



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	Code On Board
팀 명	알고리즘 서비스 연구소
문서 제목	결과보고서

Version	2.0
Date	2020-JUN-06

팀원	최 명서 (조장)
	최 지욱
	강 수련
	박 호준
	우 현웅
	칼 리드

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인Ⅰ 수강 학생 중 프로젝트 "Code On Board"를 수행하는 팀 "알고리즘 서비스 연구소"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "알고리즘 서비스 연구소"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	결과보고서-Code On Board.doc
원안작성자	최명서,
수정작업자	최명서, 최지욱, 강수련, 박호준, 우현웅

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-06-06	최명서	1.0	최초 작성	전체적인 내용 작성
2020-06-08	우현웅	1.1	내용 추가	API 연구/개발 내용 작성
2020-06-08	최지욱	1.3	내용 추가	클라이언트 연구/개발 내용 작성
2020-06-08	강수련	1.4	내용 수정	내용 수정 및 기대효과 추가
2020-06-08	최명서	1.5	내용 추가	Core 연구/개발 내용,
2020-06-08	박호준	1.6	내용 추가	문제 만들기 기능 내용 추가
2020-06-09	최명서	1.7	내용 추가	자기평가 작성
2020-06-09	우현웅	1.8	내용 추가	참고문헌 추가/ 자기평가 내용 추가
2020-06-09	최명서	2.0	내용 추가/수정	전체적인 내용 마무리

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
1.2.1	개발된 시스템 현황	4
1.2.2	문제점 및 개선방향	6
2	개발 내용 및 결과물	7
2.1	목표	7
2.2	연구/개발 내용 및 결과물	8
2.2.1	연구/개발 내용	8
2.2.2	시스템 기능 및 구조 설계도	24
2.2.3	활용/개발된 기술	27
2.2.4	현실적 제한 요소 및 그 해결 방안	27
2.2.5	결과물 목록	28
2.3	기대효과 및 활용방안	31
3	자기평가	32
4	참고문헌	34
5	부록	36
5.1	사용자 매뉴얼	36
5.1.1	다른 사용자와 대전	36
5.1.2	내 코드와 대전	42
5.1.3	코드 보기 및 수정	46
5.2	운영자 매뉴얼	47
5.2.1	문제 만들기	47
5.3	배포 가이드	52
5.3.1	클라이언트 서버	52
5.3.2	API 서버	52
5.3.3	Core 서버	52

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

1 개요

1.1 프로젝트 개요

본 프로젝트는 보드게임 기반의 알고리즘 대전을 제공하는 웹서비스를 개발하는 것이 목표이다. 사용자는 본 서비스를 이용하여 다른 사용자와 대전을 하고 점수경쟁을 통해 알고리즘 트레이닝을 지속할 수 있는 동기를 얻고 리플레이 기능과 내 코드와의 대전 기능을 통해 자신의 코드를 리뷰할 수 있는 기회를 시각적으로 얻을 수 있다.

1.2 추진 배경 및 필요성

1.2.1 개발된 시스템 현황

- 1) 백준 온라인 저지(Baekjoon Online Judge)



국내 최대 규모이고 가장 유명한 알고리즘 트레이닝 사이트로 다음과 같은 특징이 있다.

- 표준 입출력 기반으로 풀이를 작성.
- 약 50 가지 프로그래밍 언어를 지원.
- 단계별, 카테고리별 분류된 문제 제공
- 소요 시간, 메모리 사용 등 평가결과 제공
- 개인, 학교/기관 등 랭킹 제공
- 국내외 알고리즘 대회 기출문제 제공

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

- 커스텀 그룹 형성이 가능해 자체적인 대회 주최 가능

2) 프로그래머스

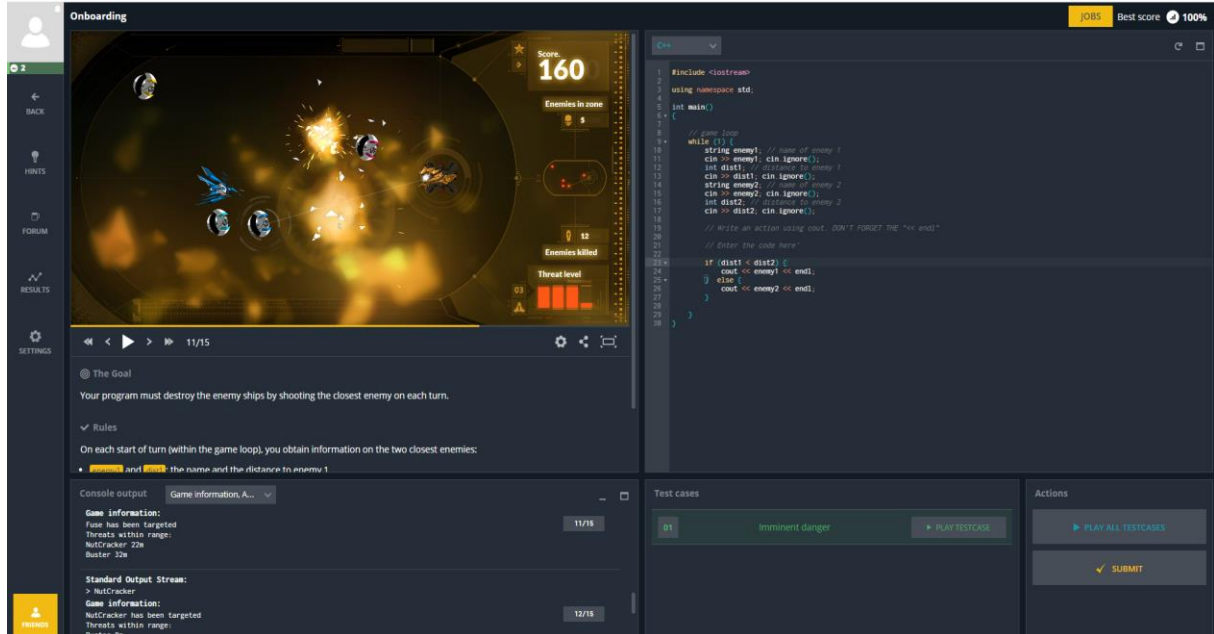


최근 코딩테스트에 특화된 플랫폼으로 다음과 같은 특징이 있다.

- 네이버, 카카오 등 국내 기업의 코딩테스트 환경 제공
- 솔루션 함수 작성 기반 문제 풀이
- 단계별 분류된 문제 제공
- 12 가지 프로그래밍 언어 제공
- 개인 랭킹 제공
- 가시적인 테스트 케이스 제공
- 백준에 비해 문제 수가 적음

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

3) Codinggame




해외 사이트로 코딩에 게임을 접목하였다. 다음과 같은 특징이 있다.

- 코딩에 게임을 접목함으로써 실제적이고 자극적인 코딩 도전 제공
- 많은 수요
- 2000 개 이상의 게임 카테고리 제공

1.2.2 문제점 및 개선방향

최근 컴퓨터적 사고를 기르기 위해 알고리즘 사이트를 이용하여 트레이닝을 하는 사람들의 수가 증가하고 있다. 동시에 국내외 알고리즘 문제풀이 사이트들에 대한 관심도 높아졌다. 수 많은 사람들이 위와 같은 사이트들을 이용하지만 알고리즘 트레이닝을 꾸준히 하는 것은 힘든 일이다. 문제에 대한 정답을 제출하고 테스트 케이스 검사를 통해 정답 유무만 알 수 있고 이 단순한 과정에서 흥미를 잃기 쉽기 때문이다. 따라서 본 프로젝트에서는 게임이라는 주제를 더해 흥미를 유발하고 랭킹 시스템을 제공하여 경쟁심을 자극함으로써 알고리즘 트레이닝을 지속할 수 있게 한다.

또한 트레이닝을 지속적으로 하는 사람들도 한번 맞춘 문제에 대해서는 다시 코드를 개선하지 않는 사람들이 많다. 하지만 정답을 맞췄다고 하더라도 자신의 코드를 리뷰하면서 개선하는 것이 코딩 실력 상승에 큰 도움이 되기 때문에 리플레이 기능과 내 코드와의 대전을 통해 알고리즘을 시각적으로 보여줌으로써 자연스러운 코드리뷰의 상황을 제공한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2 개발 내용 및 결과물

2.1 목표

게임과 대전이라는 주제를 통해 지속적인 알고리즘 트레이닝을 할 수 있도록 동기를 부여하고 리플레이 기능, 내 코드와의 대전을 통해 알고리즘을 시각적으로 표현함으로써 코드 리뷰에 도움을 줄 수 있는 웹서비스를 개발하는 것을 목표로 한다.

상기 목표를 달성하기 위해 다음과 같은 기능을 갖는다.

1. 유저와 게임에 대한 알고리즘을 제출한 후에 다른 유저와 대전하는 기능
2. 유저가 대전 기록의 리플레이를 보는 기능
3. 유저가 자신의 코드와 대전할 수 있는 기능
4. 관리자가 문제를 쉽게 추가할 수 있는 기능

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용

.2.1.1 클라이언트 서버

1. React/Redux를 사용해 개발 진행.
2. 다음과 같은 이유로 디자인 프레임워크로는 Material UI를 채택.
 - 공식 사이트에 잘 정리된 문서
 - 참고 가능한 방대한 양의 데이터 존재
 - React환경에 특화
3. 리플레이 페이지에서는 생동감 있는 화면을 위해 phaser 3 프레임워크를 이용.
4. 클라이언트 수행내용
 - 1) 코드 작성, 제출 및 관리
 - 문제를 선택하고 나면 문제에 대한 pdf 파일을 볼 수 있고 그에 따라 Codemirror 에디터에서 C, C++, Python 언어로 코드를 작성할 수 있다.
 - 각 언어마다 템플릿 코드를 제공해준다.
 - 제출시에 코드의 이름을 설정할 수 있고 코드 이름의 기본값으로 My_Code 를 설정해두었다.
 - 제출에 성공하면 Code-List 메뉴에서 제출한 코드의 상태를 확인할 수 있고 수정할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

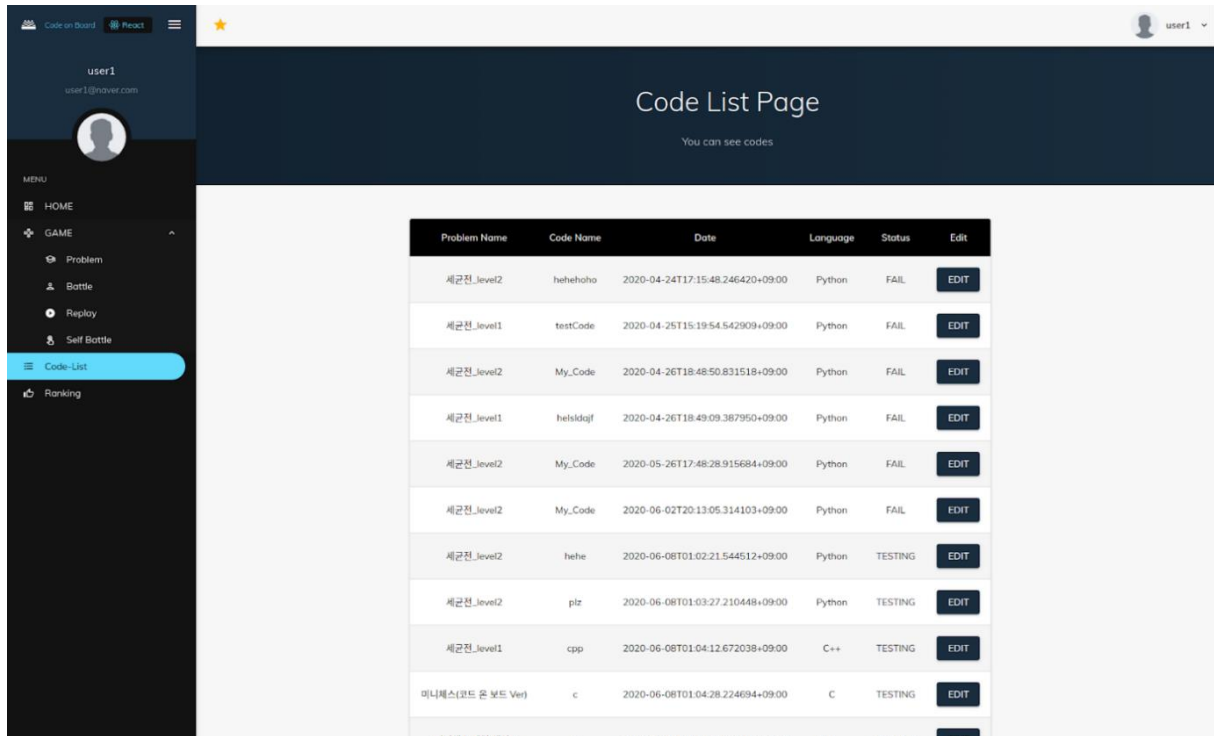


그림 1 코드 제출, 작성 페이지

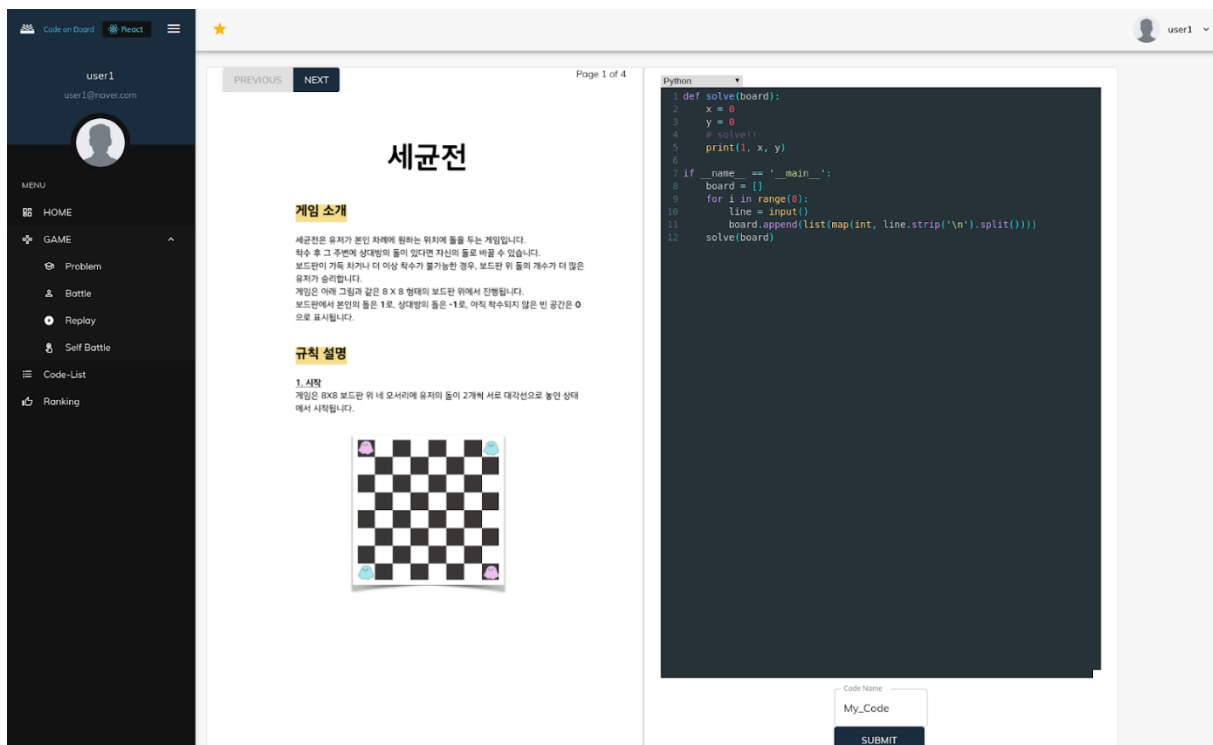


그림 2 코드 관리 페이지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2) 리플레이 기능

- 리플레이 페이지에서 진행했던 게임 목록을 볼 수 있으며 게임에 대한 정보들을 대략적으로 확인할 수 있다. (게임 결과, 매칭 되었던 사용자 정보 등)
- 리플레이 보기 버튼을 클릭하면 modal 창으로 리플레이를 볼 수 있다.
- 리플레이는 기본적으로 자동으로 진행이 되며 버튼을 클릭하여 자동과 수동모드를 변환할 수 있다.
- 자동으로 진행될 경우 0.5 초에 한 수씩 진행이 되는 것을 볼 수 있다.
- 수동으로 진행될 경우 슬라이더 바를 마우스로 조절하여 빠르게 진행 상황을 조절할 수 있다. 또한 previous, next 버튼을 클릭하여 한 수씩 관찰할 수 있다.

