

The logo features the word "RogueWrite" in a white, sans-serif font, centered within a black, irregularly shaped rectangle. This rectangle is framed by a red line that follows its perimeter with some offsets, creating a layered effect. Small red dots are placed at the corners and midpoints of the red frame.

RogueWrite

담당교수
지준 교수님

Github: <https://github.com/p9595jh/RogueWrite>

1494052 박재훈
1494037 채수현

CONTENTS

1

프로젝트 소개

- 개발 주제
- 프로젝트 구조

2

프로젝트 개발 내용

- 개발 내용 & 시행착오
- 사용 기술

3

아쉬운 점 및 느낀 점

- 아쉬운 점
- 느낀 점

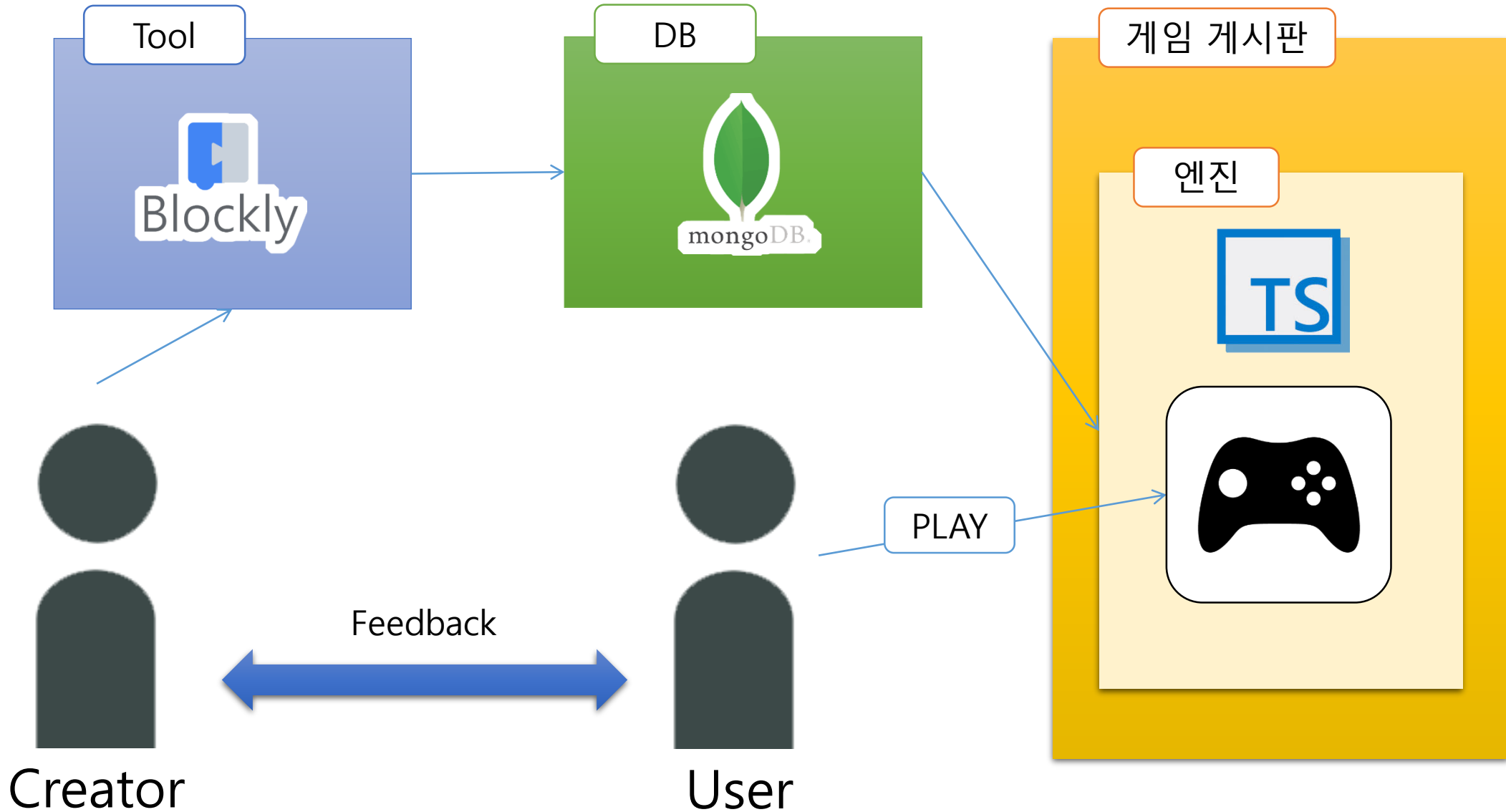


게임제작 + 사이트 플랫폼

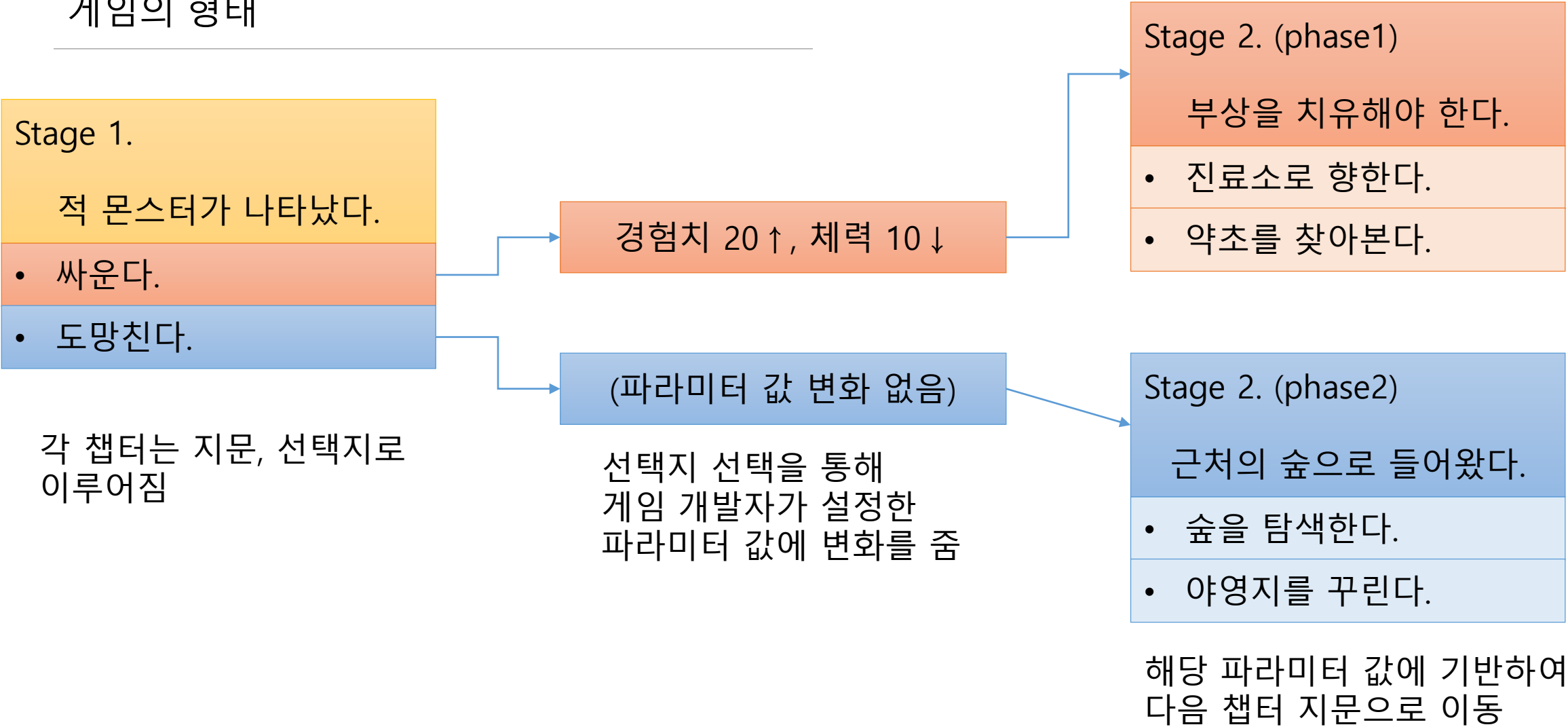
Blockly를 이용하여 만든 게임들을 웹 사이트를 통해 공유하고 피드백을 받을 수 있는 웹 플랫폼.

사이트를 통해 유저들의 게임 제작 및 공유 환경을 최대한 지원한다.

1 프로젝트 구조



게임의 형태



사이트로서의 기본적인 기능

- 회원가입, 회원정보수정 & 로그인
- 게시글 작성, 수정, 삭제
- 게시글 댓글 (+ 대댓글)

