

Code On Board

KMU Software Capstone Project 알고리즘 서비스 연구소
16조 최명서 | 최지욱 | 박호준 | 우현웅 | 강수련 | 칼리드



INDEX

- 01 질의 응답
- 02 수정 사항
- 03 수행 내용
- 04 향후 계획

질의 응답

Q & A

Q . 난이도가 높지 않은 게임을 추가하여 시연 효과를 높인다면 좋은 반응을 얻을 수 있으리라 생각합니다.

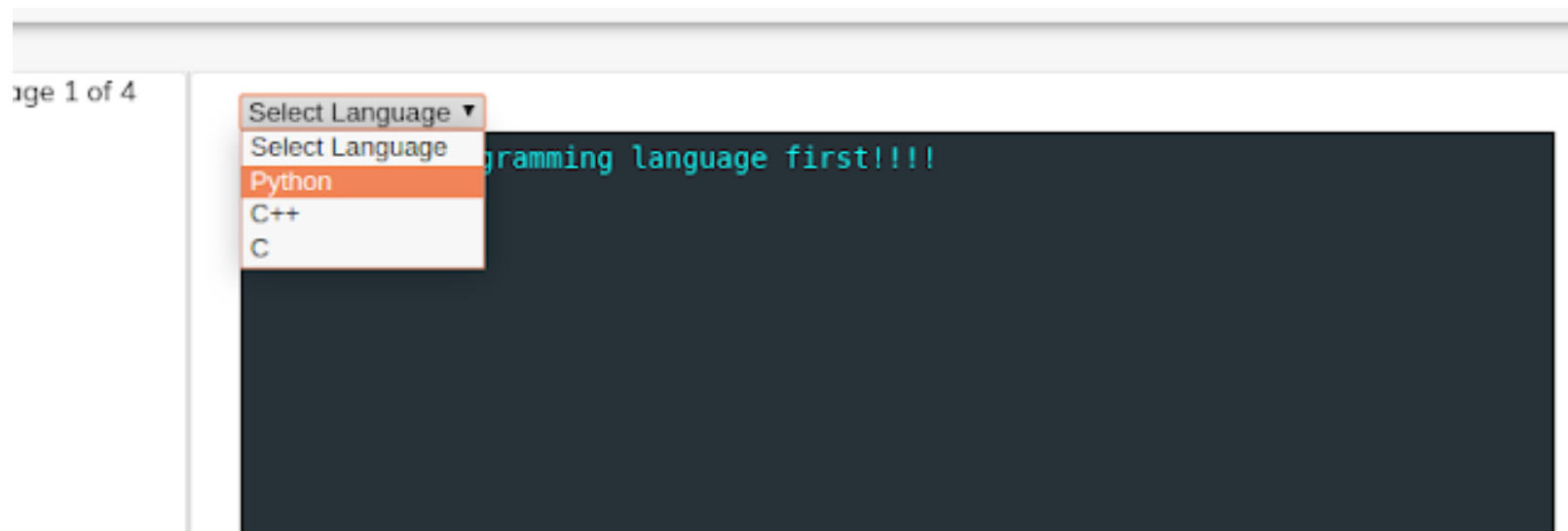
기존 보드 게임에서 규칙을 간소화하여 난이도를 낮춘 게임을 개발했습니다.

예를 들어 기존 체스 게임에서 돌의 개수를 3~4개로 줄인 미니 체스를 개발했습니다.

Q . 입력이 2개의 소스코드(또는 알고리즘)라고 생각되는데 소스코드 어떤 언어로 작성된 것인지, 언어와 무관하게 모든 언어의 소스코드인지 등을 명확히 하기 바랍니다.

프로그래머스나 백준처럼 코드 제출 시 사용 언어를 선택하도록 구성되어 있습니다.

코드 제출 페이지에서 에디터 윗 부분에 위치하고 있습니다.



Q . 새로운 게임을 만들 때 어떤 규칙들이 있는지, 규칙을 BNF, json, XML 등 어떤 포맷으로 표현했는지, 규칙들을 조합할 때 어떤 방법으로 조합했는지를 명확히 하기 바랍니다.

게임 규칙은 각 돌에 대한 규칙과 게임 종료 규칙으로 정의됩니다.

돌에 대해 규칙은 2종류로 구분되어 있습니다.

