
PORTFOLIO

Web개발자 유영훈 포트폴리오

이름	유 영 훈
생년월일	1991. 01. 30
이메일	nice130@naver.com
전화번호	010-3099-3265
주소	인천시 중구

PORTFOLIO

CONTENTS

- | | | |
|-----|------------------|--|
| 01. | Planning 웹페이지 개발 | 개인 사이드 프로젝트로 React 사용 경험 및 숙련도 향상을 위하여 진행함 |
| 02. | 룰렛게임 개발 | 미니게임 Web/App를 제작하기 위해 만든 첫번째 게임! |
| 03. | 사다리타기게임 개발 | 미니게임 Web/App를 제작하기 위해 만든 두번째 게임! |

Project

Planning 웹페이지 개발

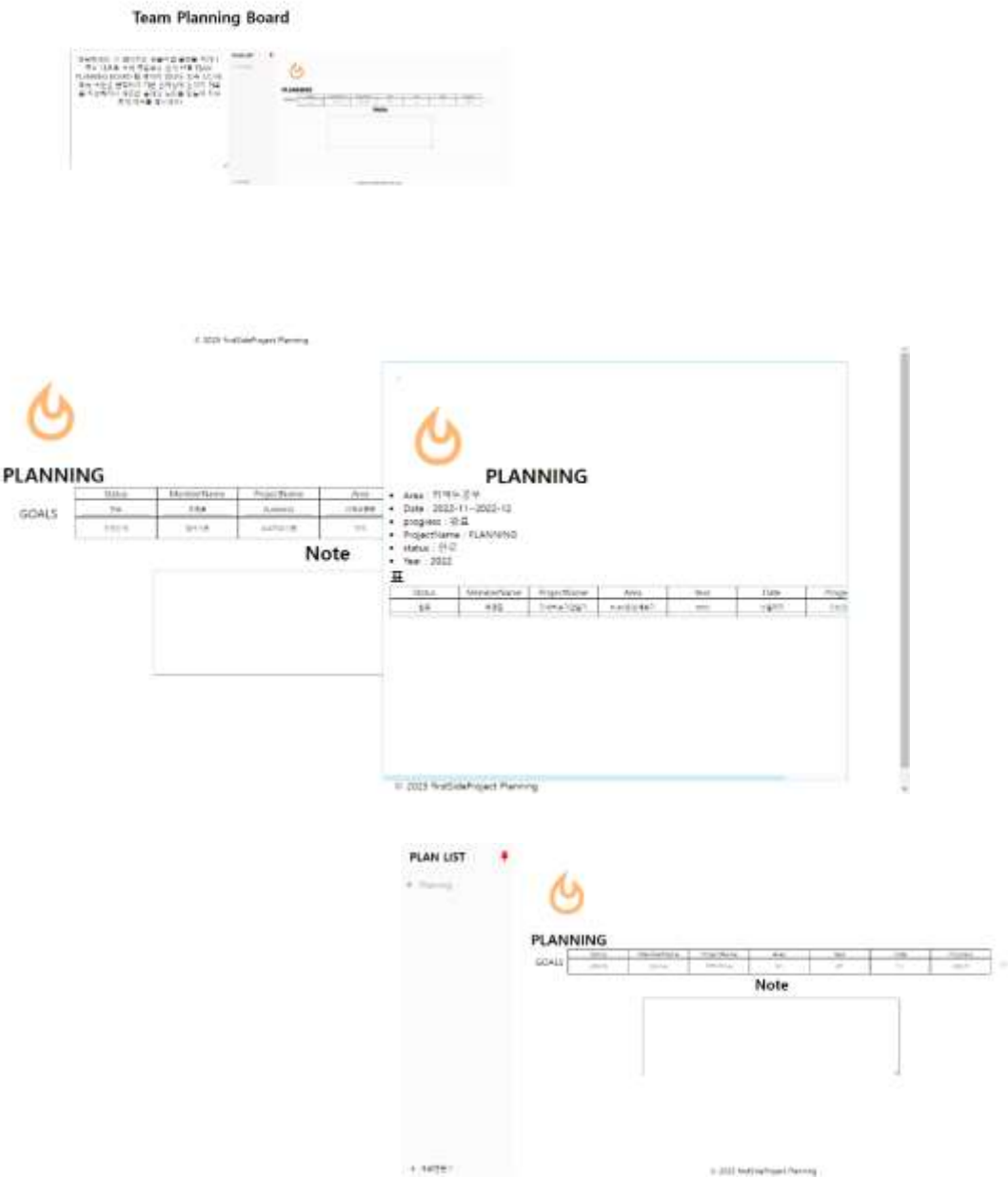
About project



- 개인 프로젝트로 React 사용 경험 및 숙련도 향상을 위하여 진행
- PLAN을 만들어 진행도와 목표등을 입력하여 기록할 수 있다

Introduce project

작업 기간	2022.11~2022.12
인력 구성(기여도)	2명 기여도 90%이상
프로젝트 목적	<div><div>- React 사용 경험 및 숙련도 향상을 위하여 진행함</div><div>- STUDY 멤버와의 계획을 공유하는 페이지 개발</div></div>
