



README

1 Description

게임 소개

최대 6인 플레이까지 지원하는 하이퍼 캐주얼 레이싱 게임입니다.

나만의 자동차를 타고 잘 포장된 도심과 험준한 자연 속을 누비며 먼저 결승선에 골인하기 위해 플레이어들과 경쟁합니다. 맵에서 얻은 랜덤한 아이템을 사용해 변수를 창출하고, 전략적인 주행으로 승리를 쟁취해 냈을 때의 성취감과 재미, 그 과정에서 발생하는 다양한 긴장감을 선사합니다.

개발 배경

팀원들과 함께 스트레스에 대해 이야기 하다가 가장 많은 공감을 얻었던 해소 방법이 드라이브였습니다. 자동차를 운전하며 다양한 풍경을 감상하고, 뻥 뚫린 도로를 달릴때면 자연스레 쌓였던 답답함이 날아갑니다. 하지만, 현실의 도로에서는 도로 위의 많은 차량과 교통법규로 제약이 있습니다.

이러한 점을 바탕으로 게임이라는 가상 환경에서 자유롭게 드라이빙 하는 컨텐츠를 구상했습니다. 여기에 단순한 드라이빙이 아닌 경쟁의 재미를 더하고자 했고, 아이템을 사용해 다양한 변수를 만들어 내는 방법을 선택하게 되었습니다.

기능 명세서

* 프로젝트 진행을 위해 분석된 요구사항과 기능 정의를 아래 표에 작성합니다.

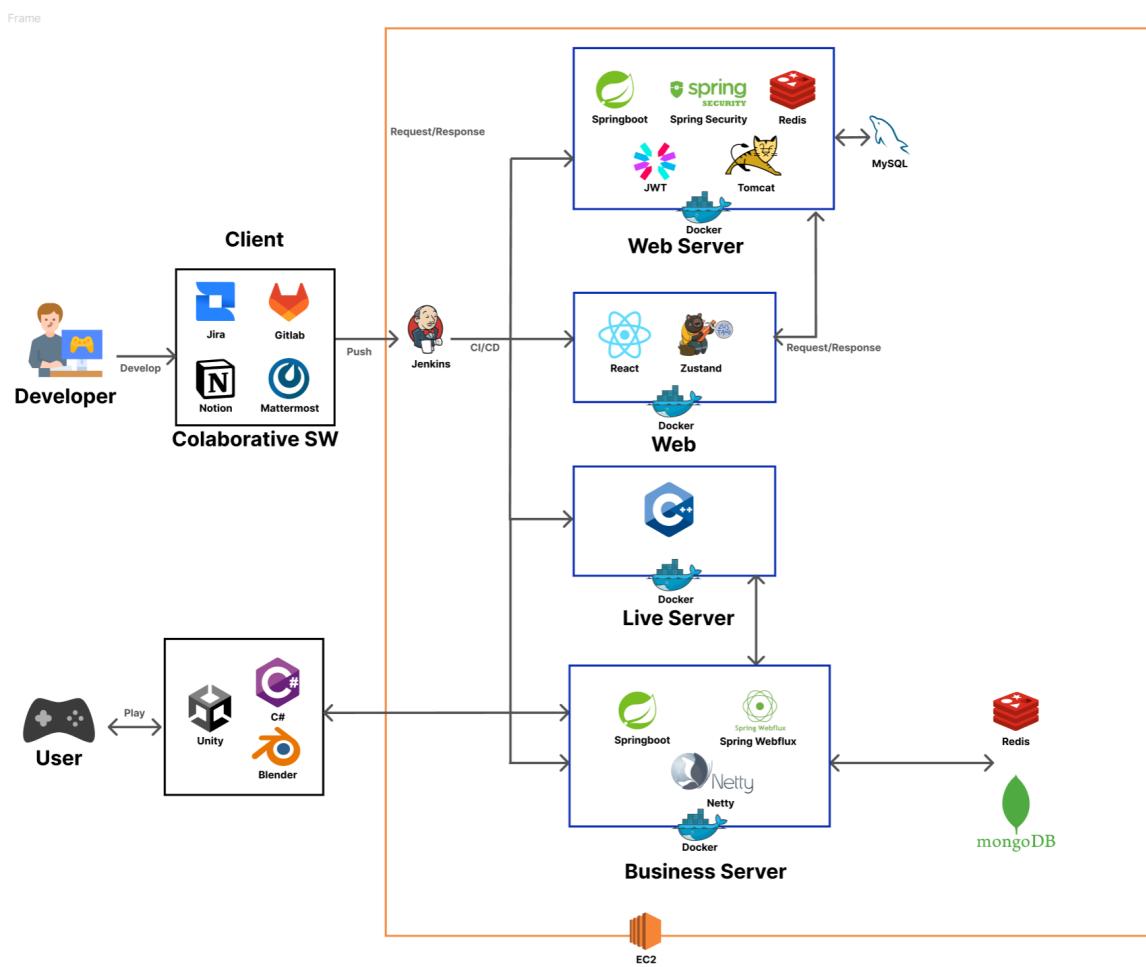
ID	요구사항	기능 정의
Req. 1.	회원 관리	회원 이메일 주소(수정 불가)와 닉네임을 등록/수정/삭제
Req. 2.	로그인/로그아웃	회원의 웹사이트 이용을 위한 로그인/로그아웃 기능을 구현
Req. 3.	게임 방 생성	게임 플레이를 위한 방을 생성
Req. 3-1.	게임 방 참가	생성된 게임 방에 참가