착수 규칙 : 돌을 놓거나 이동시킬 때 적용되는 규칙입니다.

액션 규칙 : 돌을 놓거나 이동시킨 후 발생하는 이벤트에 대한 규칙입니다.

ex. 이동 시 근처의 돌을 내돌로 변경 / 이동, 착수 위치의 적 돌을 내돌로 변경

엔딩 규칙 : 게임이 종료되는 상황에 대한 규칙입니다.

ex. 보드판이 꽉 차면 게임 종료

Q . 새로운 게임을 만들 때 어떤 규칙들이 있는지, 규칙을 BNF, json, XML 등 어떤 포맷으로 표현했는지, 규칙들을 조합할 때 어떤 방법으로 조합했는지를 명확히 하기 바랍니다.

각 문제는 rule 데이터를 가지고 있으며 다음과 같이 json으로 표현됩니다.

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

Q . 새로운 게임을 만들 때 어떤 규칙들이 있는지, 규칙을 BNF, json, XML 등 어떤 포맷으로 표현했는지, 규칙들을 조합할 때 어떤 방법으로 조합했는지를 명확히 하기 바랍니다.

각 규칙은 정수로 표현되며 매칭되는 내용은 팀 내 문서로 관리되고 있습니다.

• 착수규칙

예시)

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

✓ 착수종류

- 0. 이동만
- 1. 추가만
- 2. 둘 다

✓ 착수규칙

이동 ex) 이동 착수규칙 예시: (착수규칙, x, y)

- 0. 4방
- 1. 대각선
- 2. 8방
- 3. 커스텀

추가 ex) 추가 착수규칙 예시: (착수규칙, 방향번호)

- 4. 인접한 곳
 - 인접방향번호

• 액션규칙

예시)

"action" : { "돌의 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

• 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

Q . 과제 수행의 결과로 어떤 새로운 게임을 만들었는지 그 예를 보여주면 좋겠습니다.

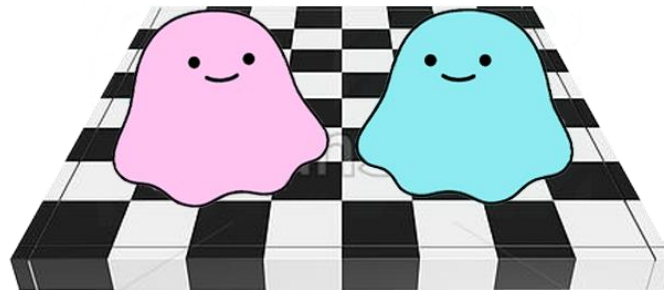
현재 문제 만들기 기능이 개발 중에 있어 해당 기능을 이용한 새로운 게임이 제작되지 않았습니다.

하지만 자체적으로는 세균전과 미니 체스를 구현하였습니다.

Q . 프로젝트 명이 좀 더 구체적이었으면 좋겠습니다.
알고리즘은 일반적인 명칭이라고 느껴집니다.

교수님의 의견에 저희도 동의하여 프로젝트 명을 변경하였습니다.

기존 Online Algorithm Battle Service에서 Code on Board로 변경하여
보드 게임에 대한 코드를 작성하는 플랫폼이라는 느낌을 주었습니다.



수정 사항

Revised Plan

코드 테스트 방법 변경

변경 전 : 테스트 케이스를 이용한 코드 테스트

변경 후 : 유저가 제출한 코드A를 이용해 A vs A 대전을 실행
대전 중에 에러가 발생하지 않으면 테스트 통과

점수를 반영하는 대전의 변경

변경 전 : 유저가 매칭을 눌러 대전 후 결과에 따라 점수 변동

변경 후 : 전수조사를 통한 대전만 점수에 반영

테스트 인터뷰

변경 전 : 베타테스트 후 테스터들에게 구글 폼을 이용한 설문조사

변경 후 : 테스터 중 적극적으로 참여한 테스터분들을 Zoom을 통해 인터뷰

사유 : 설문조사를 통하여 깊은 내용을 얻을 수 없고, 서비스에 대한 질문지를 제작하기에 경험 및 전문성이 부족하다고 판단

수행 내용

Progress

테스트 진행

4월 23일 ~ 5월 8일 (15일) 간 베타 테스트를 진행

30명 이상의 테스터 / 약 300개의 코드 / 약 1500개의 게임

세균전_level2

Ranking	Tier	User Name	Score
1	Challenger	허태허태오션월드	4100
2	Platinum	이보영2	3610
3	Platinum	user1	1000
4	Gold	woomurf	970
5	Silver	user2	610
6	Silver	upple1	-1950
7	Bronze	44443	-2040

