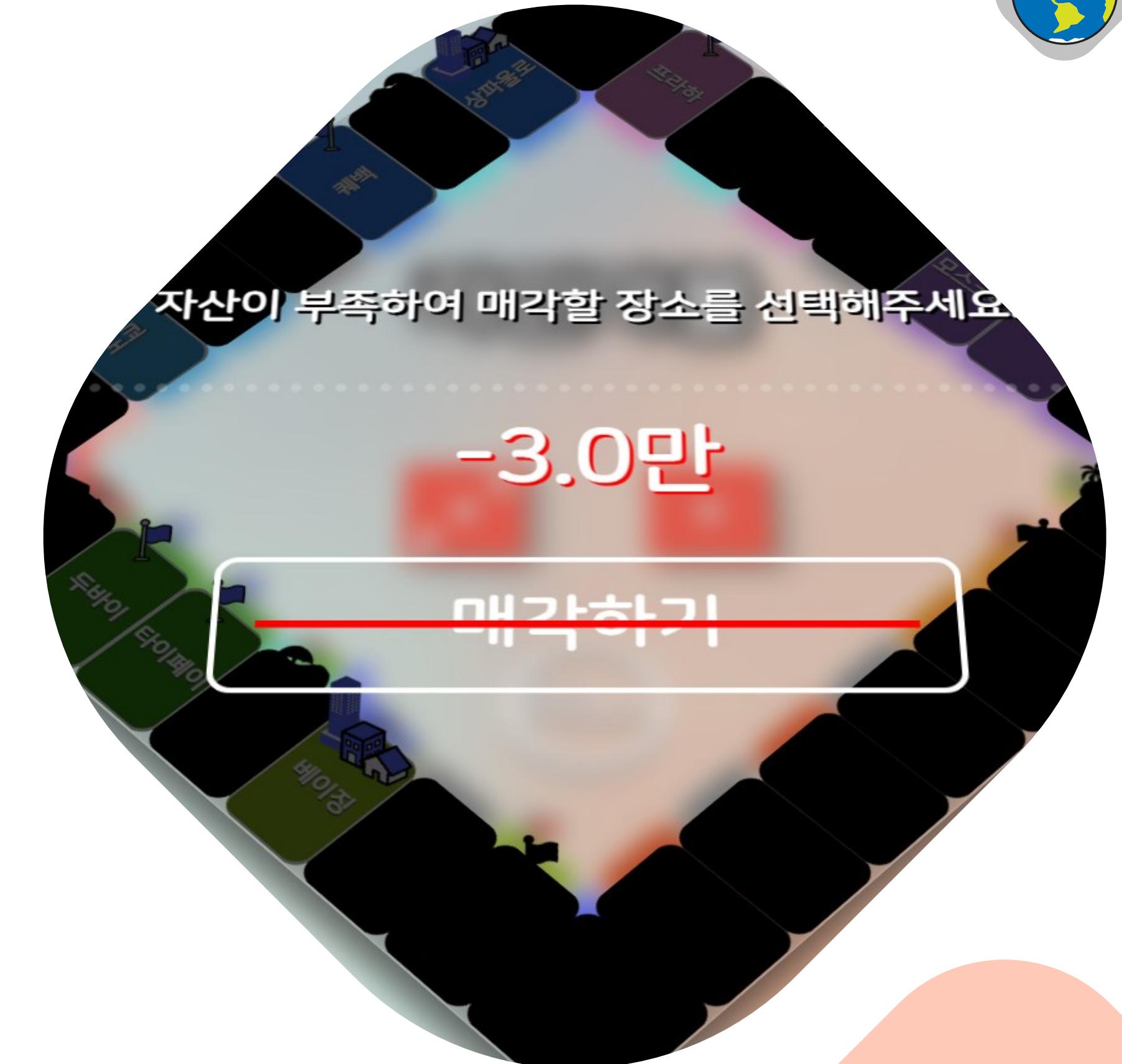
TITLE

DESCRIPTION



매각 & 게임중로

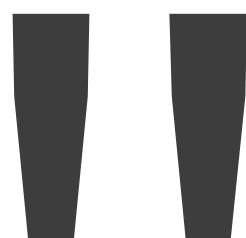
소지한 돈이 "통행료" or "세금 납부" 보다 적으면
해당 유저가 소유한 장소를 선택하여 매각할 수 있다.
만약 모든 장소를 매각하여도 돈이 부족할 경우 자동으로 패배한다.