프로젝트 내용	<div><div>- 팀의 협업을 위한 페이지</div><div>- PLAN을 세우고 그에 대한 상세내용을 공유하는 공간</div></div>
주요 업무 및 상세 역할	<div><div>- 전체적인 구성을 잡고 화면에 어떤 UI로 넣을지 계획</div><div>- Home, Planning, SideBar Modal 페이지 개발</div></div>
사용언어 및 개발 환경	React, JavaScript, Visual Studio Code
참고 자료	<div>Github 링크 : https://github.com/nice130/planning</div> <div>web 링크 : https://nice130.github.io/planning/</div>



HomePage, PlanningPage 소개

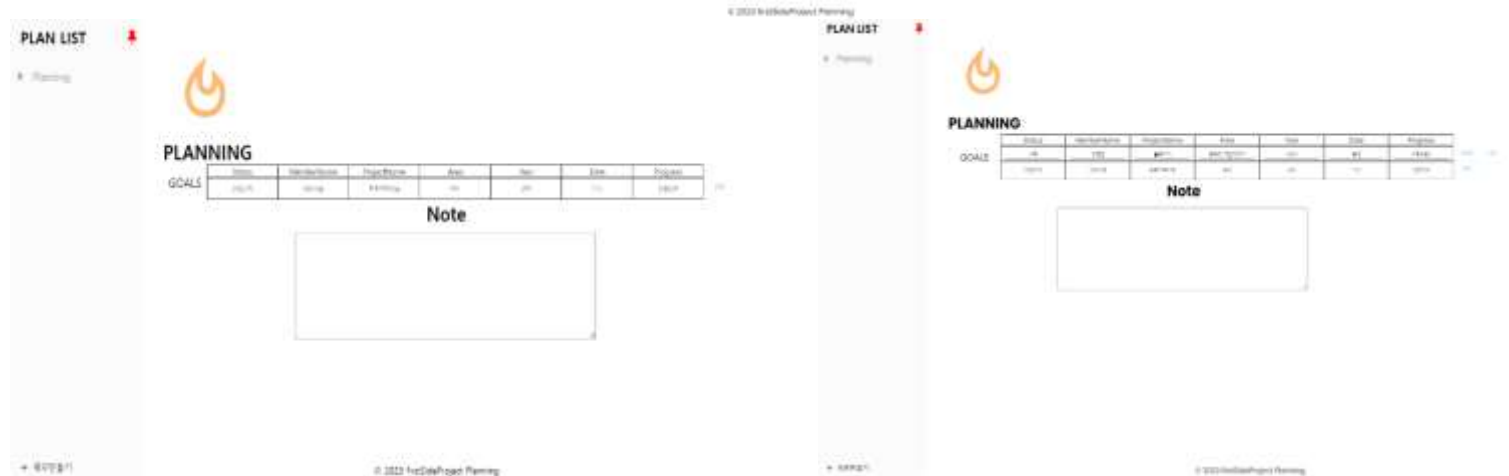
• HomePage

- Web페이지 이용 예시 영상
- 기본적인 기능을 소개



• PlanningPage

- 팀 또는 개인의 plan을 상태와 정보를 맞춰 입력
- Note란에 플랜에 대한 추가적인 내용을 입력
- 입력 후 별도의 수정 버튼 없이 바로 수정
- 자세히 버튼을 눌러 계획에 상세계획 설정