세균전_level1

Ranking	Tier	User Name	Score
1	Challenger	허태허태오션월드	3750
2	Diamond	이보영2	3660
3	Platinum	upple1	2100
4	Platinum	cs09g	1440
5	Platinum	cafeinism_test	1000
6	Platinum	황석진그는신인가	1000
7	Gold	singun11	980

1 2 3 4 304 코드정보

1 2 3 4 ... 15 16 1528 게임정보

테스트 결과

내용	일자
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
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

〈에러 리스트〉

내용	일자
공지 내용을 웹서비스 내에서 팝업화	04월 23일
Problem / Battle 페이지가 헛갈림	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

〈피드백 리스트〉

테스트 기간동안 발생한 에러와 서비스에 대한 피드백 등을 문서화 하여 정리

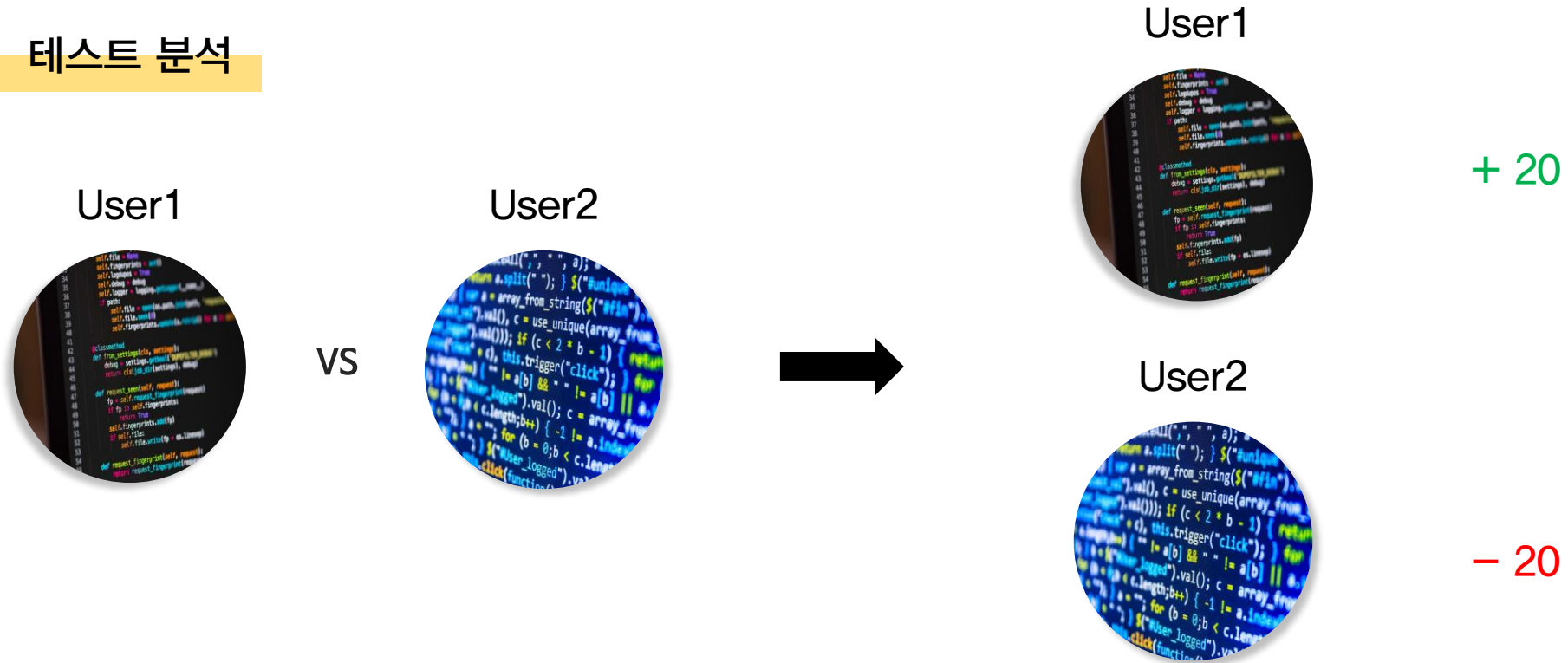
테스트 분석

1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	4100	-2040	2020년 5월 8일 10:50 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	4080	-2020	2020년 5월 8일 10:49 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	4060	-2000	2020년 5월 8일 10:48 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	4040	-1980	2020년 5월 8일 10:48 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	3960	-1960	2020년 5월 8일 10:46 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	3940	-1940	2020년 5월 8일 10:46 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	3900	-1920	2020년 5월 8일 10:45 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON
1_세균전_level2_admin	히태허태오션월드	44443	challenger	finish	3860	-1900	2020년 5월 8일 10:44 오후	650_히태허태오션월드_세균전_level2_PYTHON	340_44443_세균전_level2_PYTHON