회원가입 & 로그인

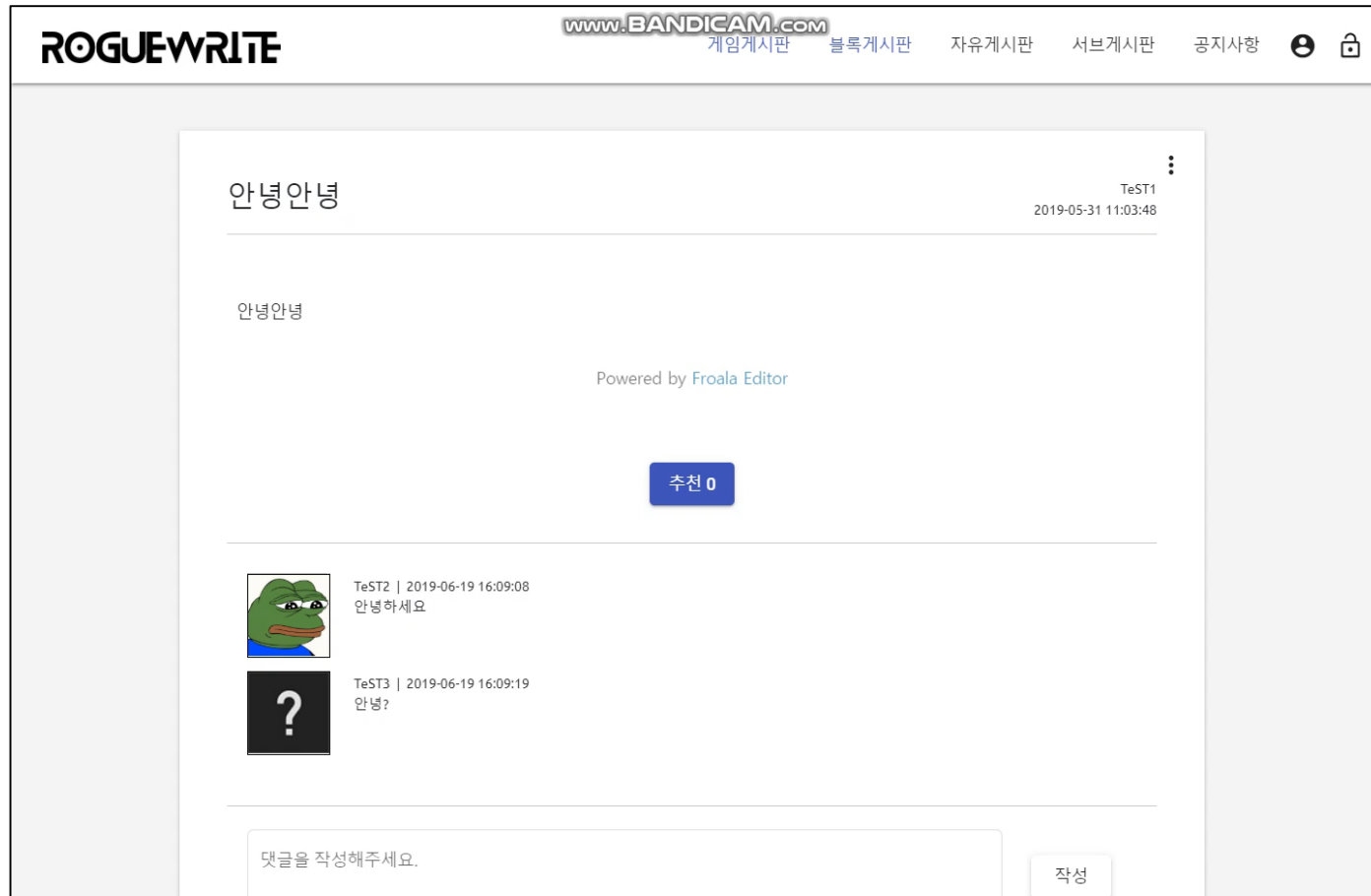
회원가입을 할 경우 비밀번호는 Bcrypt 패키지를 통해 암호화 되어 DB에 저장되며,
로그인 할 때에는 JWT 토큰 방식을 이용하여 로그인 하게 됩니다.

게시글 댓글 (+ 대댓글)

댓글로 링크를 작성하면 별다른 처리 없이 클릭할 수 있는 링크로 등록됩니다.

댓글을 쓸 때 대괄호를 두 번 ([[) 입력하게 되면 그 글에 댓글을 쓴 사람 목록을 불러와서 태그할 수 있게 됩니다.

다른 사람의 댓글의 프로필 이미지를 누르면 그 사람의 댓글에 대댓글을 쓸 수 있습니다.



- 다른 사람 태그
- 대댓글

Blockly를 이용한 하나의 IDE로써 작동

- 블록을 이용한 절차지향식 개발
- 제작, 로깅 기능과 협업 등의 개발 관리 기능을 지원

게임 만들기

ROGUEWRITE

- 게임 제작 & 로깅
- 협업
- 다른 사람의 게임 복사

게임 제작

문제점) 게임 콘텐츠를 Blockly 문자열을 띄우게 하였는데, 바이너리 파일을 넣지 못하는 문제가 생겼습니다.

해결) HTML 코드를 집어넣게 함으로써 해결하였습니다.