Home.js

Planning.js

```

// Form.js
import { useState } from 'react';
import styles from './styles.module.css';
import Detail from './Detail';

const columns = [
  {
    status: { placeholder: "状态选择", value: "" },
    name: { placeholder: "姓名", value: "" },
    room: { placeholder: "房间号", value: "" },
    area: { placeholder: "面积", value: "" },
    year: { placeholder: "年份", value: "" },
    date: { placeholder: "日期", value: "" },
    progress: { placeholder: "进度", value: "" },
  },
];

function toggleToggle, toggleToggle() {
  const toggleObject = Object.keys(columns).reduce((object, value) => {
    object[value] = !object[value];
    return object;
  }, {});

  const [val, setVal] = useState({});
  const [detail, setDetail] = useState({});
  const toggleDetail = () => {
    const id = s.target.dataset.id;
    setVal({ ...val, [id]: !val[id] });
    setDetail({ id: id });
  };

  const { ...values, setValues } = useState({});
  const { count, setCount } = useState(0);
  const { value, setValue } = useState({ ...detailObject });

  const exchange = () => {
    const { name, value } = s.target;
    setValues({
      ...values,
      [name] = value,
    });
  };

  const exchange2 = () => {
    const { name, value } = s.target;
    const id = s.target.dataset.id;
    setValues([id] ? { ...values, [name] = value } : {
      ...values,
      [name] = value,
    });
  };

  const exchange3 = () => {
    const { name, value } = s.target;
    const id = s.target.dataset.id;
    setValues([id] ? { ...values, [name] = value } : {
      ...values,
      [name] = value,
    });
  };
}

// Form.js.map
import { useState } from 'react';
import styles from './styles.module.css';
import Detail from './Detail';

const columns = [
  {
    status: { placeholder: "状态选择", value: "" },
    name: { placeholder: "姓名", value: "" },
    room: { placeholder: "房间号", value: "" },
    area: { placeholder: "面积", value: "" },
    year: { placeholder: "年份", value: "" },
    date: { placeholder: "日期", value: "" },
    progress: { placeholder: "进度", value: "" },
  },
];

function toggleToggle, toggleToggle() {
  const toggleObject = Object.keys(columns).reduce((object, value) => {
    object[value] = !object[value];
    return object;
  }, {});

  const [val, setVal] = useState({});
  const [detail, setDetail] = useState({});
  const toggleDetail = () => {
    const id = s.target.dataset.id;
    setVal({ ...val, [id]: !val[id] });
    setDetail({ id: id });
  };

  const { ...values, setValues } = useState({});
  const { count, setCount } = useState(0);
  const { value, setValue } = useState({ ...detailObject });

  const exchange = () => {
    const { name, value } = s.target;
    setValues({
      ...values,
      [name] = value,
    });
  };

  const exchange2 = () => {
    const { name, value } = s.target;
    const id = s.target.dataset.id;
    setValues([id] ? { ...values, [name] = value } : {
      ...values,
      [name] = value,
    });
  };

  const exchange3 = () => {
    const { name, value } = s.target;
    const id = s.target.dataset.id;
    setValues([id] ? { ...values, [name] = value } : {
      ...values,
      [name] = value,
    });
  };
}

```

자세한코드는 <https://github.com/nice130/planning> 에 있습니다.

Detail, NewPlan page 소개

• Detail

- Plan을 더 Detail 하게 세워 목표를 이루기 위한 Page
- 입력된 Plan의 자세히 버튼 클릭 시 Modal 오픈
- 팀원과 각자 맡은 부분을 작성하여 진행 정도를 공유



• NewPlan

- SideBar에 새로 만들기를 눌러 새로운 메뉴 생성



Detail.js

[illegible]

```
File Edit Selection View Go Run Terminal Help
# NewNote.js
import { routes } from './NewNote';
import { useEffect, useRef, useState } from 'react';

function NewNote({setModalOpen,setPlan,plan,savePlan,setSavePlan}){
  const closeModal= ()=>{
    setModalOpen(false);
  };
  const modalRef = useRef();

  useEffect(()=>{
    const handler = e =>{
      let sideChildren = modalRef.current.contains(e.target);
      if(modalRef.current && !sideChildren){
        setModalOpen(false);
      };
      document.addEventListener('mousedown',handler);
      return () =>{
        document.removeEventListener('mousedown',handler);
      };
    };
  });
  const [isId,setIsId] = useState(0);
  const onChange =(e)=>{
    const {value} = e.target;
    setPlan({
      title : value,
    });
  };

  const save =(e)=>{
    setIsId(isId+1);
    setSavePlan(array=>[...array,plan]);
    setPlan("");
    setModalOpen(false);
  }

  return(
    <div ref={modalRef} className={styles.modalparent}>
      <button className={styles.button} onClick={closeModal}>
        ✕
      </button>
      <h2 New Plan / 2 >
      <div className={styles.li}>플랜명 </div>
      <input
        className={styles.input}
        onChange={onChange}
        value={plan.title}</input>
      <button onClick={save}>저장 </button>
    </div>
  );
}
```