CODE

PARK GYEONGHO





TITLE CODE - JSON

```
marble_data = [
  {
    "id" : 1,
    "type": "city",
    "name" : "방콕",
    "country" : "태국",
    "color" : "#86AA18",
    "build_cost" : { "land" : 20000, "villa" : 10000, "building" : 40000, "hotel" : 6000 },
    "pass_cost" : { "land" : 5000, "villa" : 10000, "building" : 20000, "hotel" : 60000 },
    "take_cost" : { "land" : 40000, "villa" : 20000, "building" : 80000, "hotel" : 120000 },
    "sell_cost" : { "land" : 10000, "villa" : 5000, "building" : 20000, "hotel" : 30000 }
  },
  ...
],
{
  "id" : 4,
  "type" : "island",
  "name" : "독도",
  "country" : "한국",
  "color" : "#54E0E1",
  "build_cost" : 100000,
  "pass_cost" : { "land" : 80000, "parasol" : 160000, "bungallow" : 320000 },
  "take_cost" : null,
  "sell_cost" : { "land" : 50000, "parasol" : 50000, "bungallow" : 50000 }
},
...
]
```

방콕 모두의마블

	땅	빌라	빌딩	호텔	랜드마크
건설비용	2만	1만	4만	6만	6만
통행료	5천	1만	2만	6만	25만
인수비용	4만	2만	8만	12만	인수불가
매각비용	1만	5천	2만	3만	3만

독도·섬 관광지 모두의마블

건설비용 → 10만

통행료		매각비용	
땅	8만	땅	5만
파라솔	16만	파라솔	5만
방갈로	32만	방갈로	5만



TITLE CODE - JavaScript

```
class User_module {  
    constructor(_id, _code){  
        this.id = _id;  
        this.code = _code;  
        this.money = 1000000;  
        this.land = [];  
        this.standing = 0;  
        this.state_island = 0;  
        this.state_tax_set = 0;  
        this.state_tax_get = 0;  
        this.state_olympic = 0;  
        this.state_travel = 0;  
        this.rotate_count = 0;  
        this.win_count = 0;  
        this.loss_count = 0;  
    }  
  
    let user_module_1;  
    let user_module_2;  
  
    class Place_module {  
        constructor(_JSON){  
            this.id = _JSON.id;  
            this.name = _JSON.name;  
            this.owner = "none";  
            this.level = [1, 0, 0, 0];  
            this.set_level = [0, 0, 0, 0];  
            this.island_level = 0;  
            this.build_cost = _JSON.build_cost;  
            this.pass_cost = _JSON.pass_cost;  
            this.take_cost = _JSON.take_cost;  
            this.sell_cost = _JSON.sell_cost;  
            this.color = _JSON.color;  
            this.type = _JSON.type;  
            this.event = 1; // 올림픽 1, 2  
            this.sell_check = 0;  
        }  
  
        sell_land = () => { ...  
    }  
}
```

- 유저의 정보를 담는 CLASS 구조 구현
- 각 지역별 정보를 담는 CLASS 구조 구현
- **sell_land**
 - ↳ 해당 지역 매각 Method



```
class Program_module {  
    constructor(_data_JSON){ ...  
    }  
    create_map = () => { ...  
    }  
    move_obj = (_obj, _target, _time) => { ...  
    }  
    stand_check = (_module) => { ...  
    }  
    click_position = (_move_module, _view_type) => { ...  
    }  
    monitoring_game = () => { ...  
    }  
}  
const program = new Program_module(marble_data);  
program.create_map();
```