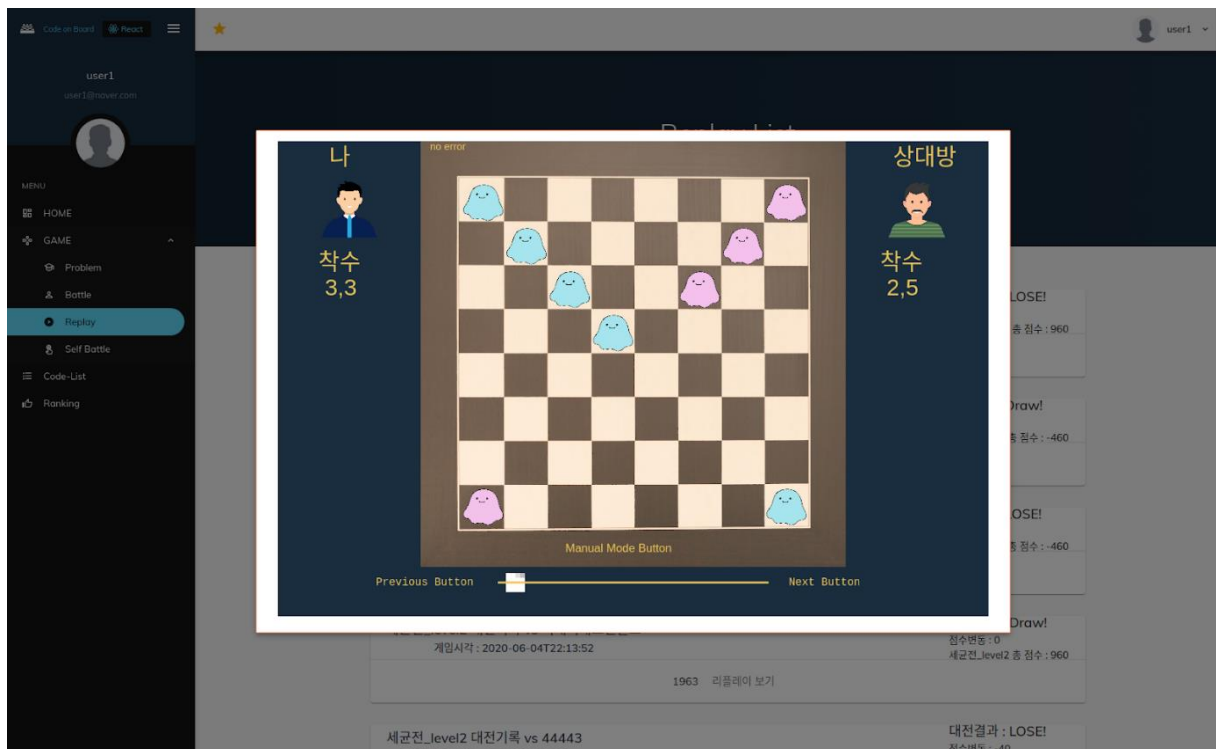


그림 4 리플레이 화면

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

3) 문제 만들기 기능

- 기존에 구현되어 있는 돌에 대한 규칙의 조합으로 새로운 문제를 만들 수 있는 페이지이다.
- 새로운 문제는 관리자 권한이 있는 계정만이 생성할 수 있다.
- 규칙 설정에서 돌의 개수는 최소 1 개에서 최대 4 개까지 설정할 수 있다.
- 게임 규칙은 다음과 같이 각 돌에 대한 규칙과 게임종료 규칙으로 정의된다.
- 착수 규칙 : 돌을 놓거나 이동시킬 때 적용되는 규칙이다.
- 액션 규칙 : 돌을 놓거나 이동시킨 후 발생하는 이벤트에 대한 규칙이다. 예) 이동 시 근접한 돌을 내 돌로 변경/ 이동, 착수 위치의 적 돌을 내 돌로 변경
- 엔딩규칙 : 게임이 종료되는 상황에 대한 규칙이다. 예) 보드판이 꽉 차게 되면 게임 종료
- 각 문제는 이와 같은 Rule 데이터를 Json 형식으로 가지고 있다.

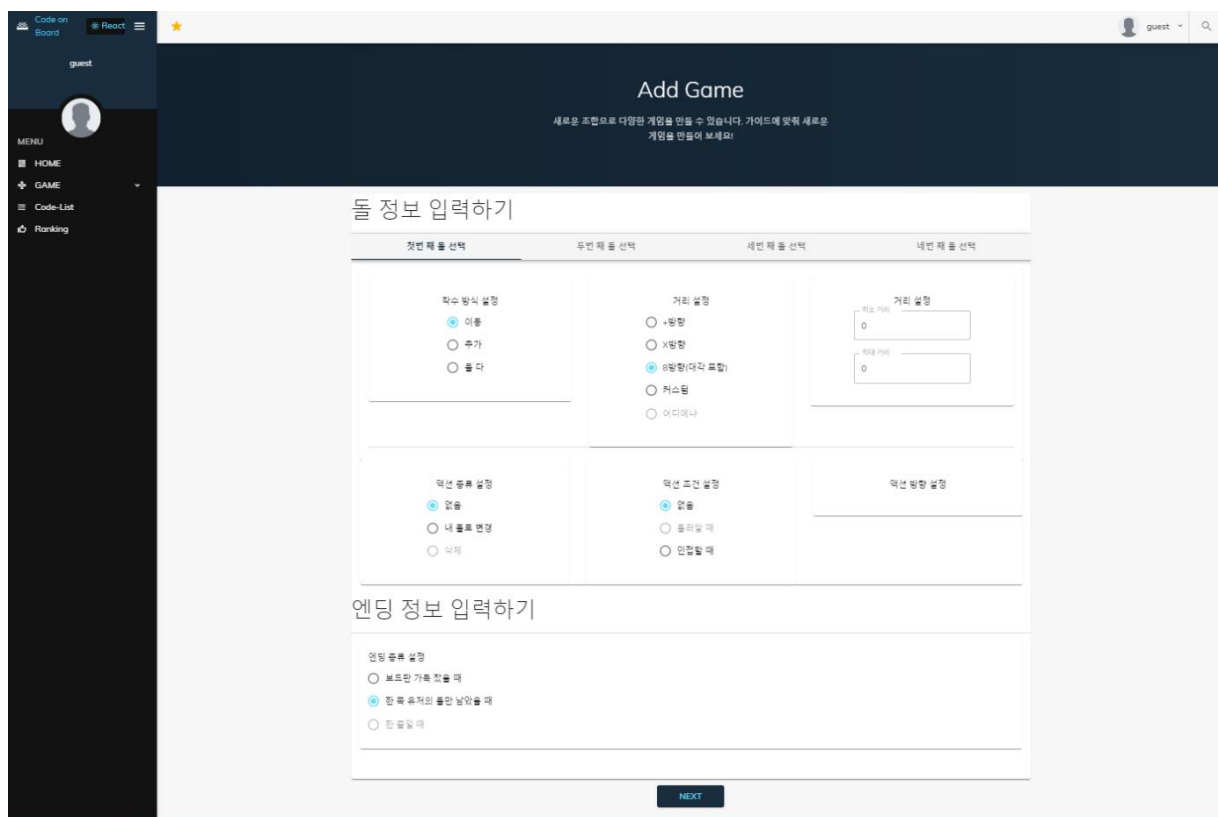


그림 5 문제 만들기 화면 1

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

4) 내 코드와 대전 기능

- 각 문제 별로 제출한 코드들을 선택하여 진행할 수 있다. 이때 코드에 대한 정보는 Code-List 페이지와 동일하게 확인할 수 있다.
- Battle 버튼을 클릭하면 내코드와 대전을 진행한다.
- 마우스 클릭 이벤트로 보드판에서 게임이 진행되며 이전에 선택한 코드와 대전을 진행하며 사용자가 challenger, 코드가 opposite 가 된다.
- 사용자가 게임을 진행하면 API 서버로부터 보드판에 대한 정보를 받아오는데 진행된 게임의 상태에 따라 다르게 렌더링된다.
- 사용자가 게임을 진행하면 자동으로 마지막 보드판을 렌더링한다. 리플레이 기능과 유사하게 슬라이더바와 버튼을 이용하여 이전 보드판의 상황을 볼 수 있다. 하지만 리플레이와는 다르게 전부 수동 모드로 진행이 되며, 이전 보드판에서 게임을 진행하면 그 뒤로는 기존에 받아왔던 보드판 정보를 초기화하고 새로운 보드판의 상황을 관찰할 수 있다.
- 이전 보드판의 상황을 보면서 사용자의 차례가 아닌 경우 보드판의 상황을 변경하려고 하면 사용자의 차례가 아니라는 경고 메시지를 준다.
- 게임이 진행될 때 마다 게임 진행 상태를 modal 창 상단에 보여준다.

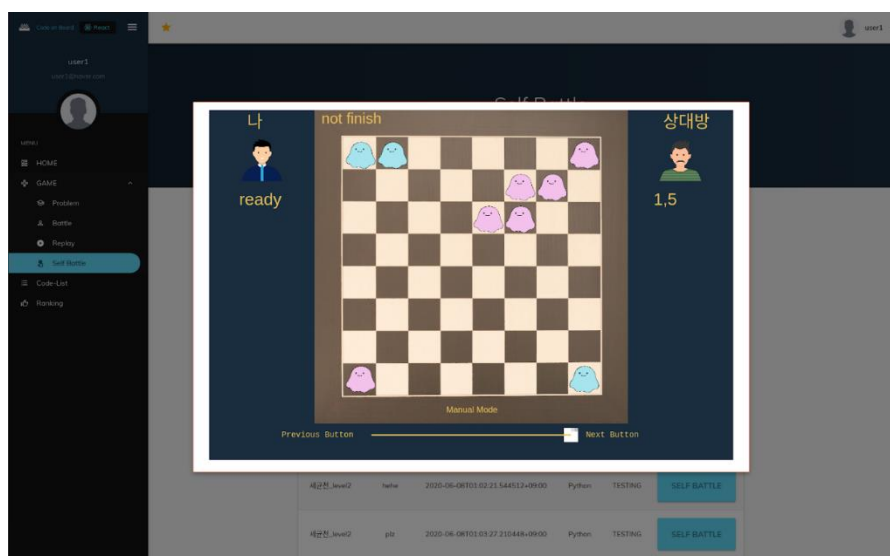


그림 8 내 코드와 대전 페이지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2.1.2 API 서버

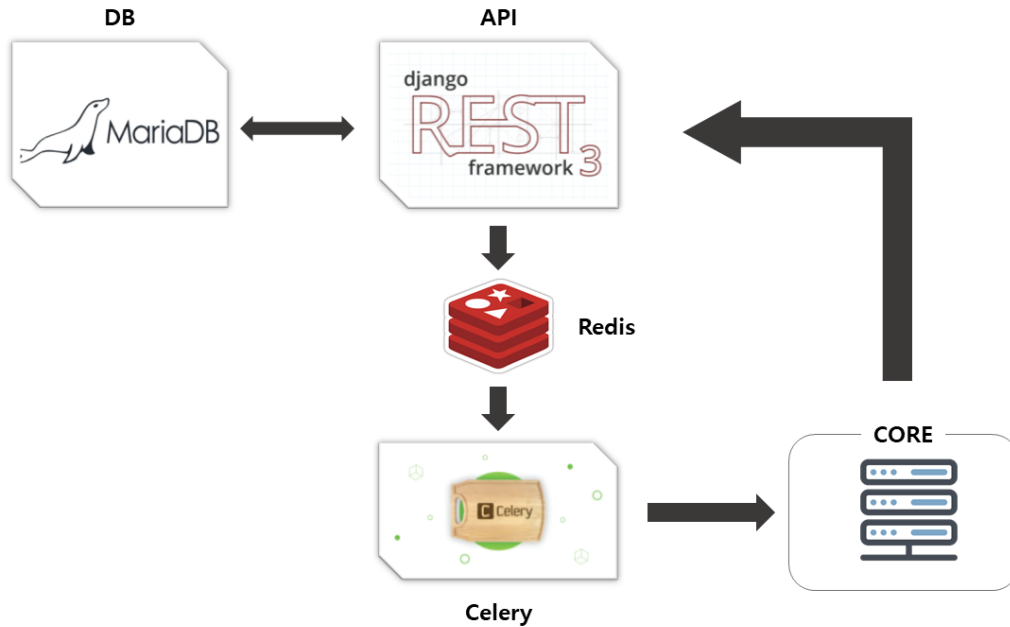


그림 9 API서버와 Core서버 구조

- ✓ Django 프레임워크 사용.
- ✓ Django Rest Framework 모듈 사용
- ✓ Celery를 사용하여 Core서버와 통신
- ✓ Celery의 브로커로 Redis 사용

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

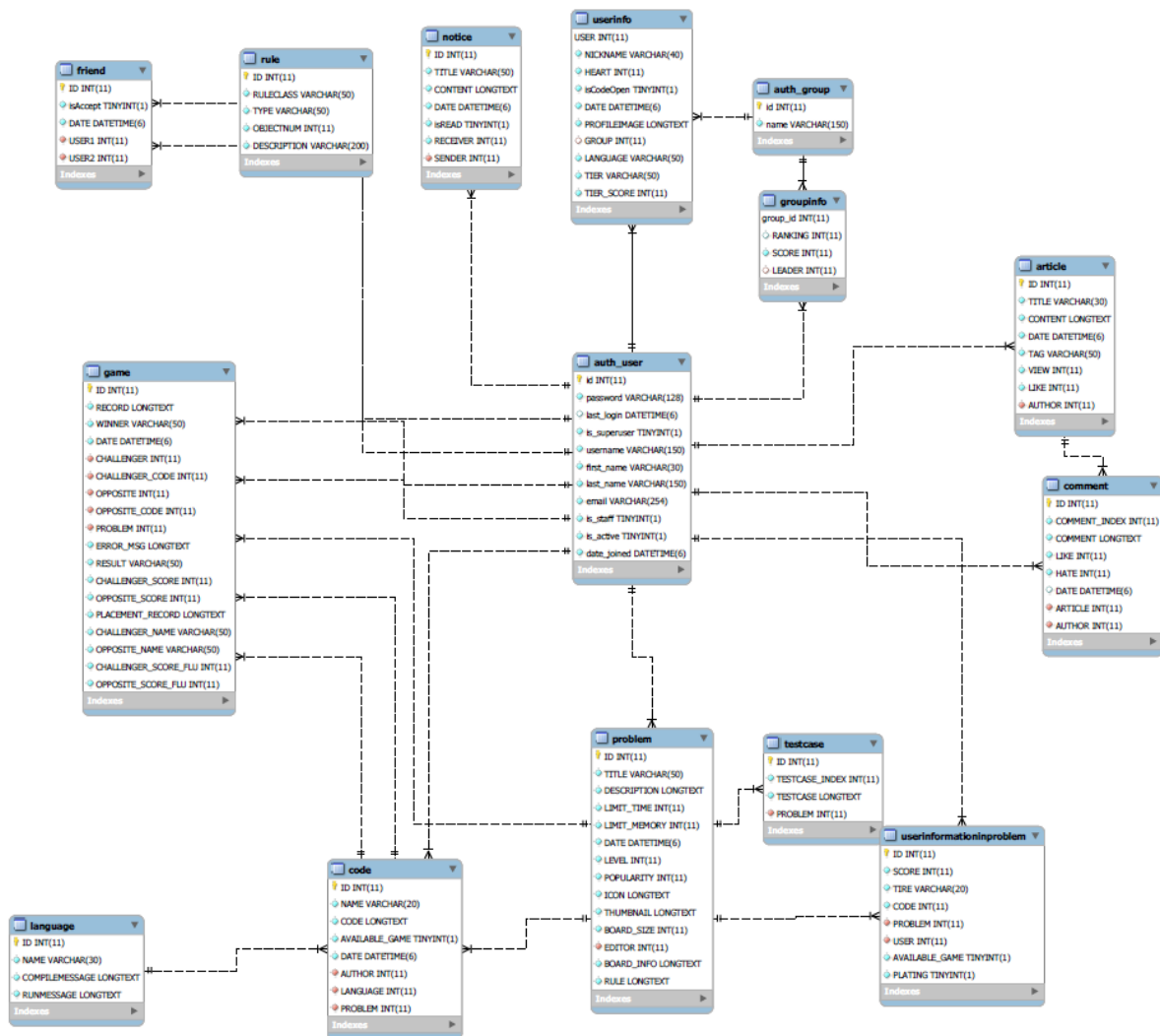


그림 10 Database 구조

서비스에 필요한 Database 모델을 설계하고 그에 맞는 Django ORM 모델을 구현하였다.