같은 코드로 여러 번 대전

동일한 코드끼리의 대전이 유의미하지 않고,
단순히 점수를 올리는 것으로 판단되어 점수 측정 방법을 변경하기로 결정

테스트 분석



기존 방식은 대전 후 바로 점수가 반영되는 방식

하루에 한번 전수 조사를 통해 점수를 반영하는 방식으로 변경

테스트 분석

선수

후수



VS



VS



VS



VS



⋮



User1



4승 0패

1 등

User2



2승 2패

2 등

User3



0승 4패

3 등

01

수행 내용 : 베타테스트

KMU Software Capstone Project



02

03

테스트 분석

04

선수

후수



VS



VS



VS



VS



⋮

User1

4승 0패

1 등

모든 유저가 선수로 한번,
후수로 한번 게임을 진행

선수, 후수가 게임에 영향을 줄 수 있기 때문

User2

2승 2패

2 등

User3

0승 4패

3 등

01

수행 내용 : 베타테스트

KMU Software Capstone Project CoB

02

03

테스트 분석

04

선수

후수

전수 조사는 하루에 한번 혹은
관리자가 원하는 만큼 실행할 수 있음

전수 조사로 인한 대전만 점수에 반영

VS

기존 대전 방식은 그대로 남겨두고
점수에 반영하지 않음

VS

VS

⋮



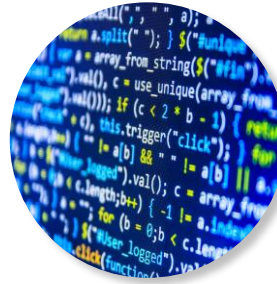
User1



4승 0패

1 등

User2



2승 2패

2 등

User3



0승 4패

3 등

01

수행 내용 : 클라이언트 서버

02

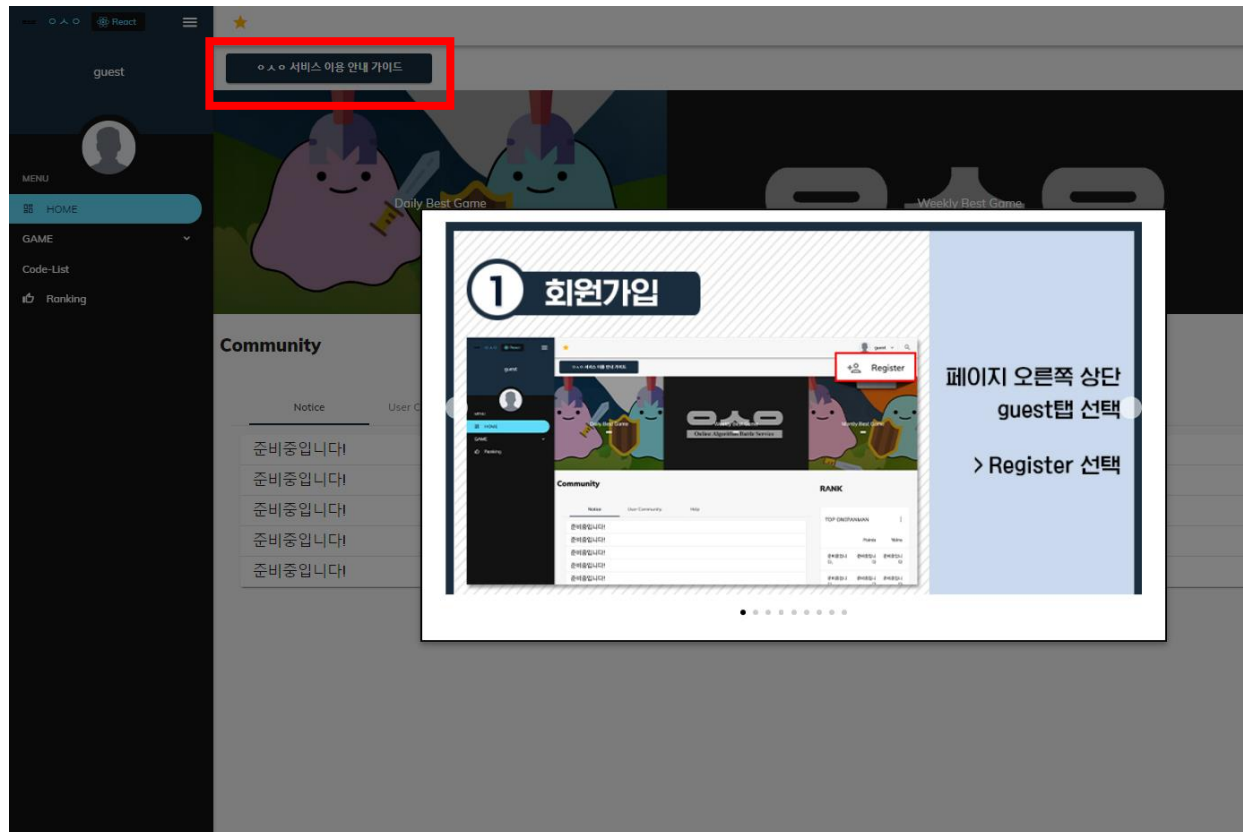
03

이전에 수행한 내용

04

- ✓ API 서버와 연동
- ✓ 리플레이 기능 구현
- ✓ 클라이언트 서버 배포

서비스 가이드 팝업



서비스 사용 방법을 담은 팝업창 제작

베타서비스 중 테스터들의 요청으로 제작하게 됨.

01

02

03

04

수행 내용 : 클라이언트 서버

KMU Software Capstone Project CoB

랭킹 페이지

user1

Ranking Page

You can see ranking