게임 제작

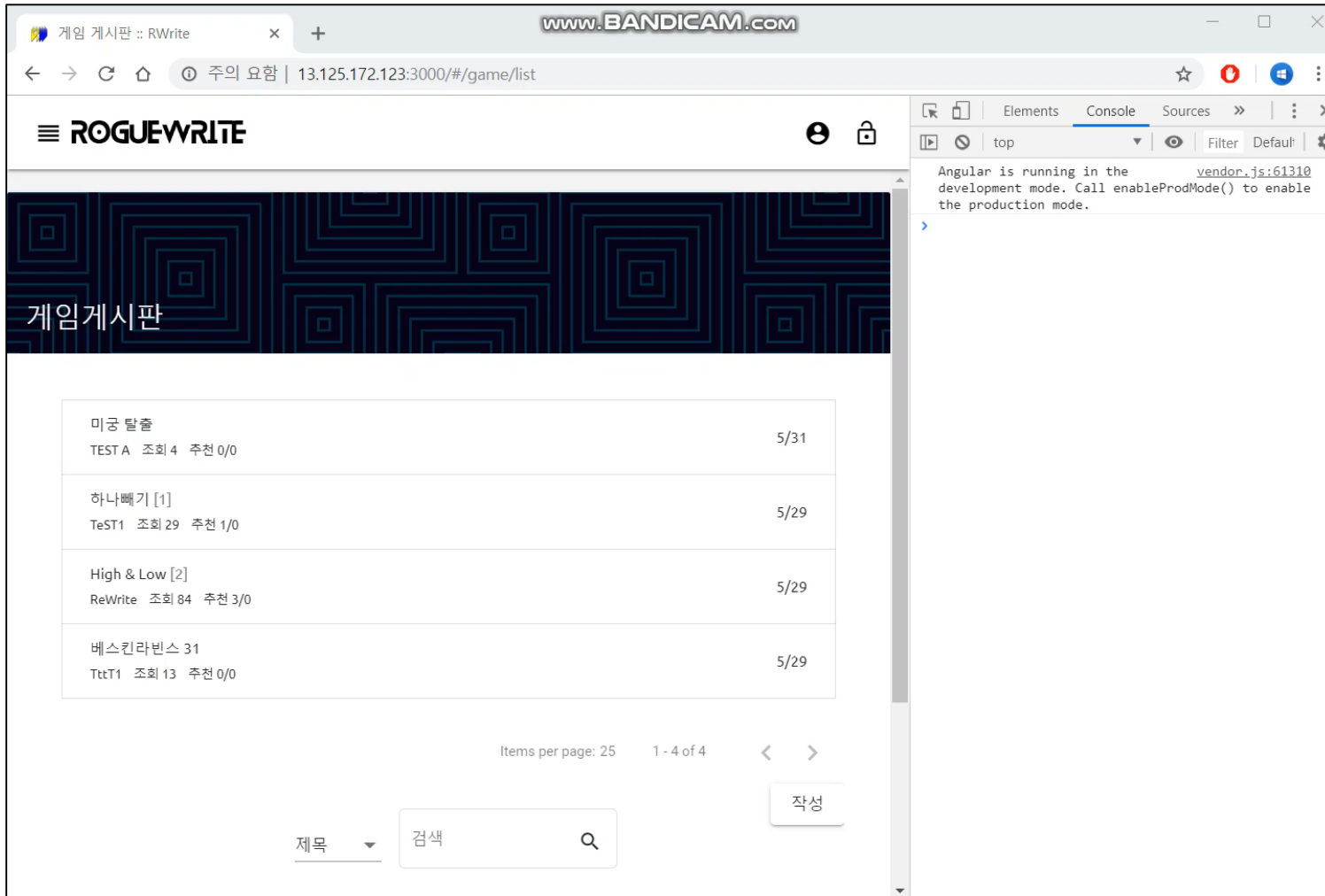
문제점) Angular에 저희가 자체적으로 수정한 Blockly를 탑재하는 데에 있어서 script 태그의 이용이 제한되어 한계가 생겼습니다.

해결) 백엔드에서 뷰 엔진(pug)을 통해 렌더링 한 뒤 프론트엔드에서 iframe 태그를 이용하여 그것을 불러오게끔 하였습니다.

게임 제작

문제점) iframe의 src를 "server-address/games/tool?uid=USER_PrimaryKey" 형식으로 잡았었는데, 이렇게 할 경우 uid에 다른 유저의 PK값을 입력할 경우 그 유저의 임시저장 목록에 접근할 수 있는 문제가 생겼습니다.

해결) 세션에 15자리 랜덤 코드를 저장한 뒤 그 코드를 iframe의 GET 메소드 변수에 추가시키고, 그것을 백엔드에서 대조하여 맞으면 화면을 띄우게 하였습니다.



- 파라미터값 미일치 시 유튜브로 연결

게임 제작

문제점) iframe에서 데이터를 받아오지 못해서, 게임을 저장할 수 없는 문제가 발생하였습니다.

해결) 백엔드에서 저장 버튼을 누르면 세션에 게임 데이터가 저장이 되고, 프론트엔드에서는 세션에 있는 데이터를 받아오는 형식으로 수정하였습니다.

커밋 & 버전 롤백

문제점) 툴에서 임시저장 버튼을 누를 경우 커밋이 되는 형태로 만들었는데, 처음에는 Blockly의 행동 이벤트를 하나하나 다 저장해버려서 게임 하나를 제작할 경우 DB에 수백 개의 배열값이 저장되게 되었습니다. 임시저장 게임을 불러올 경우 그 이벤트를 하나씩 불러와서 형태를 완성하는 형식이어서, 중간에 하나라도 제대로 저장이 안 되었을 경우 불러오거나 롤백이 아예 안 되는 상황이 발생하였습니다.