자세한코드는 <https://github.com/nice130/planning> 에 있습니다.

Project 02.

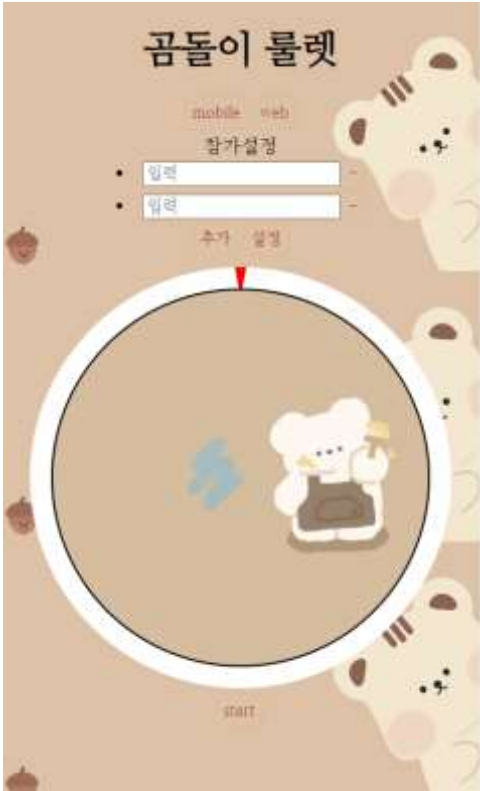
<미니게임 Web/App 만들기>

- 룰렛

About project



- 미니 게임Web/App 을 구성하기 위한 첫 번째 게임
- 룰렛, 사다리타기, 새로 구상한 게임 3개 개발 예정



Introduce project

작업 기간	2022.12~2022.12
인력 구성(기여도)	개인프로젝트
프로젝트 목적	- 다양한 미니게임을 Web/App에 모으고 싶어 개발
프로젝트 내용	- 처음 참가 설정을 하여 게임을 시작 - 추가를 눌러 입력 칸 생성 (2개 이상 설정이 되어야 함) - Retry 시에도 룰렛판을 reset 시키지 않고 계속 게임 진행 가능
주요 업무 및 상세 역할	- Web/App에서 게임을 진행 할 수 있는 환경을 구성 - 게임 진행 후에도 Reflash없는 부드러운 UI 제작 - 결과 핀이 일정하게 멈추지 않고 실제 룰렛처럼 STOP
사용언어 및 개발 환경	JavaScript, Visual Studio Code
참고 자료	Github 링크 : https://github.com/nice130/roulette web 링크 : https://nice130.github.io/roulette/