세균전_level2

Ranking	Tier	User Name	Score
1	Challenger	하태하태오션월드	4100
2	Platinum	이보영2	3610
3	Platinum	woomurf	970
4	Gold	user1	960
5	Silver	user2	610
6	Silver	upple1	-1950
7	Bronze	44443	-2000

세균전_level1

Ranking	Tier	User Name	Score
1	Challenger	하태하태오션월드	3750
2	Diamond	이보영2	3660
3	Platinum	upple1	2100
4	Platinum	cs09g	1440
5	Platinum	caffeinism_test	1000
6	Platinum	황석진그는신인가	1000
7	Gold	singun11	980

각 게임별로 랭킹을 표시

자신은 빨간색으로 표시

01

02

03

04

수행 내용 : 클라이언트 서버

KMU Software Capstone Project CoB

코드 리스트 페이지

Date	Submit Time	Language	Error
2020-01-05	12:07	Python	O

개발 중인 기능

자신의 코드 리스트 확인

코드 상태를 확인 가능
(게임 가능 , 불가능 , 테스트 중)

문제 만들기 페이지

돌 정보 입력하기

첫번째 돌 선택 두번째 돌 선택 세번째 돌 선택 네번째 돌 선택

작수 방식 설정

☒ 이동

☐ 추가

☐ 뺄다

거리 설정

☒ + 방향

☐ X 방향

☐ 8방향(대각 포함)

☐ 커스텀

☐ 어디에나

거리 설정

최소 거리

최대 거리

액션 종류 설정

☒ 없음

☐ 내 돌로 변경

☐ 삭제

액션 조건 설정

☐ 없음

☐ 돌 위치 변경

☒ 인접할 때

액션 방향 설정

☐ 없음

☐ 양 옆

☐ 위 아래

☒ X 방향

☐ + 방향

☐ * 방향

엔딩 정보 입력하기

엔딩 종류 설정

☒ 보드판 가득 찼을 때

☐ 한 쪽 유저의 돌만 남았을 때

☐ 한 돌만 때

개발 중인 기능

돌 마다 규칙을 설정하여 문제 생성

문제 정보 입력하기

게임 제목

게임 이미지 선택된 파일 없음

게임 설명 선택된 파일 없음

제한 시간 제한 시간(min)