커밋 & 버전 롤백

해결) Blockly에서 블록의 덤프를 뜨는 기능이 있다는 걸 알게 되어 이벤트를 저장하는 방식을 블록 형태를 저장하는 방식으로 바꾸었습니다.

이렇게 하여 데이터의 과다 및 로딩에서의 에러를 해결하였습니다.

커뮤니티 기능

- 자신이 제작 중인 게임에 대한 피드백
- 평가를 통한 자체 게시판 생성
- 어드민의 회원 관리 기능

블록 피드백

ROGUEWRITE

- 블록게시판
- 자체 게시판 생성
- 어드민의 회원 관리

자체게시판 생성

문제점) 자체게시판의 url을 설정함에 있어 원래는 게임 이름을 그대로 url에 적용하려고 했으나(게임 이름이 'Hello World'면 url은 'hello-world'), 중복 문제가 생기게 되었습니다. 또한 알파벳과 숫자 이외의 이름도 문제가 발생하였습니다.

해결) 알파벳, 숫자일 경우 그대로 적용하고 아닐 경우 유니코드 값을 추출하여 16으로 나눈 나머지를 16진수로 변환시켜서 적용하였습니다. 중복에 대해서는, 게임 제목의 최대 20글자만 url에 대입시키고 뒤에 3글자의 랜덤 코드를 넣었습니다.

H			
High & Low (imsitest)			
가			
건물 내려가기 (test1)			
하			
학교 졸업하기 (test1)	X	하나빼기 (test1)	X
하하후헤호 (test1)			+
흑시..? (test1)			+

H	
High & Low (imsitest)	
가	
건물 내려가기 (test1)	
하	
학교 졸업하기 (test1)	X
하나빼기 (test1)	X
하하후헤호 (test1)	+
흑시..? (test1)	+

자체게시판 리스트는 PC 화면은 3개씩, 모바일은 세로로 쌓이게 됩니다. 이것을 위해 3*n 형태의 2차원 배열을 만든 뒤, 부트스트랩의 grid 시스템을 이용하여 각각의 아이템들을 쌓게끔 하였습니다.

엔진

- 타입스크립트 언어로 작동

<https://github.com/p9595jh/RogueWrite/blob/master/angular-src/src/app/services/play.service.ts>

- 로깅 기능도 엔진 내에 포함

게임의 구성

게임의 진행 방식 스테이지로 구성되며 스테이지 내부는 페이즈, 페이즈 내부는 초이스, 그리고 초이스 내부는 if-else문들로 이루어져 있습니다. 또한 게임에 필요한 변수들을 따로 선언해주는 파트가 있습니다.

게임의 구성

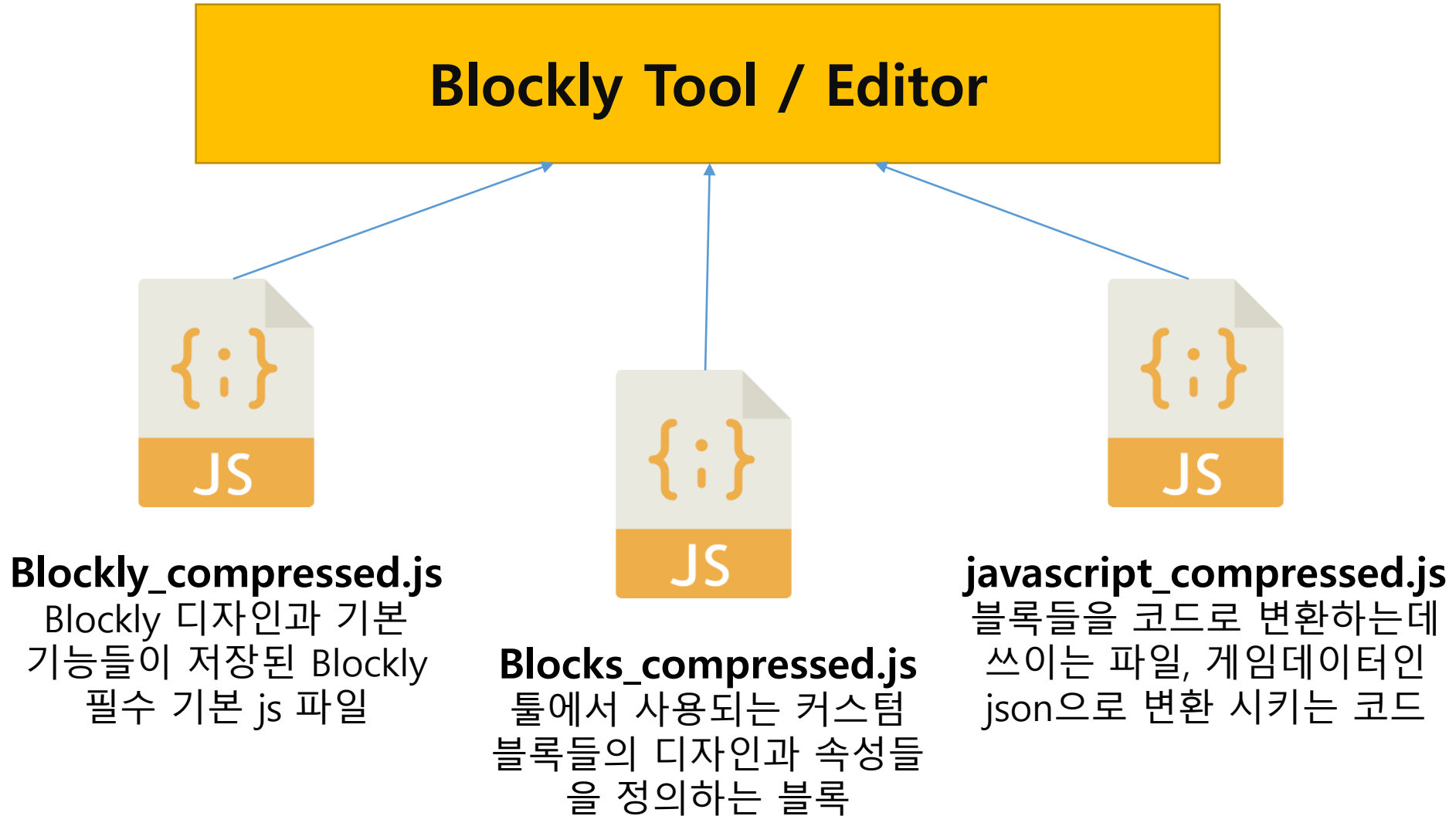
게임이 처음 로드되면 Map에 파라미터 변수들이 저장되고, 첫 스테이지를 찾아서 그 스테이지와 페이즈, 초이스를 저장합니다.