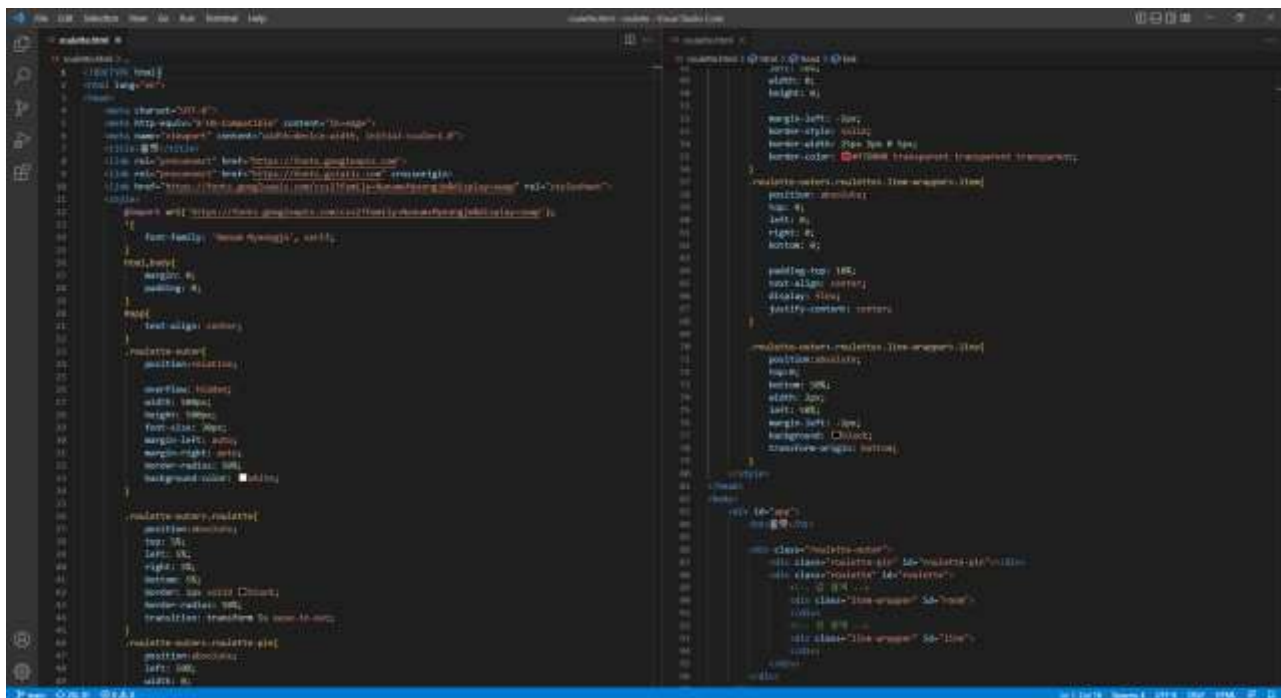
Roulette.html

- 게임방법
 - mobile/web 중 환경 선택.
 - 추가버튼과 (-)버튼으로 입력 개수 설정
 - 판에 넣을 내용을 입력
 - Start버튼 클릭 시 게임 시작
 - Retry버튼 클릭 시 게임 재시작 가능
- 주요 기능
 - 어떤 버튼을 눌러도 Refresh 없이 진행
 - 추가/삭제 한 후 설정해도 판을 새로 고치지 않음
 - 실제 룰렛을 하는 것과 같이 진행

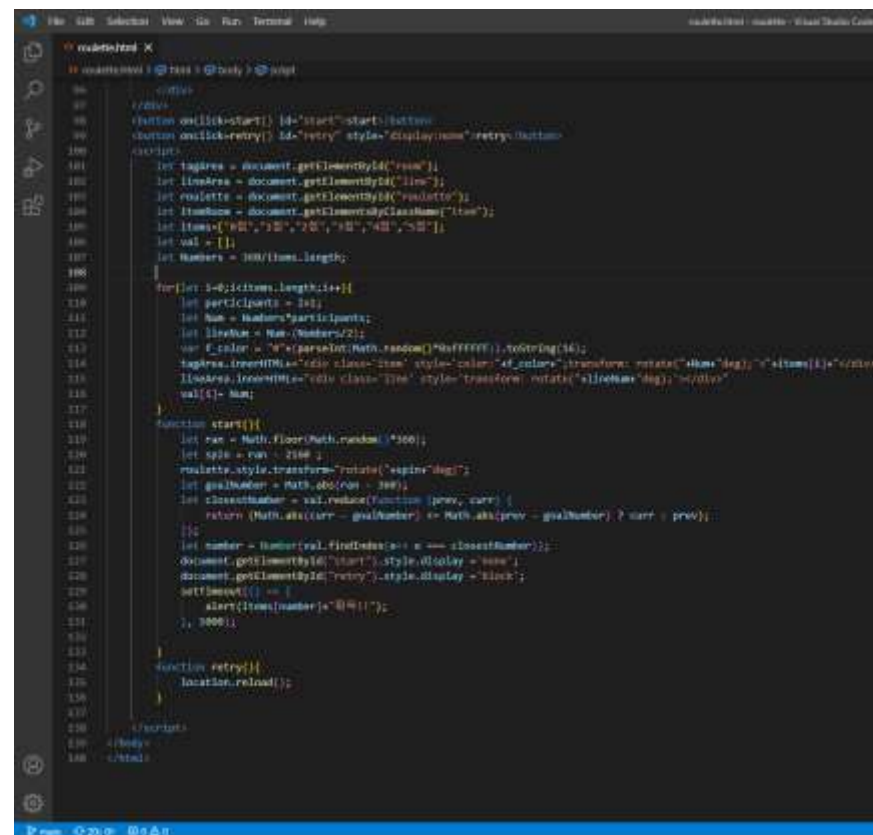


roulette.html 코드

html부분



script 부분



자세한코드는 <https://github.com/nice130/roulette> 에 있습니다.