RestFul API 서버를 개발하기 위해 Django Rest framework 를 사용하였고 이를 통해 Client 서버와 Core 서버에서 데이터를 읽고 수정할 수 있도록 CRUD API 제공한다.

추가적으로 권한, 인증을 위해 JWT 토큰을 사용하여 Rest-auth 모듈을 사용하였다.

또한 목적에 맞는 별도 API 를 개발하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

1. 매칭 API 개발

대전 기능 이용 시 상대방을 매칭해주는 API 를 개발하였다. Client 에서 매칭 API 를 호출하면 게임 가능한 유저 목록을 불러와 랜덤으로 매칭한 다음, Redis 와 Celery 를 이용하여 Core 서버에 매칭 결과를 전송해준다.

2. Code Test API

매칭 API 와 똑같은 방법으로 유저가 코드를 제출하면 Core 서버에 Celery 를 이용해서 코드 데이터를 보내주어 코드 테스트를 진행할 수 있게 한다.

3. 나와의 대전 API

기존에는 웹 소켓을 이용하여 Client 와 Core 서버간의 실시간 통신을 하려 했지만, 안정성 문제로 인해 API 를 사용하게 되었다. 유저의 착수 정보와 보드판 정보를 Core 서버로 넘겨주고, Core 서버에서의 결과를 redis 로 읽어 Client 에게 Response 해준다.

4. 점수 변동에 따른 등급(티어) 계산

유저가 대전을 하고 나서 점수가 업데이트되면 그에 맞는 티어를 바로 계산하여 적용하도록 개발하였다. 각 문제마다 등급이 존재하고, 한 유저의 등급이 변화되면 다른 유저들의 등급에도 영향이 미치므로 이 또한 다시 계산한다

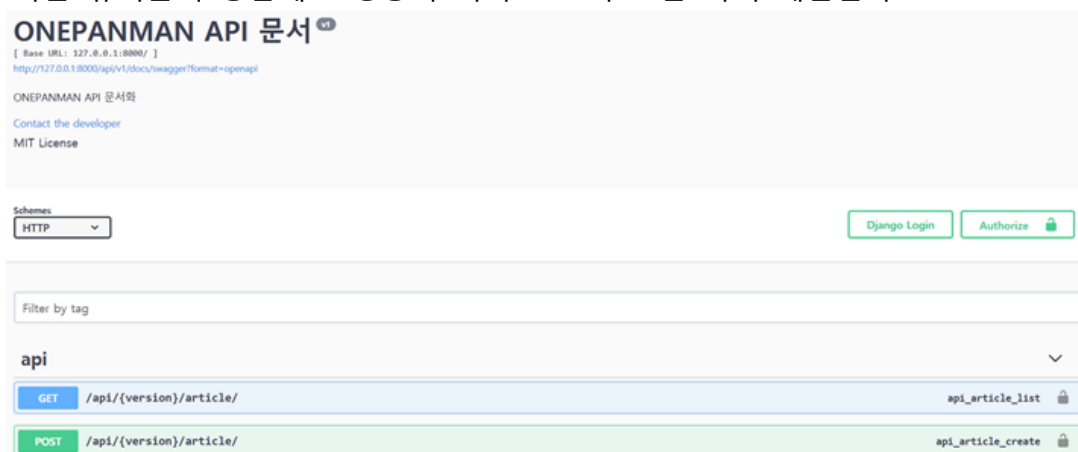


그림 11 API 문서

해당 API를 온라인 문서화하여 관리한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.1.2 Core서버

- ✓ 유저의 코드를 컴파일하고 실행하여 대전을 하기 때문에 시간이 오래 걸릴 수 있다. 이럴 경우 시간이 적게 걸리는 요청들이 실행되지 못해서 웹서비스의 품질이 떨어질 수 있다. 따라서 이 작업을 비동기적으로 처리하여 웹서비스의 품질을 향상하기 위해 분리하였다.

API 서버가 Redis 에 작업을 추가하면 Core 서버는 Celery 를 통해 작업을 전달받아 실행한다.

```

algorithm@algorithm: ~/capstone-2020-16/core
File Edit View Search Terminal Help
algorithm@algorithm:~/capstone-2020-16/core$ celery -A tasks worker --loglevel=info

----- celery@algorithm v4.4.2 (cliffs)
-----
*****
Linux-5.3.0-53-generic-x86_64-with-Ubuntu-18.04-bionic 2020-06-05 18:18:23
**
** [config]
** .> app: tasks:0x7f293a7fdc50
** .> transport: redis://localhost:6379//
** .> results: redis://localhost:6379/
** .> concurrency: 4 (prefork)
** .> task events: OFF (enable -E to monitor tasks in this worker)
*****
[queues]
.> celery exchange=celery(key=celery)

[tasks]
. tasks.play_game
. tasks.play_with_me
. tasks.test_code

[2020-06-05 18:18:23.725: INFO/MainProcess] Connected to redis://localhost:6379//
[2020-06-05 18:18:23.739: INFO/MainProcess] mingle: searching for neighbors
[2020-06-05 18:18:24.767: INFO/MainProcess] mingle: all alone
[2020-06-05 18:18:24.785: INFO/MainProcess] celery@algorithm ready.
[2020-06-05 20:42:46.408: INFO/MainProcess] Received task: tasks.play_with_me[cd597d94-8ecd-4d7e-8d4f-1dbc
c1995179]
[2020-06-05 20:42:46.423: WARNING/ForkPoolWorker-4] run container for play with me
[2020-06-05 20:42:46.445: WARNING/ForkPoolWorker-4] matchdata.json.06-05-20-42-46_2
[2020-06-05 20:42:46.445: WARNING/ForkPoolWorker-4] /home/algorithm/capstone-2020-16/core/play_with_me/mat
chdata.json.06-05-20-42-46_2
[2020-06-05 20:42:46.448: WARNING/ForkPoolWorker-4] {'challenger': 2, 'opposite': 2, 'challenger_code_id':
357, 'opposite_code_id': 357, 'challenger_code': 'MAP_SIZE = 8\nidx = [0, 0, 1, -1, 1, 1, -1, -1]\nndy = [1
, -1, 0, 0, 1, -1, 1, -1]\nnp = [8, 4, 2, 1, 1, 2, 4, 8]\nndef valid(x, y):\n    return x >= 0 and x < MAP
_SIZE and y >= 0 and y < MAP_SIZE\n\nndef solve():\n    pos = []\n    a = [list(map(int, input().split()))
] for i in range(MAP_SIZE):\n        v = [[False for i in range(MAP_SIZE)] for j in range(MAP_SIZE)]\n        for
i in range(MAP_SIZE):\n            for j in range(MAP_SIZE):\n                if a[i][j] != 1: continue\n                nx = i + dx[k]\n                ny = j + dy[k]\n                if not valid(nx, ny) or (a[nx][ny] != 0) or v[nx][ny]: continue\n                pos.append((nx, ny))\n

```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

- ✓ Core 서버에 맞는 Docker image 를 제작하였다.

사용자의 코드를 실행하기 때문에 안정성이 떨어질 수 있다. 따라서 Core 서버의 환경을 보호하기 위해 Docker container 안에서 실제 프로그램을 실행한다.

```
core > Dockerfile
1 FROM ubuntu:18.04
2 MAINTAINER CMS0503 tyms0503@gmail.com
3
4 ENV DEBIAN_FRONTEND noninteractive
5
6 RUN apt-get update
7 RUN apt-get install locales
8 RUN locale-gen ko_KR.UTF-8
9 ENV LANG ko_KR.UTF-8
10 ENV LANGUAGE ko_KR.UTF-8
11 ENV LC_ALL ko_KR.UTF-8
12 RUN update-locale LANG=ko_KR.UTF-8
13
14 RUN apt-get update \
15     && apt-get install -y python3-pip python3-dev \
16     && cd /usr/local/bin \
17     && ln -s /usr/bin/python3 python \
18     && pip3 install --upgrade pip
19
20 RUN apt-get -y install gcc mono-mcs && \
21     rm -rf /var/lib/apt/lists/
22
23 RUN apt-get -y install g++
24
25 RUN pip install requests
26 RUN pip install numpy
27 RUN pip install redis
28
29 COPY . /
```

그림 12 Dockerfile

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

- ✓ 보드게임들을 리스트업 하여 규칙들을 추출하였다. 추출한 규칙들은 착수규칙, 액션규칙, 엔딩규칙으로 나누어서 함수로 구현하였고 조합하여 게임을 만들 수 있도록 하였다.

```
{
  "obj_num" : 게임에 사용될 돌의 개수,
  "placement" : { "1" : [ "착수방식" ,[[착수규칙1,x,y],[착수규칙2,x,y]]] },
  "action" : { "1" : [액션조건,액션방향,액션옵션] },
  "ending" : [엔딩조건]
}
```

그림 13 게임 규칙 조합

- ✓ 대전 진행 로직을 구현하였다. 사용자의 착수를 검사하는 기능, 착수가 되고 문제에 맞는 액션규칙을 실행하는 기능, 문제가 종료되었는지 확인하는 기능으로 나누어서 구현하였다. 자세한 진행 방법은 아래와 같다.

1. Task 와 매칭 정보를 Celery 를 통해 API 로부터 받는다.
매칭 정보에는 아래와 같은 정보가 있다.
 - 사용자 1,2 의 idx
 - 사용자 1,2 의 코드 idx
 - 사용자 1,2 의 코드
 - 문제 idx
 - 대전 idx
 - 문제 규칙
2. Task 를 받으면 Docker Container 를 실행하여 Container 안에서 대전이 진행된다.
3. 대전은 매칭 정보에 있는 사용자의 코드를 번갈아 실행하며 진행한다.
사용자의 코드를 실행할 때 입력은 현재 보드판 상태이고 출력은 착수위치이다.
4. 사용자 코드의 출력인 착수위치가 규칙에 맞는지 확인한다.
5. 올바른 착수라면 문제에 맞는 액션 규칙을 실행하고 보드판을 갱신한다.
6. 엔딩 규칙 검사를 통해 대전이 끝났다면 승자를 정한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

- ✓ 웹서비스에서 사용할 게임을 제작하였다. 게임은 크게 2 종류가 제작되었는데 세균전과 체스이다. 세균전은 돌 추가 룰만 있는 level1 과 이동과 추가 모두 되는 level2 를 만들었고 체스는 난이도를 낮추기 위해 돌의 종류를 3 가지로 줄였고 각 돌의 룰도 간소화하였다.

- ✓ 유저가 제출한 코드가 실행 가능한 코드인지 테스트하는 기능을 구현하였다.

유저가 코드 제출시 유저의 코드끼리 대전하고 오류없이 대전이 끝난다면 제출한 코드를 게임 가능한 코드로 판단하도록 구현하였다. 예를 들어 유저가 코드 A 를 제출했다면 A vs A 를 진행하여 코드 A 에 대해서 테스트를 진행한다.

- ✓ 내 코드와의 대전 기능을 구현하였다.

1. 클라이언트로부터 API 를 통해 유저가 클릭한 곳의 착수위치를 받는다.
2. 올바른 착수라면 액션규칙에 맞는 액션을 수행하고 보드판을 갱신한다.
3. 갱신된 보드판을 유저의 코드의 입력으로 하여 코드를 실행하고 착수와 액션을 진행한다.
4. 유저가 착수했을 때의 보드판 상태와 코드가 착수했을 때의 보드판을 클라이언트에게 보낸다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2.1.4 베타 테스트

4월 23일부터 ~ 5월 8일까지 약 2주간 베타서비스를 진행하였다.

베타서비스에는 약 30명이 참여하였으며 제출된 코드는 약 300개, 게임은 약 1500판이 진행되었다.

1 2 3 4 ... 15 16 1528 게임정보

1 2 3 4 304 코드정보

테스트 기간 중 발생한 에러들을 문서화하였다.

내용	일자
공지 내용을 웹서비스 내에서 팝업화	04월 23일
Problem / Battle 페이지가 헛갈림	04월 23일
Replay 점수 반영이 한게임 느림	04월 24일
매칭에서 코드 선택이 안됨 (코드가 안보임)	04월 24일
한쪽 유저는 돌을 둘 수 없고, 다른 유저는 돌을 둘 수 있을 때의 승패	04월 24일
갑자기 점수가 대폭 하락하는 버그	04월 24일
Matching error 버그	04월 24일
게임 도중 게임이 안끝나는 버그	04월 25일
매칭에서 코드 선택 시 최대 n개 코드 출력	04월 28일
자동 로그인 시, 제대로 로그인이 되지 않아 api 호출 시 에러 발생	04월 28일
로그인 -> 리플레이 -> 리플레이 보기 시 로그아웃됨	05월 01일
연속으로 같은 사람 매칭됨	05월 02일
코드 제출 시 cors error	05월 06일
수를 둘 수 없는 상황에서 게임이 종료되지 않는 버그	05월 07일

그림 14 에러 내용

● 클라이언트

✓ 매칭에서 코드 선택이 안됨 (코드 목록이 안보임)

➔ 매칭 페이지에서 코드 리스트를 불러오는 중, 코드의 개수가 많아져
여러 번 request를 보내야 할 때 에러가 있어 이를 수정하였음.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

● API


- ✓ Replay (대전기록)에서 점수 반영이 한게임 느낌
 - ➔ 대전기록 데이터에서 점수 저장 시 저장 순서에 오류가 있어 이를 수정하였다.
- ✓ Matching error 버그
 - ➔ 게임 불가 상태인 코드를 선택할 수 있어 매칭 에러가 발생. 게임 가능한 코드만

선택할 수 있도록 수정하였다.
- ✓ 연속으로 같은 사람 매칭
 - ➔ random seed 값의 문제가 있는 듯 하여 매칭 시 전판 유저와 매칭된다면 다시 매칭되게 알고리즘을 수정하였다.

● Core

- ✓ 한쪽 유저가 착수를 할 수 없는 상황일 때의 승패 / 수를 둘 수 없는 상황에서 게임이 종료되지 않는 버그
 - ➔ 보드판을 검사하여 한쪽 유저가 착수를 둘 수 없는 상황이면 게임을 종료하고 엔딩규칙에 맞게 승패를 결정하였다.
- ✓ 갑자기 점수가 대폭 하락하는 버그
 - ➔ Core서버 로컬에서 코드를 디버깅 하는 중에 점수까지 업데이트하여 일어난 버그이다. 업데이트 하는 부분을 삭제하고 디버깅을 진행했다.

테스트 기간 중 기능적인 부분에서의 피드백을 문서화하여 정리하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

내용	일자
공지 내용을 웹서비스 내에서 팝업화	04월 23일
Problem / Battle 페이지가 핫갈림	04월 23일
코드 임시저장 / 코드 불러오기 기능	04월 24일
매칭에서 코드 선택 시 최대 n개 코드 출력	04월 28일
대전 기록에서 사용한 코드 확인 기능	04월 29일
원하는 상대와 배틀하는 기능	04월 29일
배틀 후 다시 매칭 페이지로 리다이렉트	04월 29일
대전 기록에서 점수 변동을 그래프로 표시	04월 29일
리플레이 기록을 공유하는 기능	04월 29일
리플레이를 문제별, 대전결과별로 나누어 보는 기능	04월 29일
대전 후 점수 변동 폭의 다양화	04월 30일
로그아웃 되면 로그인 화면으로 리다이렉트해주는 기능	05월 01일
리플레이 보기에서 방향키를 이용해서 이동하는 기능	05월 02일

그림 15 피드백 내용

테스트 중에 동일한 코드끼리의 대전이 유의미하지 않고, 단순히 점수를 올리는 것으로 판단되어 점수를 측정하는 방법을 변경하기로 하였다.

1. 유저가 매칭을 하여 대전할 수 있는 기능은 친구와의 대전 상위유저와의 대전같은 친선전을 위해 남겨둔다.
2. 한 게임 내에서 코드의 순위는 특정 시간에 전수조사 하여 순위를 결정한다.