제한 메모리 제한 메모리

보드 시작정보

규칙 확인하기

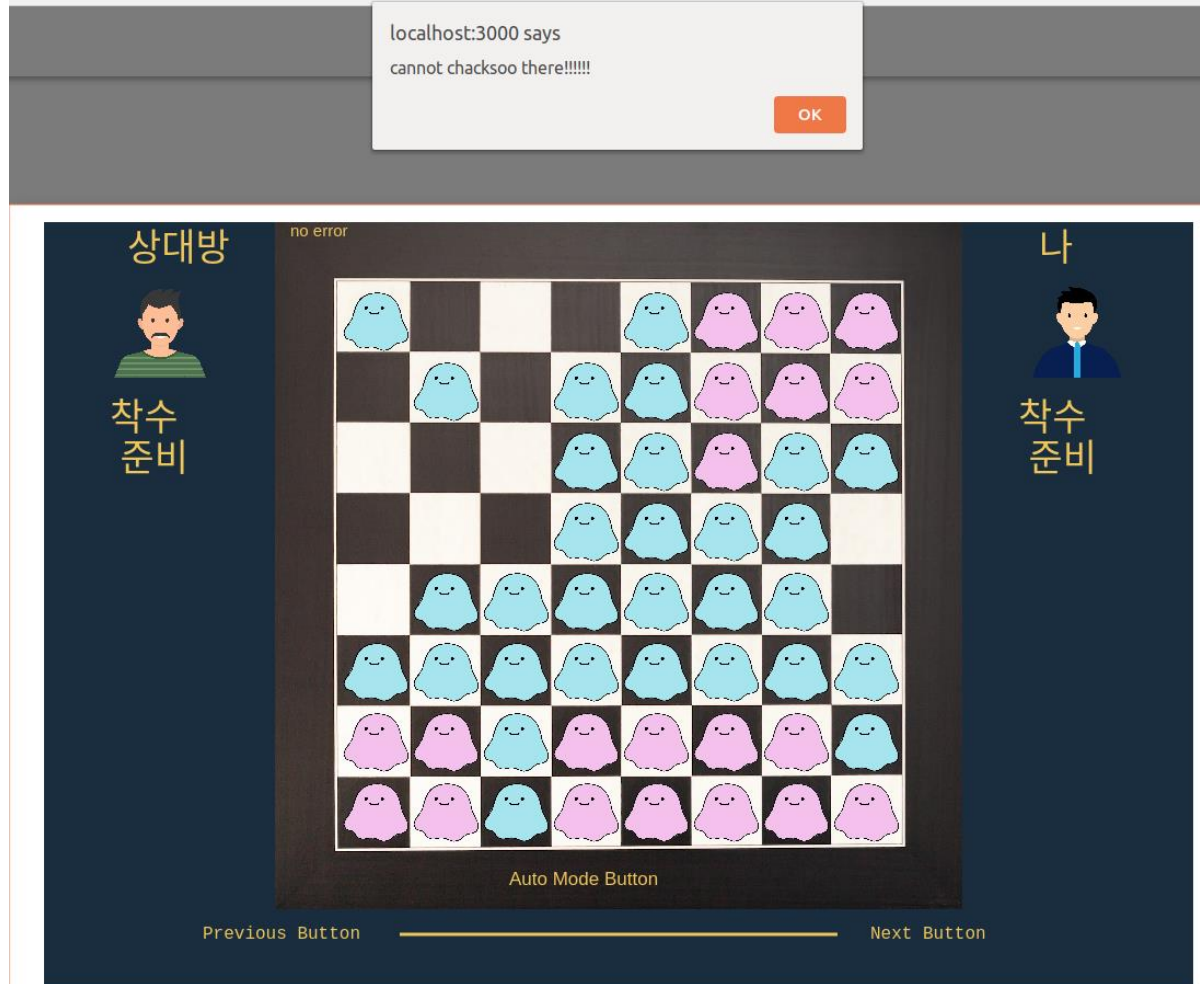
돌	작수 방식	거리 설정	커스텀 X 거리 설정	커스텀 Y 거리 설정	액션 종류 설정	액션 조건 설정	액션 방향 설정
첫번째 돌	이동	커스텀	3	3	없음	없음	X 방향
두번째 돌	이동	커스텀	3	33	없음		
세번째 돌	뺄다	커스텀	3	3	내 돌로 변경	없음	
네번째 돌	이동	커스텀	5	5	없음	인접할 때	양 옆