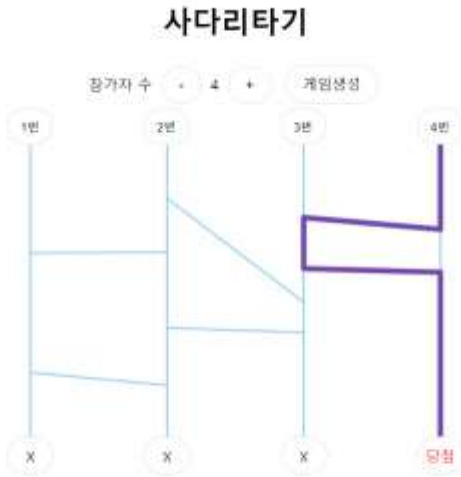
Project 03.

<미니게임 Web/App 만들기>

- 사다리타기

About project

- 미니 게임Web/App 을 구성하기 위한 두 번째 게임
- 룰렛, 사다리타기, 새로 구상한 게임 3개 개발 예정



Introduce project

작업 기간	2022.12~2023.01
인력 구성(기여도)	개인프로젝트
프로젝트 목적	- 다양한 미니게임을 Web/App에 모으고 싶어 개발
프로젝트 내용	- 참가자 수를 입력하여 게임 생성 - 마우스로 자유롭게 선을 이어 게임 판 제작 - 참가자 번호를 눌러 결과 확인
주요 업무 및 상세 역할	- 참가자 수에 맞게 게임 판 생성 - 자유로운 이음선을 만들 수 있게 제작 - 결과를 버튼을 누르기 전까지 알 수 없게끔 X로 표시
사용언어 및 개발 환경	JavaScript, Visual Studio Code
참고 자료	Github 링크 : https://github.com/nice130/ladderGame web 링크 : https://nice130.github.io/ladderGame/

Roulette.html

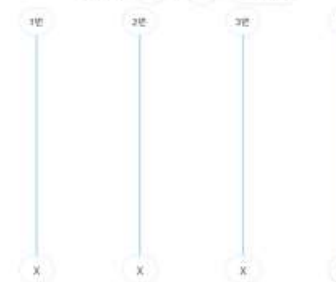
- 게임방법
 - 참가자 수를 선택 후 게임 생성
 - 각 번호의 Y축 선과 선을 이어 사다리 제작
 - 번호를 클릭하여 결과 확인
- 주요 기능
 - 참가자 수에 따라 게임 판을 reflash없이 바로 생성
 - 마우스로 원하는 사다리를 제작
 - 당첨은 번호를 클릭하기 전까지 나오지 않음

사다리타기

참가자 수 - 2 + 게임생성!