1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	4100	-2040	2020년 5월 8일 10:50 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	4080	-2020	2020년 5월 8일 10:49 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	4060	-2000	2020년 5월 8일 10:48 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	4040	-1980	2020년 5월 8일 10:48 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	3960	-1960	2020년 5월 8일 10:46 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	3940	-1940	2020년 5월 8일 10:46 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	3900	-1920	2020년 5월 8일 10:45 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je
1_세균전_level2_admin	허태하테오션월드	44443	challenger	finish	3860	-1900	2020년 5월 8일 10:44 오후	650_허태하테오션월드_세균전_level2_PYTHON	340_44443_세균전_je

그림 16 대전 상대가 동일한 매칭기록

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2.2 시스템 기능 및 구조 설계도

2.2.2.1 시스템 구조도

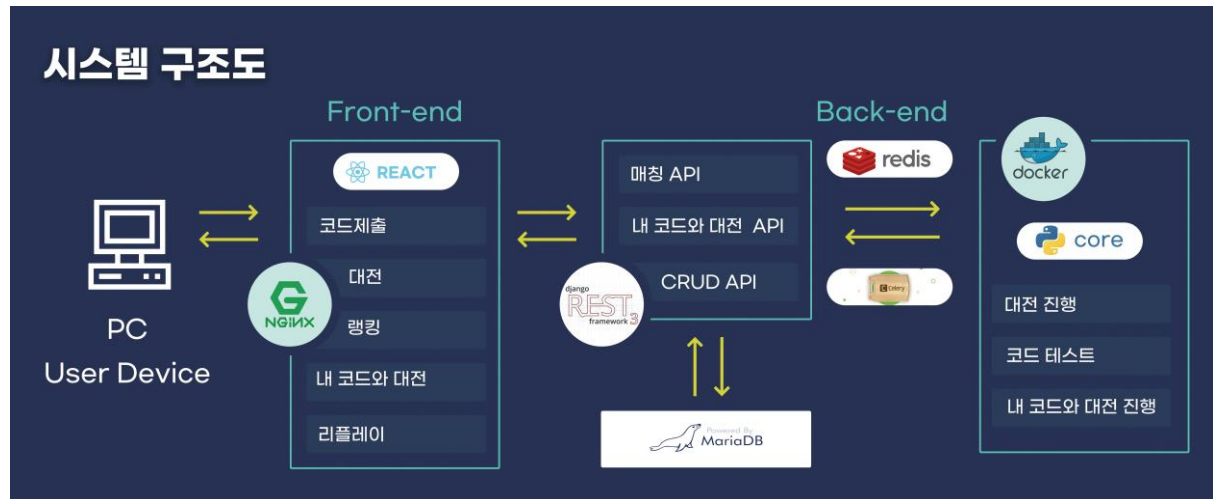
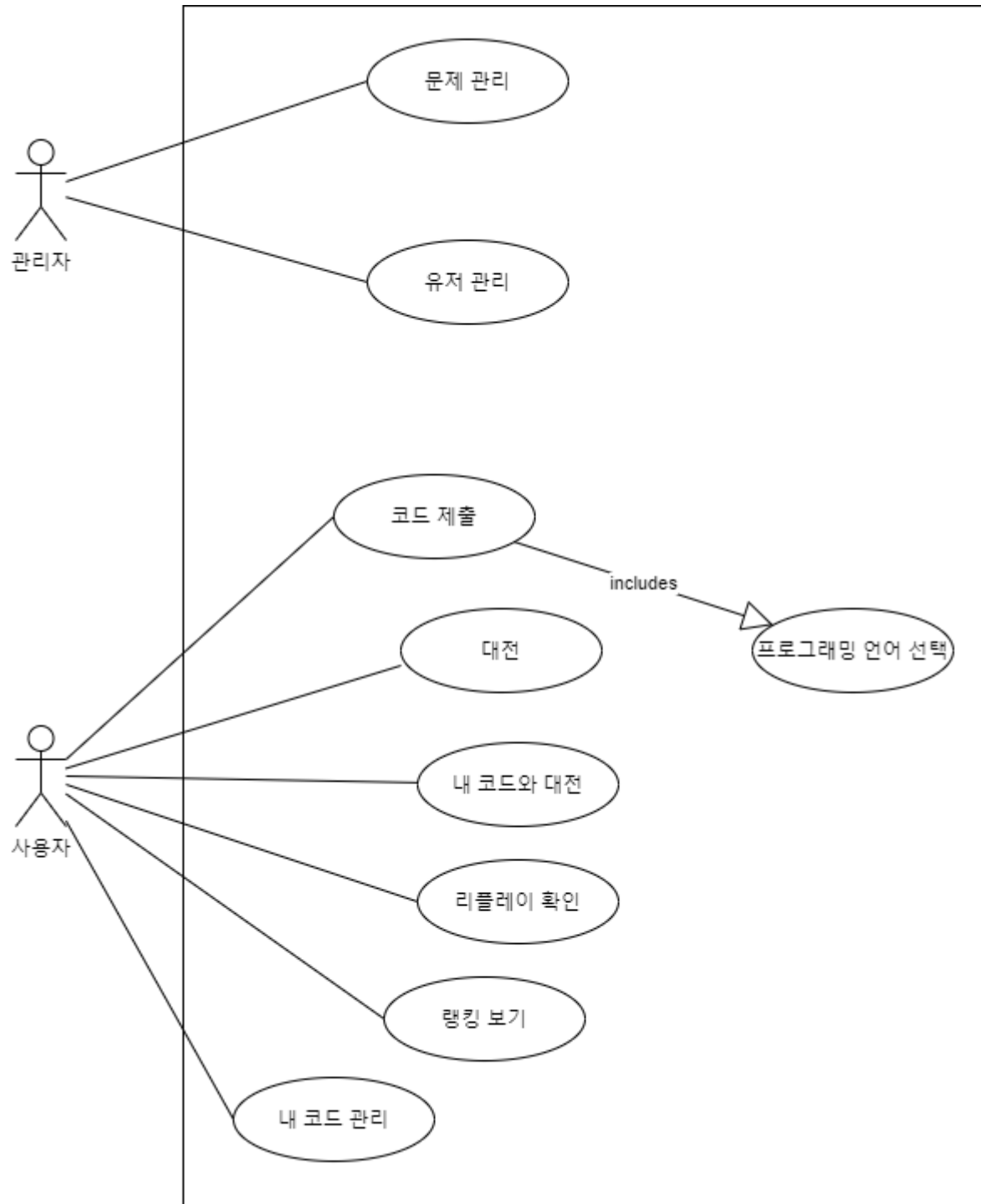



그림 17 시스템 구조도

유저가 요청을 보내면 API 를 통해 Core 서버로 요청을 보내 비동기적으로 처리한다.

2.2.2.2 Usecase Diagram



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2.2.3 기능적 요구사항

항목	내용	완료여부	기타
회원가입 하기	사용자가 웹사이트에 회원가입 할 수 있다.	○	
로그인 하기	사용자가 웹사이트에 로그인 할 수 있다	○	
문제 확인하기	문제리스트를 확인 할 수 있다	○	
대전하기	다른 사용자와 대전 할 수 있다	○	
랭킹 확인하기	문제별로 랭킹을 확인할 수 있다	○	
내 코드와 대전하기	내 코드와 대전할 수 있다.	○	
내 정보 확인하기	내 정보를 확인할 수 있다	X	
리플레이 보기	대전 결과를 확인하고 리플레이를 볼 수 있다.	○	
대전하기	두 사용자의 코드를 대전 시킨다.	○	
코드 테스트 하기	사용자가 제출한 코드가 유효한 코드인지 검사한다.	○	
코드 관리 하기	사용자가 자신의 코드를 확인하고 수정할 수 있다.	○	
문제 만들기	기존 규칙을 조합하여 새로운 문제를 만들 수 있다.	○	

2.2.2.4 비기능적 요구사항

항목	내용	완료여부	기타
보안성	등록된 사용자만이 시스템에 접근할 수 있어야 한다.	○	
확장성	클라이언트, API, CORE 서버를 분리하여 개발한다.	○	
품질	동시 접속자 수 1000명일 때, 초당 10000트랜잭션을 처리할 수 있어야 한다.	○	
사용성	사용자들이 쉽게 사용 가능하도록 간단한 UI를 사용하고 도움말을 제공한다.	○	
정확성	게임이 규칙과 코드에 맞게 진행된다.	○	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2.3 활용/개발된 기술

- ✓ **Docker** : 컨테이너 기반의 오픈소스 가상화 플랫폼이다. 실행환경을 컨테이너로 추상화하고 동일한 인터페이스를 제공하여 프로그램의 배포 및 관리를 단순하게 해준다.
- ✓ **Celery** : 비동기 작업 처리를 위한 파이썬 프레임워크이다.
- ✓ **Redis** : 메모리 기반의 "키-값" 구조 데이터 관리 시스템이며, 모든 데이터를 메모리에 저장하고 조회하기에 빠른 Read, Write 속도를 보장하는 비 관계형 데이터베이스이다. 본 프로젝트에서는 Celery 의 브로커로써 사용했다.
- ✓ **React-Redux** : 자바스크립트 라이브러리의 하나로서 사용자 인터페이스를 만들기 위해 사용되었다.
- ✓ **phaser 3** : HTML 5 게임 프레임워크로서 문제 상황판을 구성하는데 사용되었다.
- ✓ **Nginx** : 웹 서버 소프트웨어로, 클라이언트 서버를 배포하는데 사용되었다.
- ✓ **Material-UI** : 리액트 UI 프레임워크로서 각종 레이아웃과 버튼 등을 만들어 놓은 것이다.
- ✓ **Django Rest Framework** : python 라이브러리로서 web API 를 구현하는 데에 사용된다. 본 프로젝트에서 API 서버를 구현하는데 사용된다.

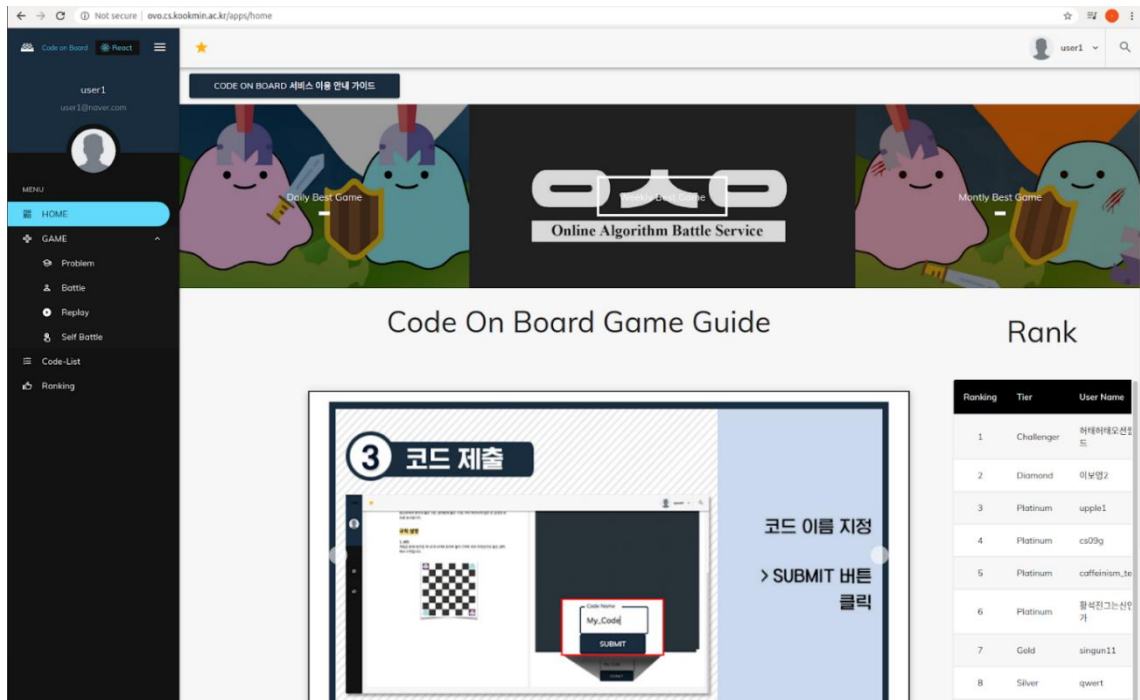
2.2.4 현실적 제한 요소 및 그 해결 방안

한정된 서버 자원 : 클라우드 플랫폼에 서버를 구축하여 자동으로 스케일링 가능하도록 해결할 수 있다.

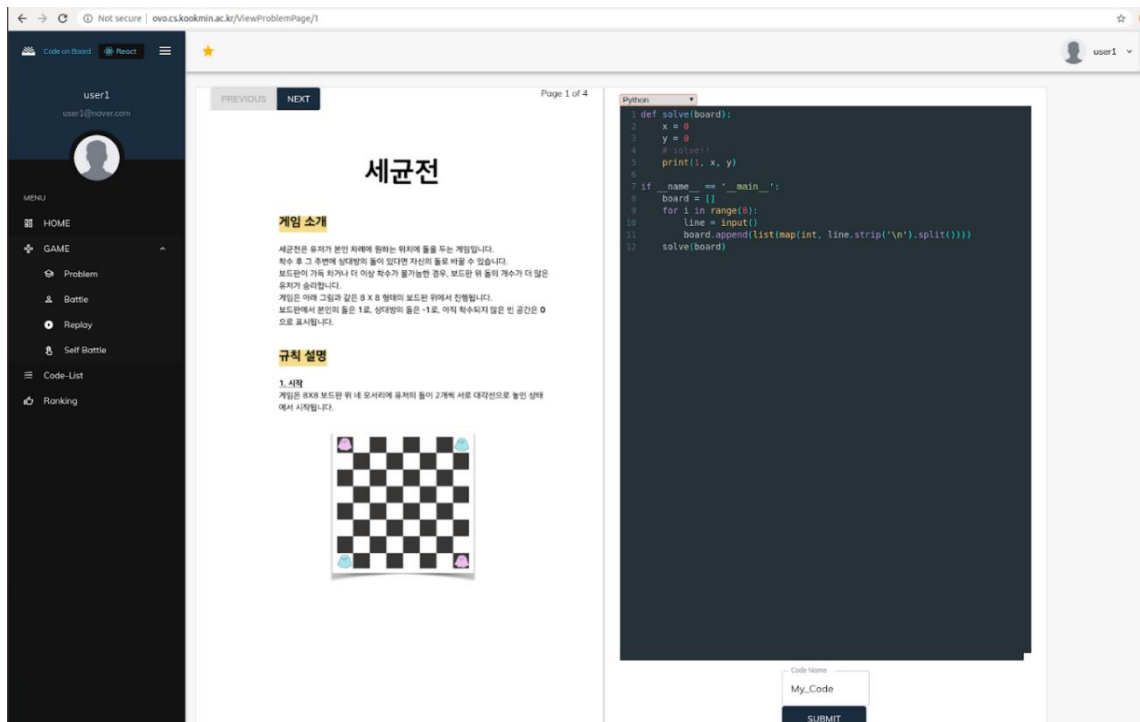
 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.2.5 결과물 목록