TITLE

CODE - JavaScript

- **create_map**

↳ 각 지역 정보를 담은 발판 생성 Method

- **move_obj**

↳ 유저 캐릭터 움직임 및 게임 턴 관련 Method

- **stand_check**

↳ 도착 장소에 따라 이벤트 발생 Method

- **click_position**

↳ 올림픽, 세계여행, 매각 관련 담당 Method

- **monitoring_game**

↳ 실시간으로 변경되는 값을 읽는 Method



TITLE CODE - Node.JS

```
const express = require('express');
const mysql = require('mysql');
const body_parser = require('body-parser');
const app = express();
const con = mysql.createConnection({ ...
});

app.use(express.static(__dirname + ' ../../'));
app.locals.pretty = true
app.set('view engine', 'pug');
app.set('views', './src/pug/');
app.use(bodyParser.urlencoded({extended: false}));

app.get('/marble', (req,res)=>{
    console.log('/marble URL Connected');
    let select_list = `SELECT * FROM marble ORDER BY win DESC, l
con.query(select_list, (err, result, field)=>{
    if(err) throw err;
    res.render('marble', {list: result});
})
});
```

- Express 기반 Pug, MySQL 모듈 사용
- GET 실행 시 DB TABLE의 RECORD를 승, 패 횟수에 따른 정렬 후 Pug에 전달
- Pug에서 해당 배열을 반복문을 통해 출력



TITLE CODE - Node.JS

```
app.post('/marble', (req,res)=>{
  console.log('POST SEND DATA');
  let new_check = [0, 0];
  let win_name = req.body.win_name;
  let loss_name = req.body.loss_name;
  let name_list = [win_name, loss_name];
  let win_count = [1, 0]; //--- 승자 승, 패 횟수
  let loss_count = [0, 1]; //--- 패자 승, 패 횟수
  let count_list = [win_count, loss_count];
  let select_list = `SELECT * FROM marble ORDER BY win DESC, loss ASC`;
  con.query(select_list, (err, result, field)=>{
    if(err) throw err;
    result.forEach((v,i,a)=>{
      if(v.name == win_name){...}
      }
      if(v.name == loss_name){...}
    })
    new_check.forEach((v,i,a)=>{
      if(v == 0) {
        con.query(`INSERT INTO marble (name, win, loss) VALUES ("${name_list[i]}",
        ${count_list[i][0]}, ${count_list[i][1]})`, (err,result,field)=>{});
      } else {
        console.log( count_list[i][0] + ' / ' + count_list[i][1] + ' / ' +
        name_list[i] );
        con.query(`UPDATE marble SET win=${count_list[i][0]}, loss=${count_list[i]
        [1]} WHERE name = "${name_list[i]}"`, (err,result,field)=>{});
      }
    })
    setTimeout(()=>{
      res.redirect('/marble');
    }, 500)
  })
}
```

- POST 실행 시 Body-Parser으로 값 받아오기
- 기존 Table 값과 비교하여 해당 닉네임이 기존 유저 & 신규 유저인지 검사
- 기존 유저일 경우 UPDATE 진행
- 신규 유저일 경우 INSERT 진행



REVIEW

PARK GYEONGHO





후기

꾸준히 성장 중...



- 이번 부루마블 게임을 제작하기에 앞서 직접 보드게임을 사서 게임 규칙을 다시 읽어보고 JSON, 지역 구조를 클래스로 구성하기에 필요 정보를 사전 정리하였습니다. 그럼에도 코드 작업하면서 필요한 부분이 추가로 생겨 중간중간 막혔을 때도 있어 사전 작업을 더욱 길게 가져가야 할 필요성을 느꼈습니다.
- 학원 초반에는 직접 이와 같은 게임을 코드로 만들 수 있을지 상상도 못했었지만 완성을 하고 나서 짧은 시간 내에 빠르게 성장하고 있다는 것을 크게 느꼈으며 더욱 자유롭게 코드 작업을 할 수 있도록 꾸준히 배워나갈 것입니다.

박경호 | PARK GYEONGHO

Thank you!

감사합니다.