선택지가 선택되면 해당 if-else문에 따라 파라미터 Map의 요소 값들이 변경되고, 다음 스테이지와 해당 페이즈, 초이스를 저장합니다.

게임의 구성

파라미터에 맞춰 해당하는 페이지가 존재하지 않을 경우, noCondition() 함수가 실행되면서 게임이 종료됩니다.

엔딩에 돌입할 경우 end() 함수가 실행되면서 게임이 종료되고, 최종 점수가 계산됩니다. 점수에 따라 최고/최저 플레이어로 DB에 기록됩니다.



툴의 기본적인 사용 방법 (tutorial)

ROGUEWRITE

▲ 클릭하면 튜토리얼 페이지로 이동합니다.

(크롬 브라우저로 연결해야 합니다.)

툴/에디터

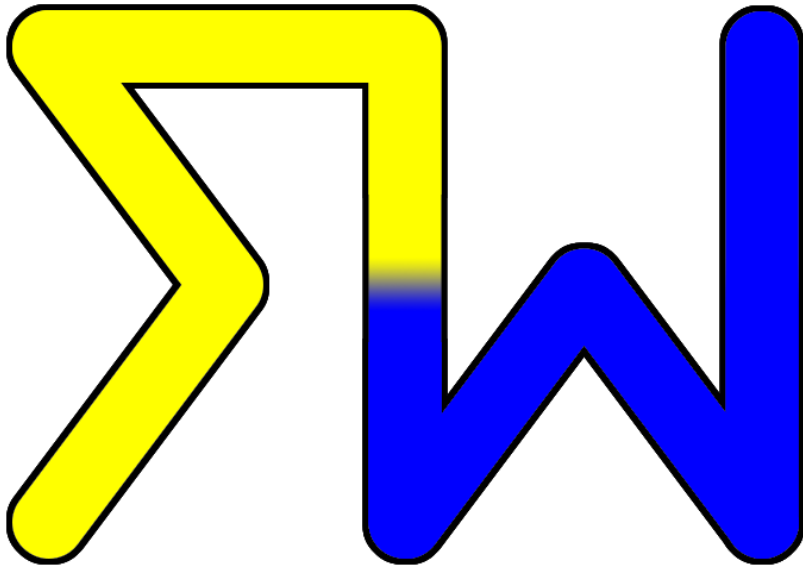
문제점) 실제 게임을 개발하면서 처음 생각 했던 툴의 기능보다 더 다양한 기능의 추가를 필요로 함

해결) 다양한 블록들을 추가하고 엔진의 구동 방식을 바꿈. 추가 된 블록 (조건을 변수가 특정 숫자일때만 -> or & and블록, 비교, 부정 블록 / 선택지 선택 후 변수 변화 -> 선택지 선택 후 상황 별 변수 변화 / 지문 내용은 고정적 -> 변수 내용을 지문에 출력 가능 / 모든 변수는 숫자 -> 문자열 변수도 가능하고 게임 진행중 입력을 받을 수 도 있음