● 메인 페이지

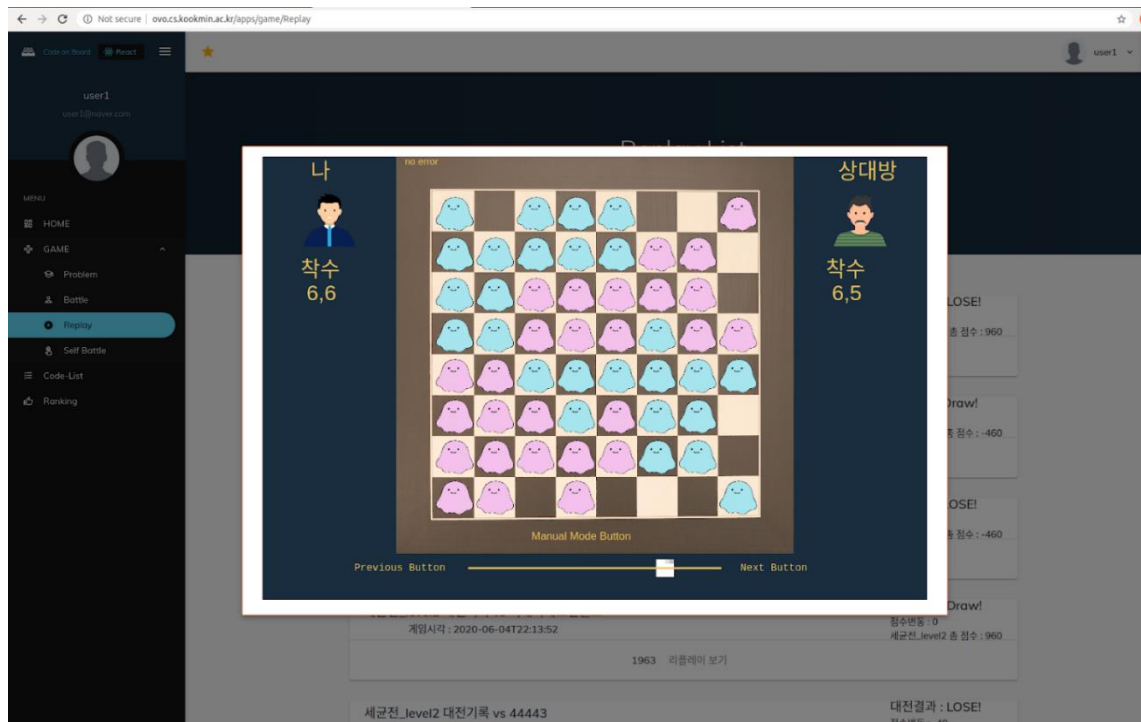


● 사용자 문제 제출 페이지

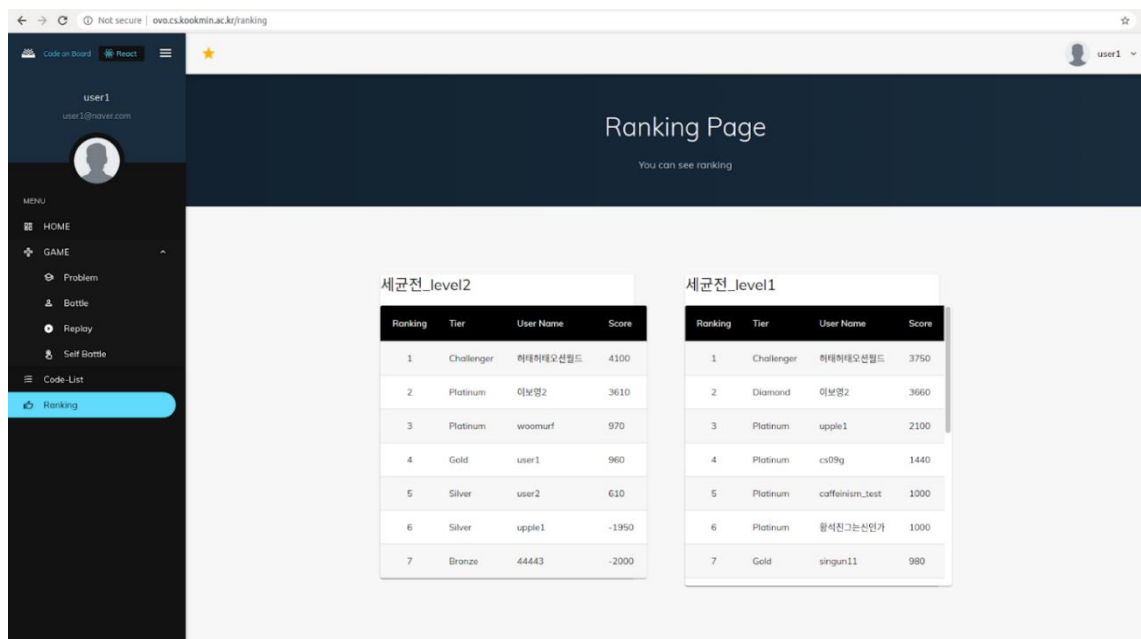


 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

● 리플레이 화면

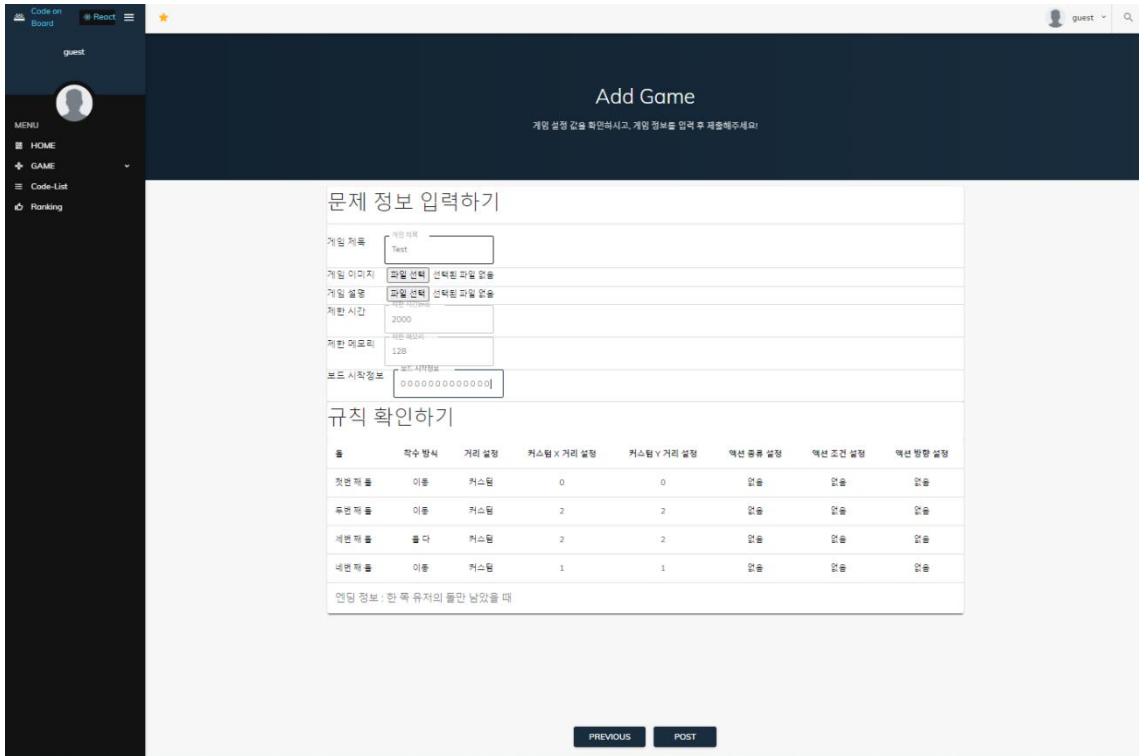


● 랭킹페이지



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

● 문제 만들기 페이지




● 서비스에 사용되는 API 를 제공하는 API 서버

```

algorithm@algorithm: ~/capstone-2020-16.backup/api
File Edit View Search Terminal Help
[09/Jun/2020 14:53:38] "GET /api/v1/rank/?problem=2 HTTP/1.1" 200 1112
Not Found: /
[09/Jun/2020 15:17:21] "GET / HTTP/1.1" 404 2542
[09/Jun/2020 15:31:07] "OPTIONS /api/v1/rank/?problem=2 HTTP/1.1" 200 0
OrderedDict([('id', 63), ('user', 68), ('problem', 2), ('score', 3750), ('tier', 'Challenger'), ('code', 653), ('available_game', False), ('playing', False)])
[09/Jun/2020 15:31:07] "GET /api/v1/rank/?problem=2 HTTP/1.1" 200 1112
[09/Jun/2020 15:31:10] "OPTIONS /api/v1/problem/ HTTP/1.1" 200 0
[09/Jun/2020 15:31:10] "GET /api/v1/problem/ HTTP/1.1" 200 2709
[09/Jun/2020 15:31:11] "OPTIONS /api/v1/game/my HTTP/1.1" 200 0
[09/Jun/2020 15:31:11] "GET /api/v1/game/my HTTP/1.1" 200 305544
[09/Jun/2020 15:31:13] "OPTIONS /api/v1/game/1967/ HTTP/1.1" 200 0
authenticated?: True, user: 2
[09/Jun/2020 15:31:14] "GET /api/v1/game/1967/ HTTP/1.1" 200 10049
[09/Jun/2020 16:04:34] "GET /admin/onepanman_api/code/ HTTP/1.1" 200 66904
[09/Jun/2020 16:04:34] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6170
[09/Jun/2020 16:04:34] "GET /static/admin/css/base.css HTTP/1.1" 200 16378
[09/Jun/2020 16:04:34] "GET /admin/jsi18n/ HTTP/1.1" 200 8330
[09/Jun/2020 16:04:34] "GET /static/admin/css/fonts.css HTTP/1.1" 304 0
[09/Jun/2020 16:04:34] "GET /static/admin/css/responsive.css HTTP/1.1" 200 17944
[09/Jun/2020 16:04:34] "GET /static/admin/js/jquery.init.js HTTP/1.1" 304 0
[09/Jun/2020 16:04:34] "GET /static/admin/js/core.js HTTP/1.1" 200 7135
[09/Jun/2020 16:04:34] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 271751
[09/Jun/2020 16:04:34] "GET /static/admin/js/actions.js HTTP/1.1" 200 6538
[09/Jun/2020 16:04:34] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 6918
[09/Jun/2020 16:04:34] "GET /static/admin/js/urlify.js HTTP/1.1" 200 8972
[09/Jun/2020 16:04:34] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1530
[09/Jun/2020 16:04:34] "GET /static/admin/img/icon-no.svg HTTP/1.1" 304 0
[09/Jun/2020 16:04:34] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 128820
[09/Jun/2020 16:04:34] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 304 0
[09/Jun/2020 16:04:34] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 304 0
[09/Jun/2020 16:04:34] "GET /static/admin/img/sorting-icons.svg HTTP/1.1" 200 1097
[09/Jun/2020 16:04:34] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
[09/Jun/2020 16:04:34] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
[09/Jun/2020 16:04:34] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184

```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2.3 기대효과 및 활용방안

본 서비스를 통해 다음과 같은 기대효과를 예상한다.

1. 알고리즘 트레이닝 동기 제공

문제와 대전이라는 주제를 통해 경쟁심을 자극함으로써 알고리즘 트레이닝을 지속할 수 있는 동기를 제공한다.

2. 자연스러운 코드 리뷰 습관 함양

사용자들은 대전을 진행하고 승패 여부와 획득한 점수에 따라 등급이 매겨진다. 이때 육안으로 보여지는 대전 결과가 사용자의 경쟁심을 자극하고 승리를 위해 코드를 수정하도록 유도한다. 베타테스트 이후 시행한 인터뷰에 따르면 대부분의 사용자가 대전에서 패한 경우 본인 코드의 부족한 부분을 찾기 위해 코드 리뷰를 했다고 응답했다. 또한 코드 리뷰의 편의성을 위해 리플레이 기능과 내 코드와의 대전 기능을 활용할 수 있다.

본 서비스의 다음과 같은 활용방안을 제시한다.

1) 불특정 사용자에게 서비스를 제공할 수 있다.

서비스를 공개적으로 제공한다. 순위는 사용자가 직접 대전을 하여 매긴다.

2) 특정 그룹별로 서비스를 제공할 수 있다.

서비스를 특정그룹에게 제공한다. 순위는 사이트에서 전수조사 하여 매긴다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

3 자기평가


본 프로젝트는 기존에 존재하는 알고리즘 트레이닝 서비스와는 다르게 게임이라는 장르를 통해 알고리즘 트레이닝에 더 흥미를 느끼게 하고 더 높은 순위를 위해 자신의 코드를 수정/보완하는 상황을 제공하기 위해 진행되었다는 점에서 차별화되었다고 볼 수 있다. 이를 위해 기존에 존재하는 백준, 프로그래머스 등 여러 플랫폼을 조사하여 개선할 점을 찾아 초기 기획을 하였다.

첫 기획단계에서 목표를 대전 기능, 리플레이 기능, 내 코드와 대전 기능 그리고 문제 생성 기능 4가지 기능을 개발하는 것으로 정하였다. 결과적으로 4가지 기능 모두 개발 완료하였기 때문에 완성도 높은 프로젝트라고 자체 평가할 수 있다.

대전 기능은 유저가 대전 요청을 보내면 유저 정보를 API 서버로 보내고 API 서버는 매칭 후에 받은 유저정보와 매칭정보를 취합하여 Core 서버로 보내 대전을 진행하는 방식이다. Core 서버에서는 매칭 정보를 받아 Docker container 안에서 유저의 코드를 번갈아 실행하며 대전을 진행한다. 대전이 완료되었다면 Core 서버는 API 서버를 통해 DB 에 대전결과를 저장하고 API 서버는 대전 결과를 클라이언트 서버에게 보내준다. 대전 기능은 시간이 오래 걸리는 작업이기 때문에 Block 현상을 일으킬 수 있다. 따라서 Celery 와 Redis 를 통해 비동기적으로 처리할 수 있도록 구현하였다. 또한 유저의 코드를 실행하기 때문에 서버 환경의 안정성이 떨어질 수 있다. 따라서 Docker 를 사용하여 실제 프로그램 실행 환경을 분리하였다.

리플레이 기능은 API 서버를 통해 대전 결과를 받아와서 시각적으로 보여준다는 점에서 타 알고리즘 사이트들과 차별화 되었다고 볼 수 있다. 문제마다 리플레이에 필요한 정보(둘 이미지 등)를 가지고 있고, 이 정보들을 이용하여 리플레이를 진행하기 때문에 새로운 문제가 추가되더라도 리플레이 기능에 관한 코드를 수정하지 않아도 된다. 즉, 코드의 재사용성이 높아서 새로운 문제를 추가하는데 따르는 오버헤드가 없다. 대전 결과를 로그형식이 아닌 시각적으로 제공하고 수동 모드에서는 슬라이더바를 이용해 보드판의 상황을 빠르게 탐색할 수 있어서 사용자의 편의를 증대시켰다.

내 코드와의 대전은 실시간으로 자신의 코드와 대전이 가능하도록 구현하였다. 사용자가 착수및 이동을 하면 해당 정보를 API 서버를 통해 Core 서버로 보내서 코드와 대전하는 방식이다. Core 서버로부터 응답을 받을 때 까지는 스피너를 렌더링하여

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

진행상황을 명확히 하도록 구현하였다. 굳이 다른 사용자의 코드와 대전을 하지 않아도 자신의 코드 리뷰를 시각적 도움을 받아서 할 수 있다. 베타 테스터들에게 인터뷰한 결과 해당 기능에 대한 긍정적인 답변이 많은 것을 근거로, 기능의 수요가 많을 것이라 예상한다.

문제생성 기능은 관리자가 UI로 비교적 쉽게 문제를 생성할 수 있는 기능이다. 기존에 리스트업 한 규칙들을 나열하고 관리자가 규칙들을 선택하고 조합하여 문제를 만들 수 있도록 구현하였다. 규칙들을 재조합함으로써 새로운 문제를 창작해야 하는 어려움을 해결할 수 있다. 이렇게 조합된 규칙들을 json 형식 변환하여 API 서버를 통해 DB에 저장한다.

사용자들을 위해 UI를 직관적으로 만들고 사용자 가이드를 제공하여 서비스 이용을 쉽게했다.

클라이언트 서버, API 서버 그리고 게임 진행을 하는 Core 서버로 구성되어있기 때문에 웹 서비스를 안정적으로 제공할 수 있다. 또한 서버를 분리해놓아서 서버의 확장성이 높다.