엔딩 정보 : 보드판 가득 찼을 때

PREVIOUS

POST

나와의 대전 페이지



개발 중인 기능

보드판을 클릭하면 착수 가능

착수할 수 없는 곳을 클릭하면
오류 경고창을 띄워줌

이전에 수행한 내용



DB 설계



API 구현 및 온라인 API문서 제작



JWT 토큰을 이용한 로그인 기능

전수조사 매칭기능 구현

```

class Matchall(APIView):

    check = False

    def post(self, request, version):

        problemid = int(request.data['problemid'])
        queryset = UserInformationInProblem.objects.all().filter(
            problem=problemid, available_game=True)

        problems = Problem.objects.all().filter(id=problemid)
        problem = problems[0]

        rule = problem.rule
        rule = json.loads(rule)

        for query in queryset:
            print(query.user.username)

        for challenger_ in queryset:
            for opposite_ in queryset:
                challenger = UserInformationInProblem.objects.all().filter(id=challenger_.id)[0]
                opposite = UserInformationInProblem.objects.all().filter(id=opposite_.id)[0]

                if challenger.user.pk == opposite.user.pk:
                    continue

                if challenger.available_game is False or challenger.code.available_game is False:
                    break

                if opposite.available_game is False or opposite.code.available_game is False:
                    continue

```

문제마다 모든 유저들이 선수로 1번,
후수로 1번 대전하도록 하는 전수조사 기능을 개발

코드 테스트 기능

```

def create(self, request, *args, **kwargs):
    data = super().create(request, *args, **kwargs)
    data = data.data

    # 여기서 celery 코드 추가!
    tasks.test_code.delay()

    return Response(data)

def update(self, request, *args, **kwargs):
    data = super().update(request, *args, **kwargs)
    data = data.data

    available = data["available_game"]

    # 게임 가능한 코드이면
    if available:
        user = data["author"]
        problem = data["problem"]

        queryset = UserInformationInProblem.objects.all().filter(user=user, problem=problem)

        # userInformationInProblem 객체가 존재하는지 확인하고 생성한다.
        if len(queryset) < 1:
            create_instance(user, problem, data["id"])

    return Response(data)

```

코드가 제출되면 core서버로 테스트를 요청

Core 서버에서 테스트 결과를 업데이트

01

수행 내용 : Core서버

02

03

이전에 수행한 내용

04



대전 기능



세균전 게임 구현

게임 진행 순서

1. 보드판을 Input으로 유저의 코드를 실행
2. Output으로 착수 값을 받아 착수 규칙에 맞는지 확인
3. 착수 규칙에 맞다면 보드판을 갱신하고 액션 규칙을 실행 (액션 규칙이 없다면 pass)
4. 엔딩 규칙을 통해 게임이 끝났는지 검사
5. 갱신된 보드판을 상대방에게 Input으로 넘겨줌
6. 1 ~ 5번 로직을 게임이 끝날 때 까지 반복

난이도가 낮은 게임 개발

진입 장벽이 높을 것을 생각하여 쉬운 난이도의 게임을 개발

기존 세균전에서 규칙을 줄여 난이도를 낮춘 세균전 Level1 개발

기존 체스에서 돌의 개수를 3~4개로 줄여 난이도를 낮춘 미니 체스 개발

코드 테스트 기능

Code A



Redis & Celery



코드 제출

Code A



VS

Code A



제출된 코드로 대전을 진행



Error 발생하면 게임 불가 / Error 발생 안하면 게임 가능



API서버에 결과 업데이트

향후 계획

Plans

개발 계획

유저 정보 페이지

코드 리스트 페이지

문제 만들기 기능

나와의 대전 기능

감사합니다

Code on Board

KMU Software Capstone Project 알고리즘 서비스 연구소

16조 최명서 | 최지욱 | 박호준 | 우현웅 | 강수련 | 칼리드