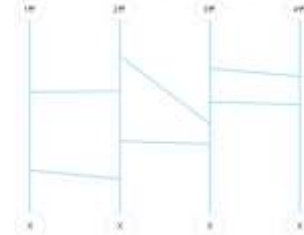
사다리타기

참가자 수 - 4 + 게임생성



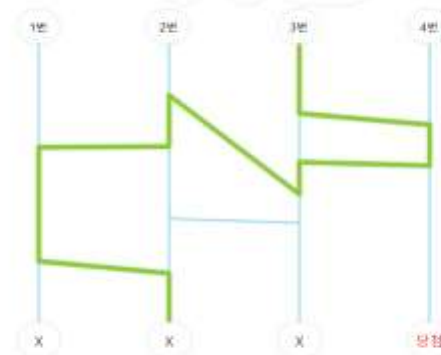
사다리타기

참가자 수 - 4 + 게임생성



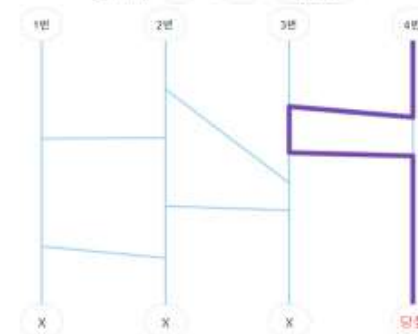
사다리타기

참가자 수 - 4 + 게임생성



사다리타기

참가자 수 - 4 + 게임생성



html부분

```
File Edit Selection View Go Run Terminal Help
index.html - ladderGame

index.html X
  index.html > html > body > script > arraySort
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>사다리타기</title>
8   <style>
9     @import url("ladderStyle.css");
10  </style>
11 </head>
12 <body>
13   <div id="app">
14     <div>사다리타기</div>
15     <div id="participant">
16       <!-- 인원수 설정 -->
17       참가자 수
18       <button class="btn1" onclick="participant(this)" value="minus">-</button>
19       <!-- 인원번호추가 -->
20       <span id="participantNumber"></span>
21       <button class="btn1" onclick="participant(this)" value="plus">+</button>
22       <button class="btn1" onclick="start()">게임생성</button>
23     </div>
24     <div style="text-align: center;">
25       <div id="game_field">
26         <!-- 벽돌배열 -->
27       </div>
28     </div>
29     <canvas id="canvas" width="600" height="500"> </canvas>
30   </div>
31
32   <script>
33     var canvas = document.getElementById("canvas");
34     var ctx = canvas.getContext("2d");
35     let width = canvas.width;
36     let height = canvas.height;
37   </script>
```

script 부분

```
File Edit Selection View Go Run Terminal Help
index.html - ladderGame - Visual Studio Code

index.html X
  index.html > html > body > script > arraySort
  index.html > html > body > script > arraySort
38   function start(option=1){
39     let random = 0;
40     participant = [];
41     $game_field.innerHTML +=
42     init();
43     data = [];
44     lineData = [];
45     line = option.line || participant.length;
46     for(let i=0; i<participant.length; i++){
47       participant[i] = i+1;
48     }
49     random = Math.floor(Math.random()*participant.length);
50     for(let i=0; i<participant.length; i++){
51       let coordinatesX = (1 / participant.length * width * i) + random;
52       let num = i+1;
53       $game_field.innerHTML += <button class="btn" type="button" style="float:
54       left; margin-right: 10px;">
55       <div class="btn" id="res_1" style="float: left; margin-right: 10px;">
56       $game_field.innerHTML += <div class="btn" id="res" style="float: left; margin-right: 10px;">
57       </div>
58       $game_field.innerHTML += <div class="btn" id="res_1" style="float: left; margin-right: 10px;">
59       </div>
60     }
61     for(let i=0; i<participant.length; i++){
62       let id = "btn" + (i+1);
63       document.getElementById(id).addEventListener("click", (event) =>{
64         ladderGame.find(i, "x" + (Math.random()*0xfffff).toString(16));
65       });
66     }
67   }
68   lineY();
69   //벽돌배열 그리기
70   function lineY(){
71     for(let i = 0; i < participant.length; i++){
72       let coordinatesX = (1 / participant.length * width * i) + random;
73       ctx.save();
74       ctx.beginPath();
75       ctx.lineWidth = defaultLineWidth;
76       ctx.strokeStyle = defaultLineColor;
77       ctx.moveTo(coordinatesX, height * 0.1);
78       ctx.lineTo(coordinatesX, height * 0.9);
79       ctx.stroke();
80       ctx.closePath();
81       ctx.restore();
82       let arr = [
83         {x: coordinatesX, y: height * 0.1},
84         {x: coordinatesX, y: height * 0.9}
85       ];
86       data.push(arr);
87     }
88     arraySort();
89   }
90   //인원수 변경
91   const participant = {val: 1};
92   value = val.value;
93   // $game_field.getElementById("participantNumber").innerHTML += value;
94   if(value == "minus"){
95     if(participant.val == 2){
96       alert("참가자는 2명 이상이어야 합니다.");
97       return;
98     }
99     participant.val--;
100     $participantNumber.innerHTML = participant.val;
101   } else {
102     participant.val++;
103   }
104 }
```

자세한코드는 <https://github.com/nice130/ladderGame> 에 있습니다.

End of Document