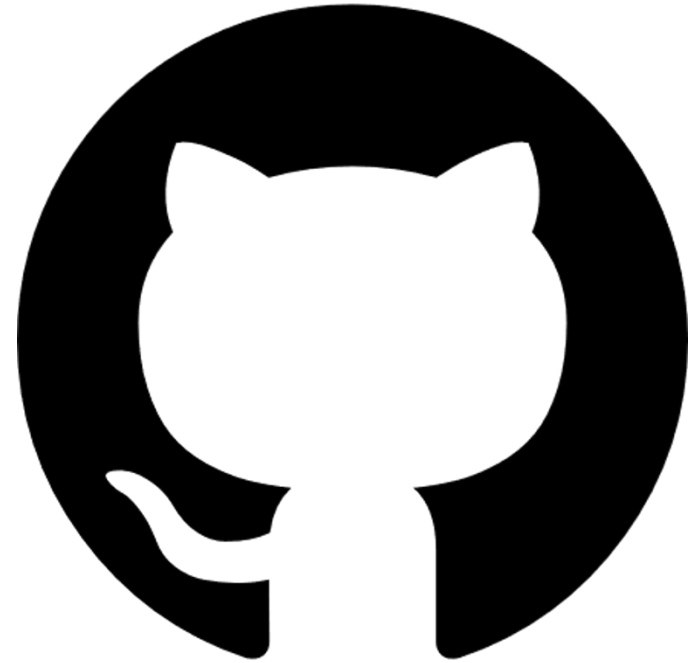
툴/에디터

문제점) 실제 게임을 개발하면서 처음 생각 했던 툴의 기능보다 더 다양한 기능의 추가를 필요로 함

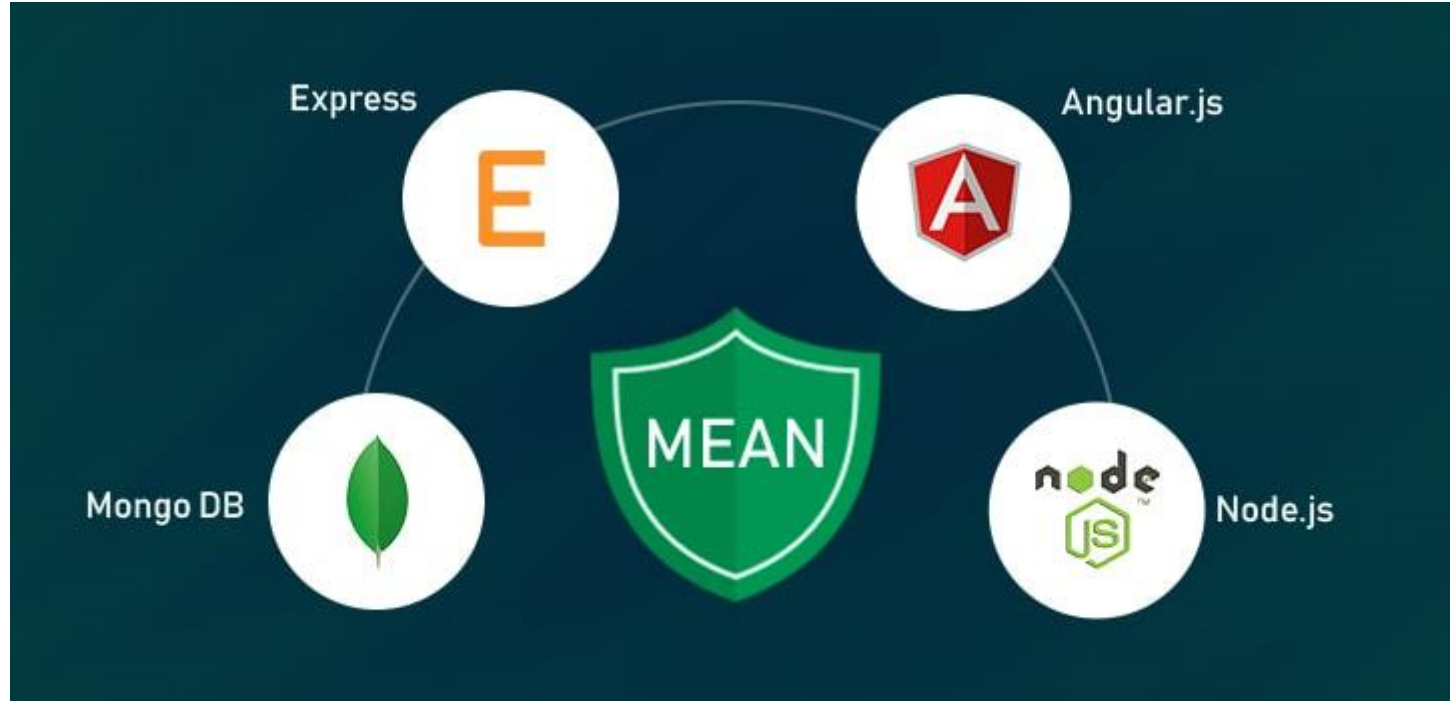
해결) 다양한 블록들을 추가하고 엔진의 구동 방식을 바꿈. 추가 된 블록 (조건을 변수가 특정 숫자일때만 -> or & and블록, 비교, 부정 블록 / 선택지 선택 후 변수 변화 -> 선택지 선택 후 상황 별 변수 변화 / 지문 내용은 고정적 -> 변수 내용을 지문에 출력 가능 / 모든 변수는 숫자 -> 문자열 변수도 가능하고 게임 진행중 입력을 받을 수 도 있음