웹 서비스를 제공하는 프로젝트이기 때문에 베타 테스트를 진행하였고 이를 통해 서비스 오픈후에 사용자들의 반응을 유추할 수 있었다. 베타 테스터들의 피드백을 적극 활용하여 서비스와 기능적인 면에서 많은 개선을 이루었고 프로젝트의 방향을 조정할 수 있었다. 베타 테스트를 진행하였기 때문에 프로젝트를 완성도 높게 마무리할 수 있었다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

4 참고문헌

번호	종류	제목	출처	발행년도	저자	기타
1	웹페이지	Coding game	www.codinggame.com			
2	웹페이지	우리는 알고리즘의 주인이 아니다	http://www.snunews.com/news/articleView.html?idxno=20981			
3	웹페이지	알고리즘 사이트 비교 및 추천	https://readystory.tistory.com/120			
4	웹페이지	Empire of Code	https://checkio.org/			
5	웹페이지	Front-End 프레임워크 선택하기	https://medium.com/aha-official/%EC%95%84%ED%95%98-%ED%94%84%EB%A1%A0-%ED%8A%B8-%EA%B0%9C%EB%B0%9C-%EA%B8%B0-0-angular-react-vue-js-%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC-%EC%84%A0-%ED%83%9D-f797392118d0			
6	웹페이지	Material-Ui Docs	https://material-ui.com/			
7	웹페이지	Django Rest Framework Document	https://www.django-rest-framework.org			

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

8	웹 페이지	Django Rest Framework Github	https://github.com/enc ode/django-rest-			
9	웹 페이지	Django rest-auth Document	https://ko.reactjs.org/d ocs/			

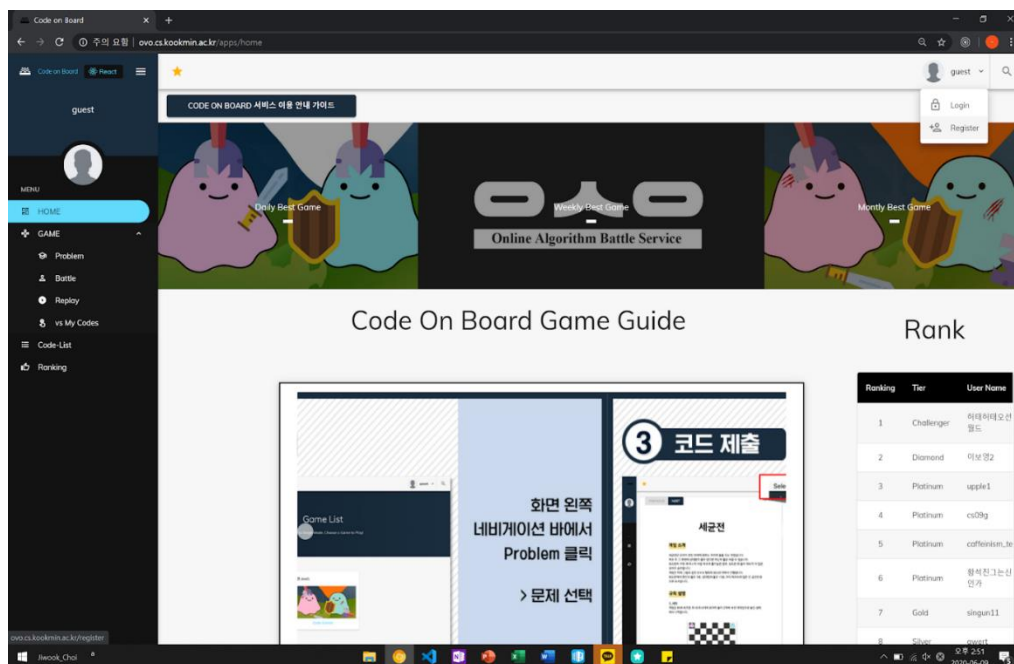
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

5 부록

5.1 사용자 매뉴얼

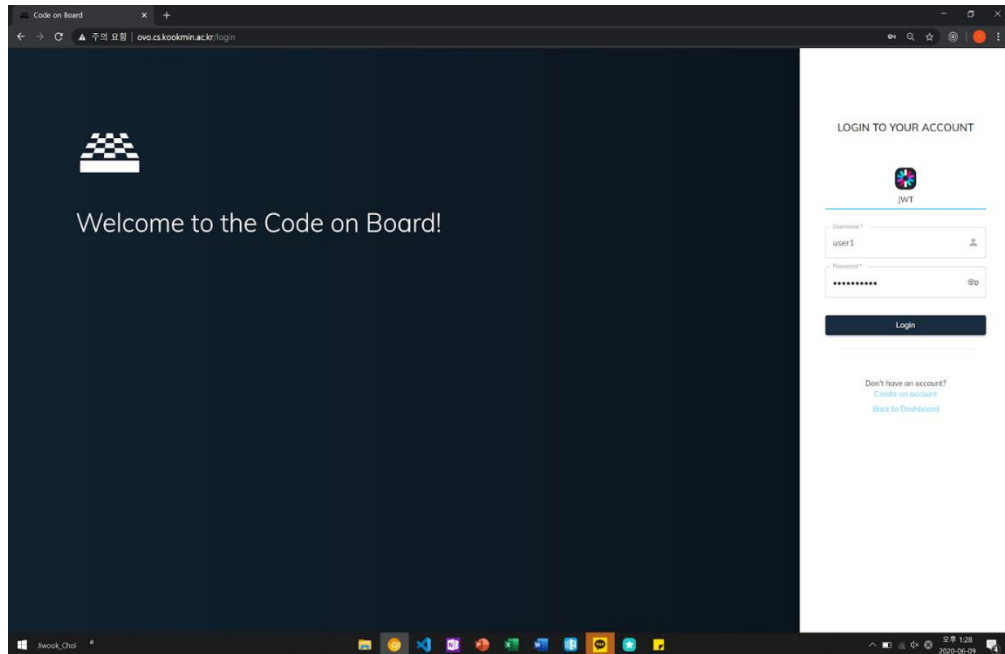
5.1.1 다른 사용자와 대전

1. 상단의 guest 프로필을 선택하여 Register 메뉴를 클릭하여 회원가입을 진행한다.

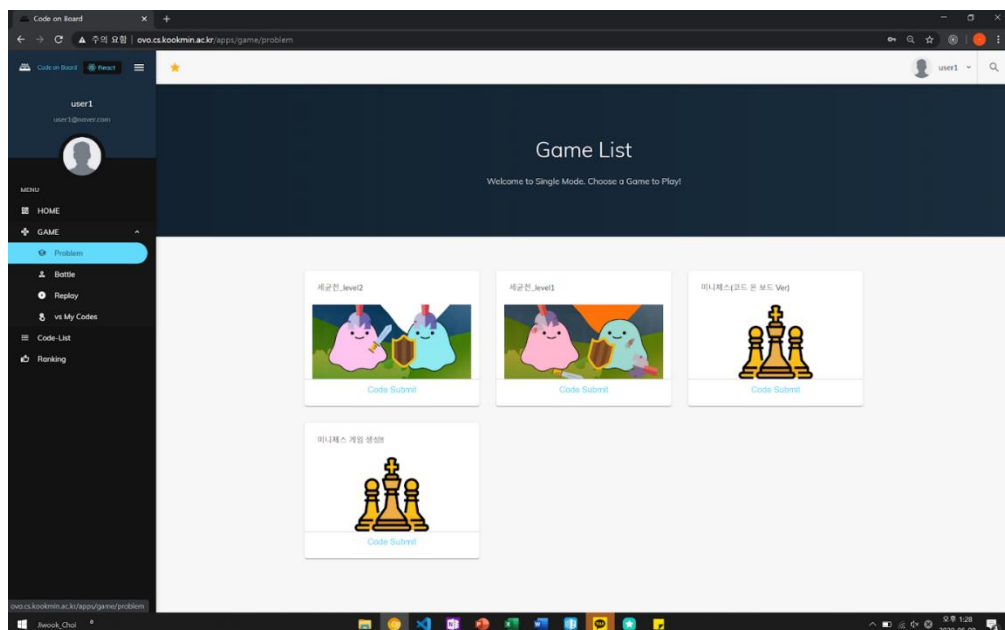


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

2. 회원가입 진행 후에 로그인한다.

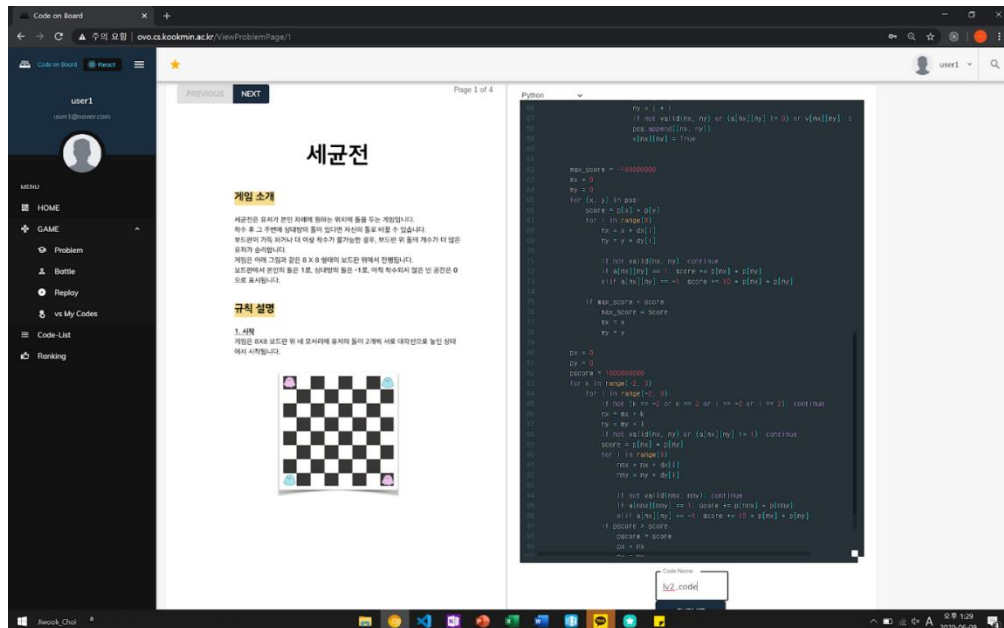


3. 좌측의 GAME 메뉴의 Problem 을 선택한 뒤 풀고자 하는 문제를 선택한다.

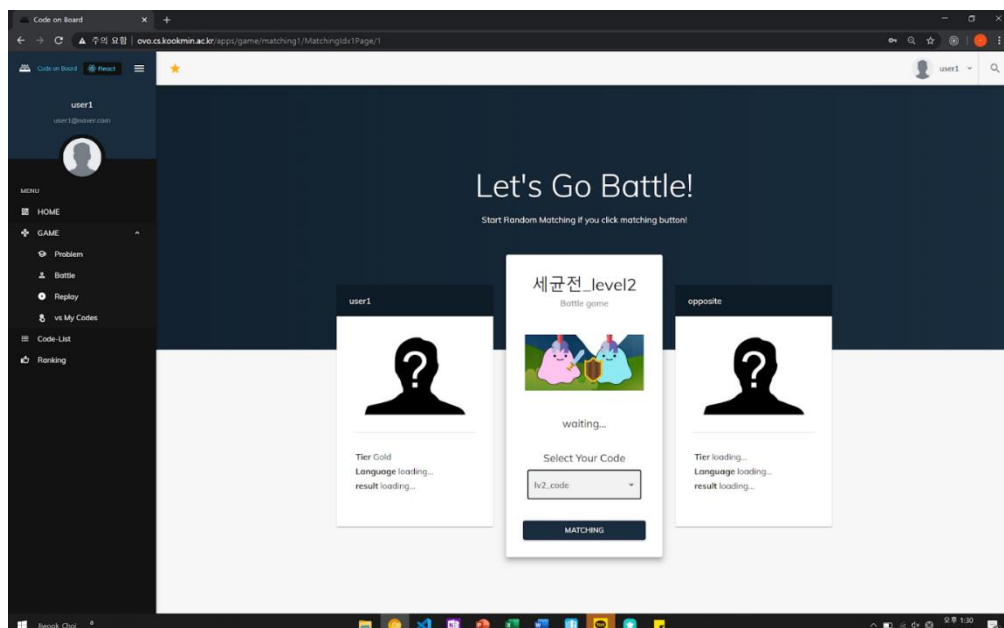



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

- 문제의 설명을 보고 프로그래밍 언어를 선택한 뒤 코드를 작성한다. 코드를 작성이 끝나면 하단에서 코드 이름을 설정하고 SUBMIT 버튼을 클릭하여 제출한다. 이 때 코드가 정상적인 대전이 가능한지 테스트하고 가능하다면 대전 가능 코드가 되고 불가능하다면 대전 불가능 코드가 된다.

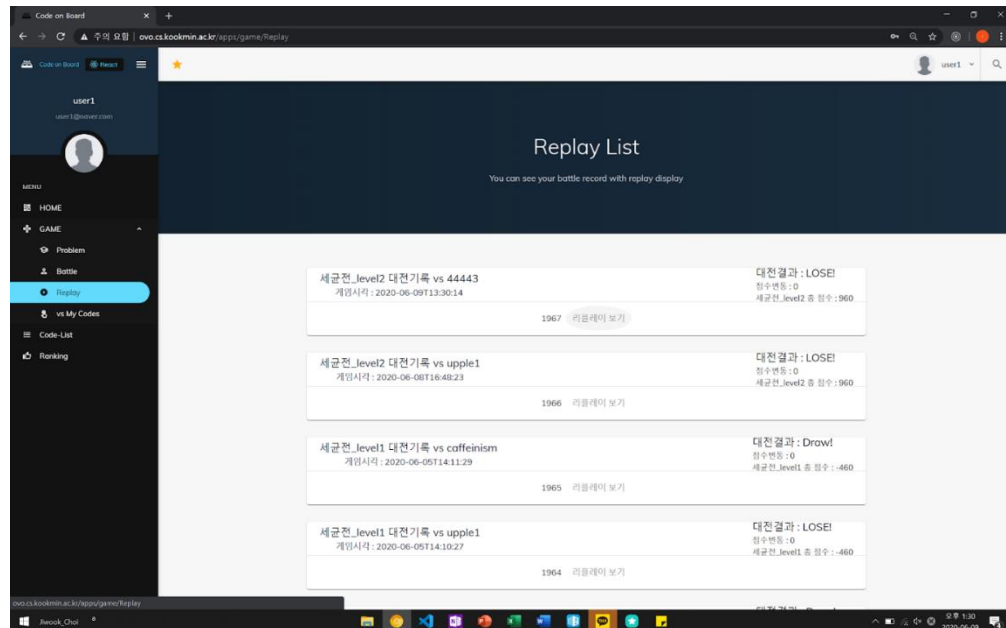


- 대전 가능한 코드를 제출하고 좌측의 GAME 메뉴의 Battle 을 선택하여 코드를 제출했던 문제를 선택한다. 대전하고자 하는 코드를 선택한 뒤 MATCHINBG 버튼을 클릭하여 대전을 진행한다.



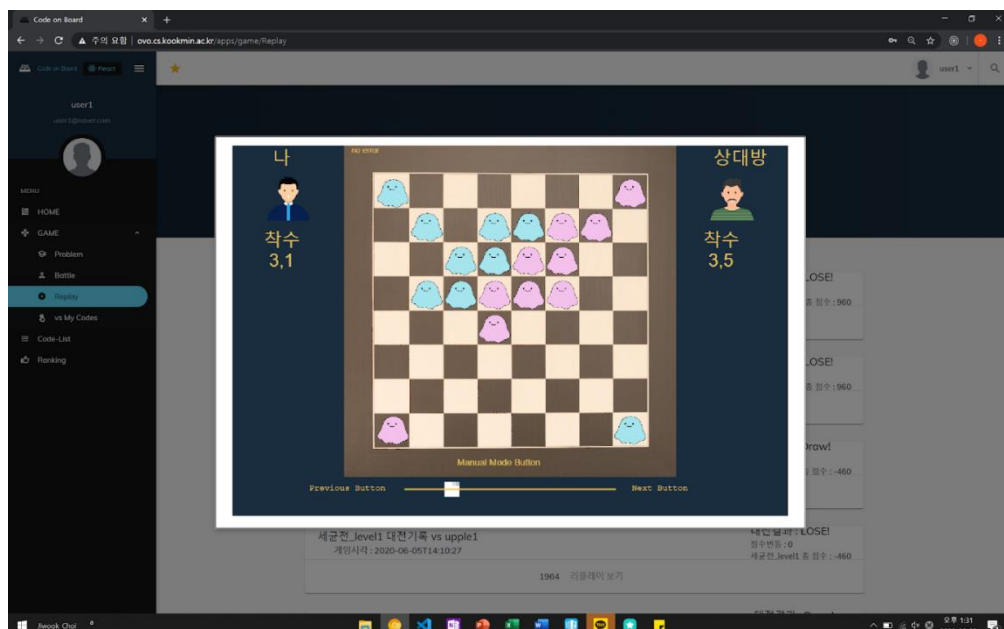
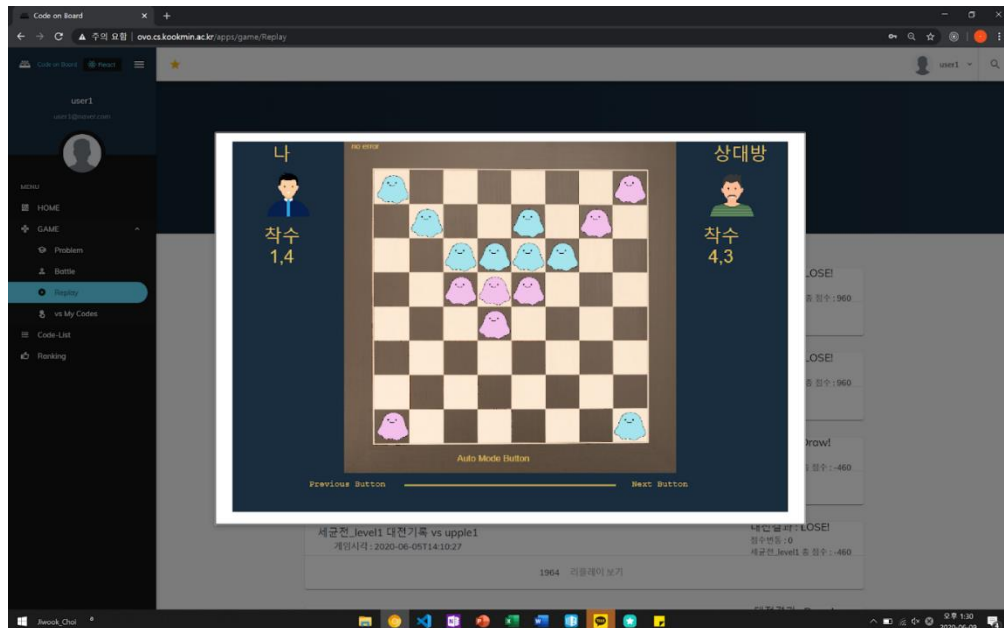
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

6. 유저가 그 동안 했던 모든 대전 기록은 Replay 에서 확인할 수 있다. Replay 를 통해 대전의 승패여부와 대전시각, 대전상대, 점수변동 등을 알 수 있다. 대전 내용을 시각적으로 확인하려면 Replay 리스트 내의 '리플레이 보기' 버튼을 클릭한다.