RogueWrite Site (only Chrome Browser)



RogueWrite Github



MEAN stack



Blockly : 구글에서 제공하는 오픈소스로, 직관적인 그래픽 환경을 제공함.
다양한 프로그래밍 언어로 변환 가능한 점을 이용해 툴 개발에 사용.



프론트엔드

Angular



백엔드

Node.js & Express



DB

MongoDB



Bootstrap 4

CSS

Google Material Design

Twitter Bootstrap



아쉬운 점

웹을 개발함에 있어서, 동학기 웹프레임워크2 수업을 들으면서 RESTful API에 대해 처음 배웠습니다. 그에 따라 프론트엔드/백엔드를 나눠서 개발하면서도 REST 설계규칙을 따르지 않았다는 점이 아쉬웠습니다. 그리고, 지난 학기에 Angular를 처음 써보고 두 번째로 쓰면서 나름 많이 늘었다고 생각했음에도 개발이 끝나고 보니 아쉬운 점들이 제법 있었습니다. Angular가 꽤나 강력한 프론트엔드임에도 전부 활용하지 못한 점이 살짝 아쉽습니다. 마지막으로, 크롬으로만 디버깅 하여 그 외의 브라우저와의 호환이 잘 안 된다는 점이 아쉽습니다.

느낀 점

구상부터 설계, 최종 발표까지 팀원과 함께 하면서 꽤나 즐거웠습니다. 앞으로 개발자로 살아가면서 원하는 코딩을 할 일이 얼마나 있을지 모르겠지만, 시간과 예산을 들여 이렇게 원하는 코딩을 한다는 경험이 꽤나 좋았습니다. 팀으로 활동을 하면서 의견이 항상 합치하는 것은 아니어서 그런 상황들을 해결하는 것이 단체로 무언가를 함에 있어서 일어날 수 있는 일들이라는 것을 느꼈습니다. 처음 시작할 때는 막막함이 다소 있었지만, 팀원과 함께 의견을 공유하고 지도교수님께 첨언을 받아가면서 프로젝트가 하나둘씩 진행되는 것을 보면서 나름의 흐뭇함도 느꼈습니다. 만약 기회가 된다면 다시금 이런 프로젝트를 해 보고 싶습니다.



아쉬운 점

이번 프로젝트의 에디터로 blockly를 썼는데 게임 툴이라는 목적으로 만들었기에 blockly가 구현 가능한 실제 기능의 2할밖에 못쓴 것 같아서 아쉬웠습니다.
또 게임을 만들 때 다양한 기능을 이것저것 추가해서 게임 제작의 볼륨을 늘리고 싶었지만 아이디어나 시간들이 짧아 더 구현을 못한 것과 안드로이드를 2주정도 개발하다가 치명적인 문제점이 발견돼서 개발이 중지되었는데, 다른 방법의 개발이 확립되면 앱의 연동도 다시 해보고 싶습니다.

느낀 점

처음 에디터를 만든다고 했을 때 지준 교수님이 Blockly라는 API를 추천해주셨습니다. 첫 날 프로젝트 구상할 때 툴을 만든다고만 구상하고 어떻게 만들지는 생각을 하지 못했는데 교수님이 바로 해답을 주셨기에 저도 관심을 가지고 Blockly에 대해 알아 보려고 했지만 한글로 된 자료는 매우 희박했습니다. Blockly를 사용한 여러 툴은 이미 써본 기억은 있었습니다. Scratch라든가 App Inventor 같은 다른 사람들이 만든 Blockly기반의 프로그램들을 여럿 써본 경험을 생각해보니 이만한 툴 제작 API가 없다고 생각했습니다. 쉬울 것이라고 생각을 했지만 막상 하려고 보니 자세한 정보는 대부분 영어로 되어있고, 구글의 설명 시트에도 자세히 적혀있지 않았습니다. 직접 코드를 뜯고 이것저것 바꾸고 저장 해나가면서 Blockly의 반 이상은 이해하고 게임의 툴이라는 목적에 맞춰 이것저것 수정해서 이번 프로젝트의 툴을 만들었습니다. Blockly를 사용해서 툴을 만들 때 마다 "이건 정말 범용성 있고, 포텐이 있다"라고 생각이 들었고 나중에 기회가 된다면 제대로 해보고 싶다고 생각이 들 정도로 저에게 크게 다가왔습니다. 이번 프로젝트로 또 하나 공부가 되었고, 나중에 기회가 된다면 지금 팀으로 다시 한 번 이것 저것 시도해 보고 싶습니다.