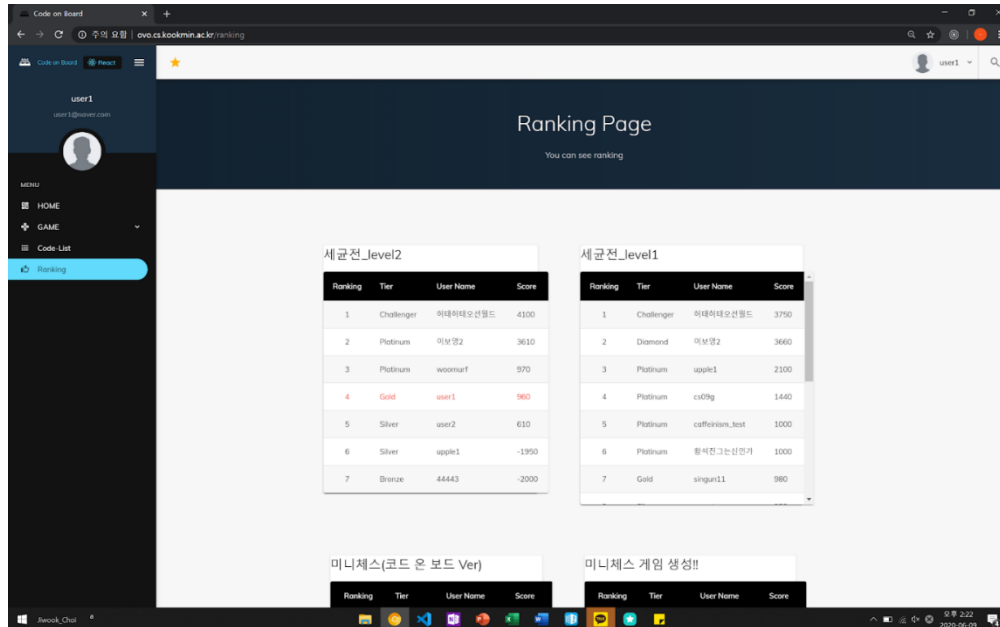
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

7. '리플레이 보기' 버튼을 눌러 나타난 modal 창을 통해 나와 상대방의 대전 내용을 시각적으로 확인할 수 있다. modal 창 하단의 슬라이드 바 혹은 'Previous button'과 'Next Button'을 이용하여 사용자의 코드로 실행한 게임이 어느 위치에 착수를 했는지 알 수 있다. (자동 모드와 수동 모드로 전환 가능)



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

9. Ranking page에서는 문제별 사용자 랭킹과 점수, 등급을 알 수 있다.

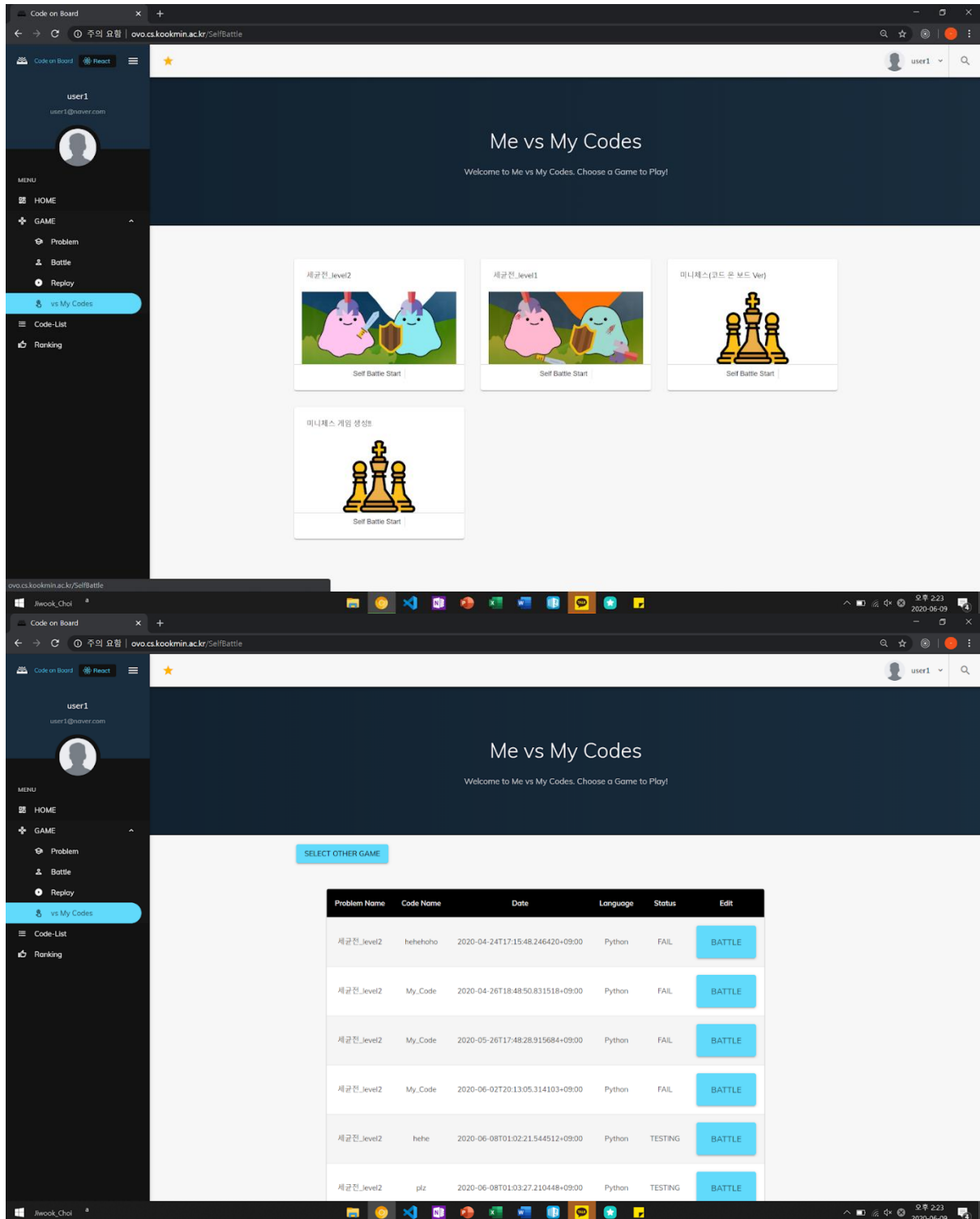


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

5.1.2 내 코드와 대전

1. 좌측의 GAME 메뉴의 vs My Codes 를 선택한다.
2. vs My Codes 페이지에서 문제별로 코드를 선택할 수 있다.
3. 선택한 코드의 BATTLE 버튼을 클릭하면 내 코드와 대전이 가능하다
4. 착수는 빈 곳을 클릭하면 되고 이동은 이동하고자 하는 돌을 클릭한 뒤 규칙에 맞는 빈 곳을 선택하면 이동이 완료된다.
5. 장면의 순서는 다음과 같다.
 - a. 보드판 초기 상태
 - b. 사용자가 변경한 보드판, 이 순서에만 유저는 착수 및 이동이 가능하다.
 - c. 사용자에게 의해 변경된 보드판
 - d. 코드에 의해 변경된 보드판
 - e. 이 뒤로는 b, c, d 보드판이 반복된다.
6. 리플레이 페이지와 동일하게 슬라이더바와 next, previous 버튼을 클릭하여 보드판을 이동할 수 있다.
7. 사용자는 previous 버튼 또는 슬라이더바를 이용해서 이전 보드판에서 다시 착수 및 이동이 가능하다. 이때 사용자가 선택한 시점 이후로는 보드판이 초기화된다.
8. modal 창 상단에 게임 진행 상태가 표시된다.
 - 1) Not finish : 게임 진행중
 - 2) Winner is challenger/opposite/draw : challenger 는 사용자의 승리, opposite 는 코드의 승리, draw 는 무승부
 - 3) Challenger error : 사용자의 규칙 위반
 - 4) Opposite error : 코드의 규칙 위반

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09



Me vs My Codes
Welcome to Me vs My Codes. Choose a Game to Play!

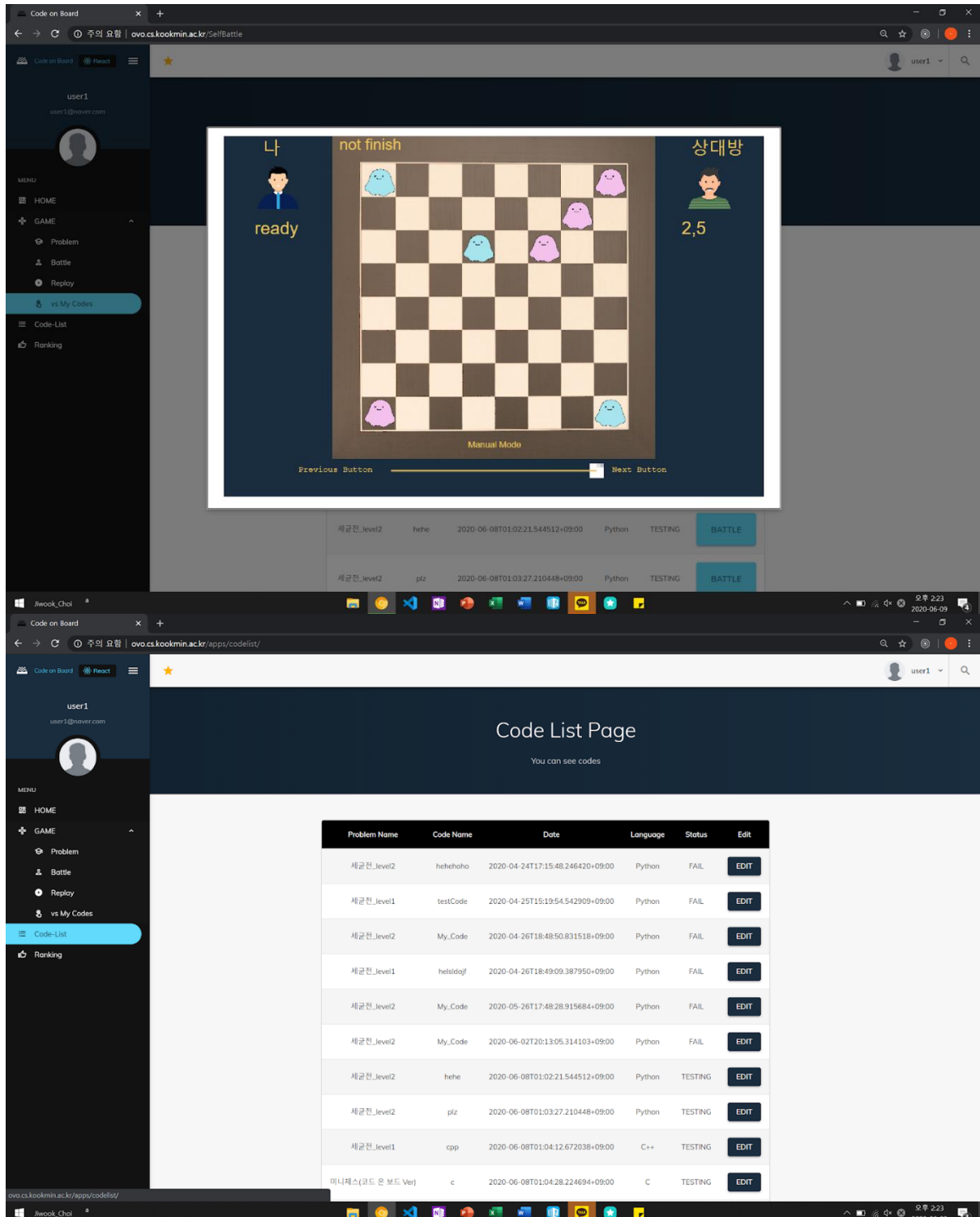
Game Cards:

- 세균전_level2
- 세균전_level1
- 미니체스(코드 폰 보드 Ver)
- 미니체스 게임 생성

SELECT OTHER GAME


Problem Name	Code Name	Date	Language	Status	Edit
세균전_level2	hehehoho	2020-04-24T17:15:48.246420+09:00	Python	FAIL	BATTLE
세균전_level2	My_Code	2020-04-26T18:48:50.831518+09:00	Python	FAIL	BATTLE
세균전_level2	My_Code	2020-05-26T17:48:28.915684+09:00	Python	FAIL	BATTLE
세균전_level2	My_Code	2020-06-02T20:13:05.314103+09:00	Python	FAIL	BATTLE
세균전_level2	hehe	2020-06-08T01:02:21.544512+09:00	Python	TESTING	BATTLE
세균전_level2	plz	2020-06-08T01:03:27.210448+09:00	Python	TESTING	BATTLE

 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09



The screenshot displays two views of the 'Code On Board' web application. The top view shows a game interface with a 10x10 grid, player avatars, and a 'Manual Mode' label. The bottom view shows the 'Code List Page' with a table of code submissions.

Problem Name	Code Name	Date	Language	Status	Edit
세균전_jevel2	hehehoho	2020-04-24T17:15:48.246420+09:00	Python	FAIL	EDIT
세균전_jevel1	testCode	2020-04-25T15:19:54.542909+09:00	Python	FAIL	EDIT
세균전_jevel2	My_Code	2020-04-26T18:48:50.831518+09:00	Python	FAIL	EDIT
세균전_jevel1	helsidajf	2020-04-26T18:49:09.387950+09:00	Python	FAIL	EDIT
세균전_jevel2	My_Code	2020-05-26T17:48:28.915684+09:00	Python	FAIL	EDIT
세균전_jevel2	My_Code	2020-06-02T20:13:05.314103+09:00	Python	FAIL	EDIT
세균전_jevel2	hehe	2020-06-08T01:02:21.544512+09:00	Python	TESTING	EDIT
세균전_jevel2	plz	2020-06-08T01:03:27.210448+09:00	Python	TESTING	EDIT
세균전_jevel1	csp	2020-06-08T01:04:12.672038+09:00	C++	TESTING	EDIT
미니캡스(코드 온 보드 Ver)	c	2020-06-08T01:04:28.224694+09:00	C	TESTING	EDIT

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

Code on Board

주요 사항

ovo.cs.kookmin.ac.kr/ViewProblemPage/1

user1

user1@naver.com

HOME

GAME

Problem

Battle

Replay

vs My Codes

Code-List

Ranking

PREVIOUS

NEXT

Page 1 of 4

세균전


게임 소개

세균전은 유자가 방인 자판에 행하는 위치를 통해 두는 게임입니다. 착수 후 그 주변에 상대방의 돌이 있다면 자신의 돌로 바꿀 수 있습니다. 보드판이 가득 차거나 더 이상 착수가 불가능한 경우, 보드판 위 돌의 개수가 더 많은 유자가 승리합니다. 게임은 아래 그림과 같은 9x8 형태의 보드판 위에서 진행됩니다. 보드판에서 본인의 돌은 1로, 상대방의 돌은 -1로, 아직 착수되지 않은 빈 공간은 0으로 표시합니다.

규칙 설명

1. 시작

게임은 8x8 보드판 위 네 모서리에 유자의 돌이 2개씩 서로 대각선으로 놓인 상태에서 시작합니다.



Python

```

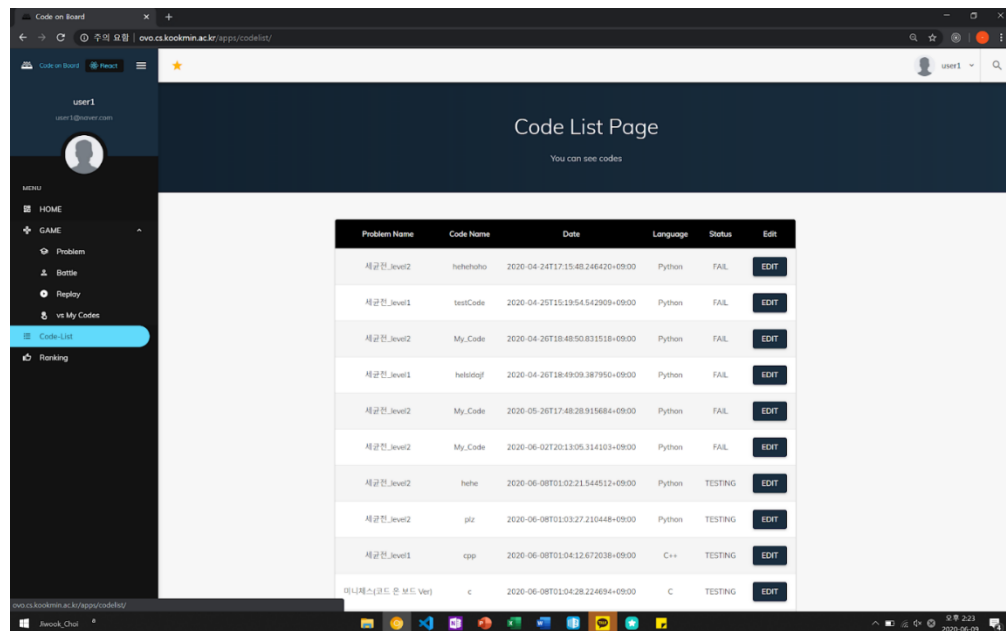
1 MAP_SIZE = 8
2 dx = [0, 0, 1, -1, 1, 1, -1, -1]
3 dy = [1, -1, 0, 0, 1, -1, 1, -1]
4 d = [1, 2, 4, 8, 0, 4, 2, 1]
5
6 def valid(x, y):
7     return 0 <= x < MAP_SIZE and 0 <= y < MAP_SIZE
8
9 def solve():
10    pos = []
11
12    a = [[list(map(int, input().split())) for i in range(MAP_SIZE)]
13         for j in range(MAP_SIZE)]
14    v = [[False for i in range(MAP_SIZE)] for j in range(MAP_SIZE)]
15    for i in range(MAP_SIZE):
16        for j in range(MAP_SIZE):
17            if a[i][j] != 1: continue
18            for k in range(8):
19                nx = i + dx[k]
20                ny = j + dy[k]
21                if not valid(nx, ny) or (a[nx][ny] != 0) or v[nx][ny]: continue
22                pos.append((nx, ny))
23                v[nx][ny] = True
24
25    max_score = -100000000
26    mx = 0
27    my = 0
28    for (x, y) in pos:
29        score = 0
30        for i in range(8):
31            nx = x + dx[i]
32            ny = y + dy[i]
33            if not valid(nx, ny): continue
34            if a[nx][ny] == 1: score += 10 * p[nx] * p[ny]
35            elif a[nx][ny] == -1: score += 10 * p[nx] * p[ny]
36
37    if max_score < score:
38        max_score = score
39        mx = x
40        my = y
41    return mx, my

```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

5.1.3 코드 보기 및 수정

1. Code-list 페이지에서 사용자가 그동안 제출한 문제의 이름과 코드의 이름, 제출시각, 사용 언어, 대전 가능 여부와 수정 버튼을 확인할 수 있다. 제출한 코드에 대한 정보를 확인할 수 있고, 원한다면 Edit 버튼을 통해 해당 코드를 수정할 수 있다.



2. Edit 버튼 클릭 시 코드 수정이 가능하다.

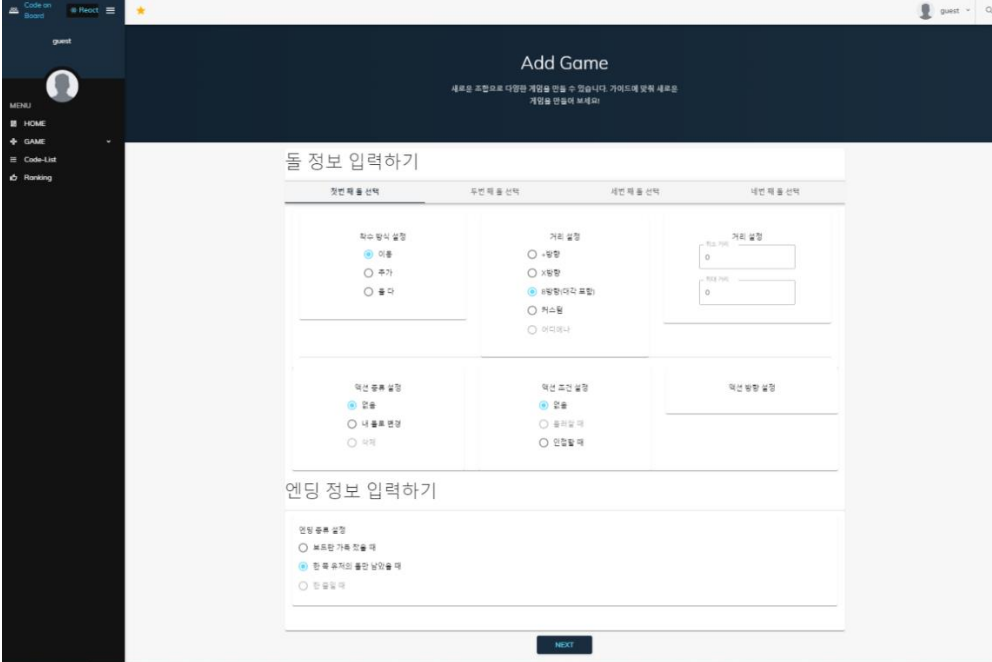
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

5.2 운영자 메뉴얼


5.2.1 문제 만들기

1. 관리자 권한이 있는 계정으로

<http://ovo.cs.kookmin.ac.kr/apps/game/addgame> 에 접속한다.



2. 페이지에 접속하면, 현재 구현되어 있는 돌에 대한 규칙이 리스트업 되어있다. 상단에 각 돌의 탭을 선택하여 규칙을 정할 수 있다.
3. 규칙은 착수규칙, 액션규칙, 엔딩규칙으로 이루어져 있다. 아래와 같은 방법으로 규칙을 정할 수 있다.

 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

● 착수규칙 [↗]

예시) [↗]

"placement" : { "돌의 번호" : [착수종류, [[착수규칙1, 착수규칙2...]], [↗]
[착수옵션1, 착수옵션2...]] }[↗]

✓ 착수종류[↗]

0. 이동만[↗]
1. 추가만[↗]
2. 둘 다[↗]

[↗]

✓ 착수규칙 [↗]

이동[↗]

0. 4방 (착수규칙, 최소, 최대)[↗]
1. 대각선 (착수규칙, 최소, 최대)[↗]
2. 8방 (착수규칙, 최소, 최대)[↗]
3. 커스텀 (착수규칙, x, y)[↗]

추가 ex) 추가 착수규칙 예시: (착수규칙, 방향번호)[↗]

4. 인접한 곳 [↗]


➤ 인접방향번호[↗]

0. 4방[↗]
1. 대각선[↗]
2. 8방[↗]

5. 아무데나[↗]

✓ 착수 옵션[↗]

1. 돌을 건너뛸 수 있는가[↗]
2. 착수한 곳에 상대말 있으면 먹기[↗]
3. 상대방 돌을 뒤집을 수 있는 곳에만 돌 추가 가능[↗]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

● 액션규칙

예시)

"action" : { "들의 번호" : [액션조건, 액션방향, 액션방법] }

✓ 액션 조건

0. 없음

1. 둘러싸기

2. 인접한 곳

✓ 액션 방향

0. 없음

1. 양 옆

2. 위아래

4. 4방

5. 대각선

6. 8방

✓ 액션 방법

0. 없음

1. 뒤집기

2. 삭제

 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

• 엔딩규칙

예시)

"ending" : { [[엔딩조건]] } -> 옵션은 구현 안돼서 제외했음

✓ 엔딩 조건

0. 보드판이 가득 찼을 경우

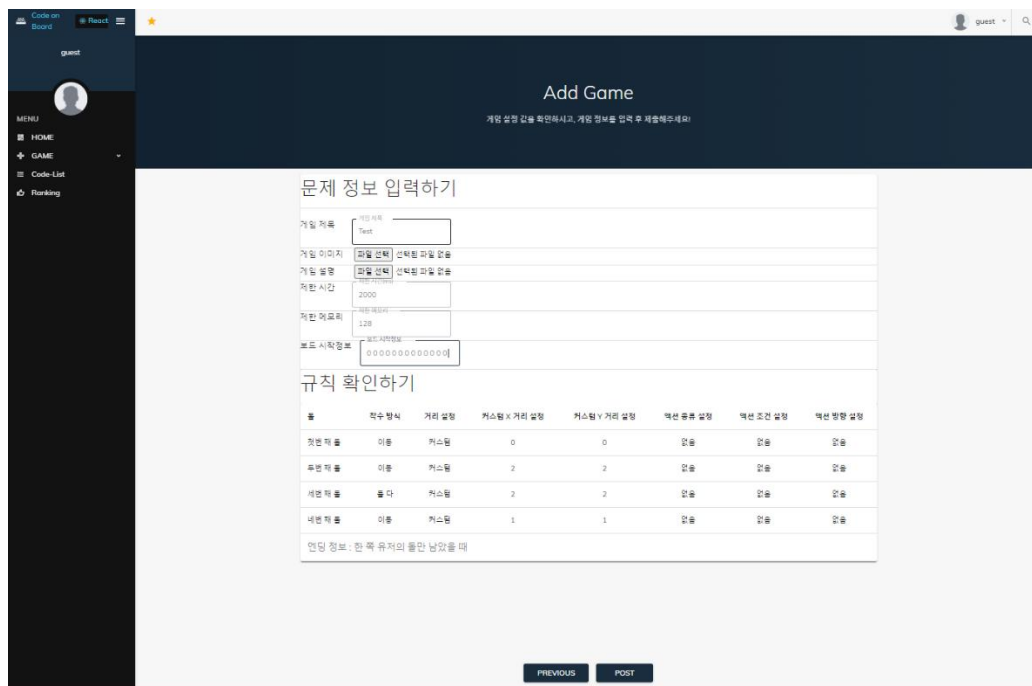
1. 한쪽 유저의 돌만 남았을 경우

2. 한 줄일 때

✓ 엔딩 옵션

0. 한 줄일 때 돌의 개수

4. 규칙을 모두 정하고 NEXT 버튼을 눌러 다음 단계로 이동한다.



문제 정보 입력하기

가일 제목:

가일 이미지:

가일 설명:

제한 시간:

제한 점수:

보드 시작정보:

규칙 확인하기

종	적수 형식	거리 설정	커스텀 X 거리 설정	커스텀 Y 거리 설정	적선 종류 설정	적선 조건 설정	적선 방향 설정
첫번 적 돌	이동	커스텀	0	0	없음	없음	없음
두번 적 돌	이동	커스텀	2	2	없음	없음	없음
세번 적 돌	돌다	커스텀	2	2	없음	없음	없음
네번 적 돌	이동	커스텀	1	1	없음	없음	없음

엔딩 정보: 한 쪽 유저의 돌만 남았을 때

PREVIOUS POST

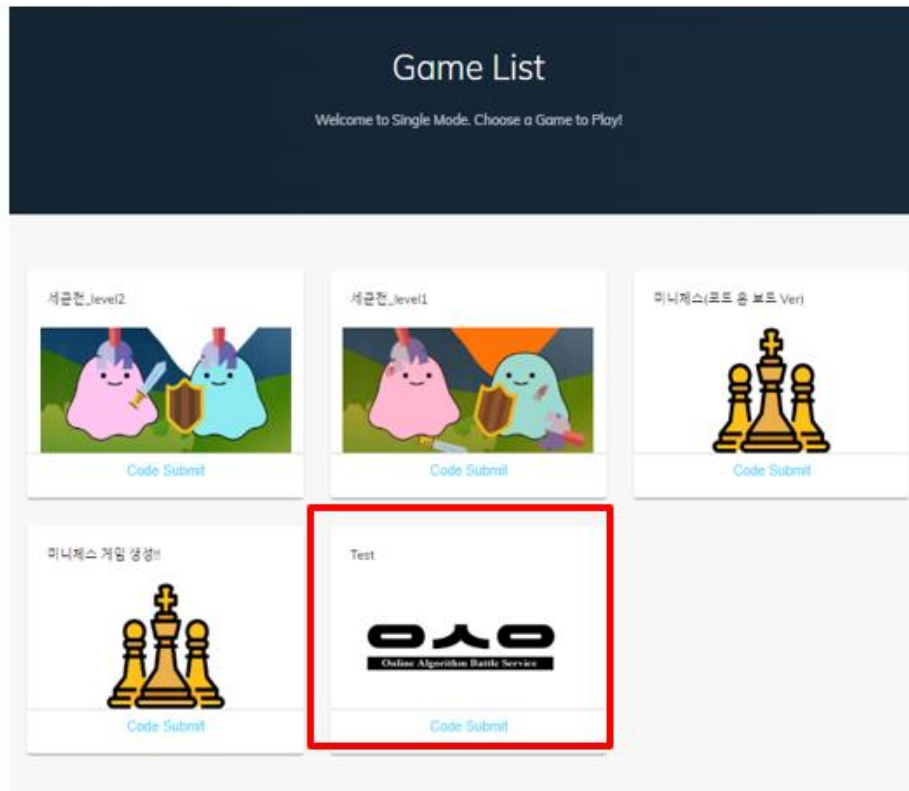
5. 문제에 대한 정보를 입력한다.

6. 마지막 표에서 돌에 대한 규칙들을 확인할 수 있다.

7. 모든 설정을 완료하면 POST 버튼을 눌러 문제를 생성한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

8. 문제목록 페이지에서 만든 문제를 확인할 수 있다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Code On Board	
	팀 명	알고리즘 서비스 연구소	
	Confidential Restricted	Version 2.0	2020-JUN-09

5.3 배포 가이드

gitclone을 이용해 Repository를 받는다

5.3.1 클라이언트 서버

1.Installation

```
# clone the repo
$ git clone https://github.com/kookmin-sw/capstone-2020-16.git

# go to app's directory
$ cd capstone-2020-16/front/

# install app's dependencies
$ yarn install
```

2.Create React App

```
# dev server with hot reload at http://localhost:3000
$ yarn start
```

3.build

```
# build for production with minification
$ yarn build
```

5.3.2 API 서버

1. pip install requirements.txt 를 사용하여 필요한 모듈을 설치한다.
2. capstone-2020-16/api 위치에서 다음 명령어를 실행한다.
3. python3.6 manage.py runserver 0.0.0.0:port
4. 설정한 포트로 API서버를 배포할 수 있다.

5.3.3 Core 서버

- 1.Docker, Celery 와 redis 를 설치한다.
- 2.capstone-2020-16/core 디렉토리에서 sudo docker build -t core . 명령어를 통해 docker image 를 생성한다.
- 3.터미널 창에서 redis-server 를 통해 redis 를 켜다
- 4.capstone-2020-16/core 디렉토리에서 Celery -A tasks worker -loglevel=info 명령어를 통해 Celery 를 실행한다.