ID	요구사항	기능 정의
Req. 3-2.	게임 방 나가기	로비로 이동 및 방장 권한 인가 및 방에 남은 인원이 없을 경우 방 제거
Req. 4-1.	게임 초대	게임 방에 사용자를 초대
Req. 4-2.	캐릭터 선택	레이싱에서 사용할 캐릭터를 선택
Req. 4-3.	카트 선택	레이싱에서 사용할 카트를 선택
Req. 4-4.	게임 시작	게임을 시작
Req. 5.	아이템 랜덤 생성	맵의 특정 장소에 아이템이 생성 및 리스폰
Req. 5-1.	카트 운행	레이싱 중 카트의 운행을 조작
Req. 5-2.	아이템 획득	레이싱 중 아이템을 획득
Req. 5-3.	아이템 이용	획득한 아이템을 사용
Req. 6.	게임 종료	최초로 결승점을 통과할 경우 현재 게임의 남은 시간을 10초로 설정하여 게임 종료 예고
Req. 6-1.	점수 종합	각 유저들의 결승점 통과 기록을 산출하여 랭킹에 반영
Req. 6-2.	로비로 이동	게임 종료 이후 모든 유저들을 로비로 이동

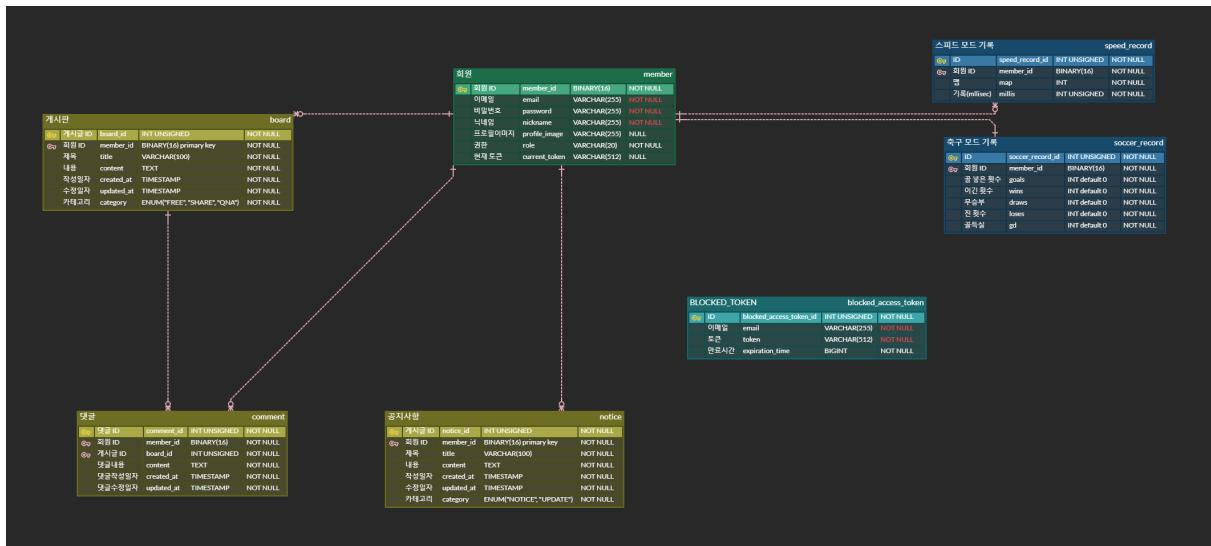
활용 언어 및 주요 주요 기술

- 활용 언어
 - C++
 - C#
 - Java
 - JavaScript(Typescript)
- 주요 기술
 - Message Pack
 - Unity
 - Spring Boot
 - **추가하기**

프로젝트 아키텍처 (파이어베이스 추가 필요)



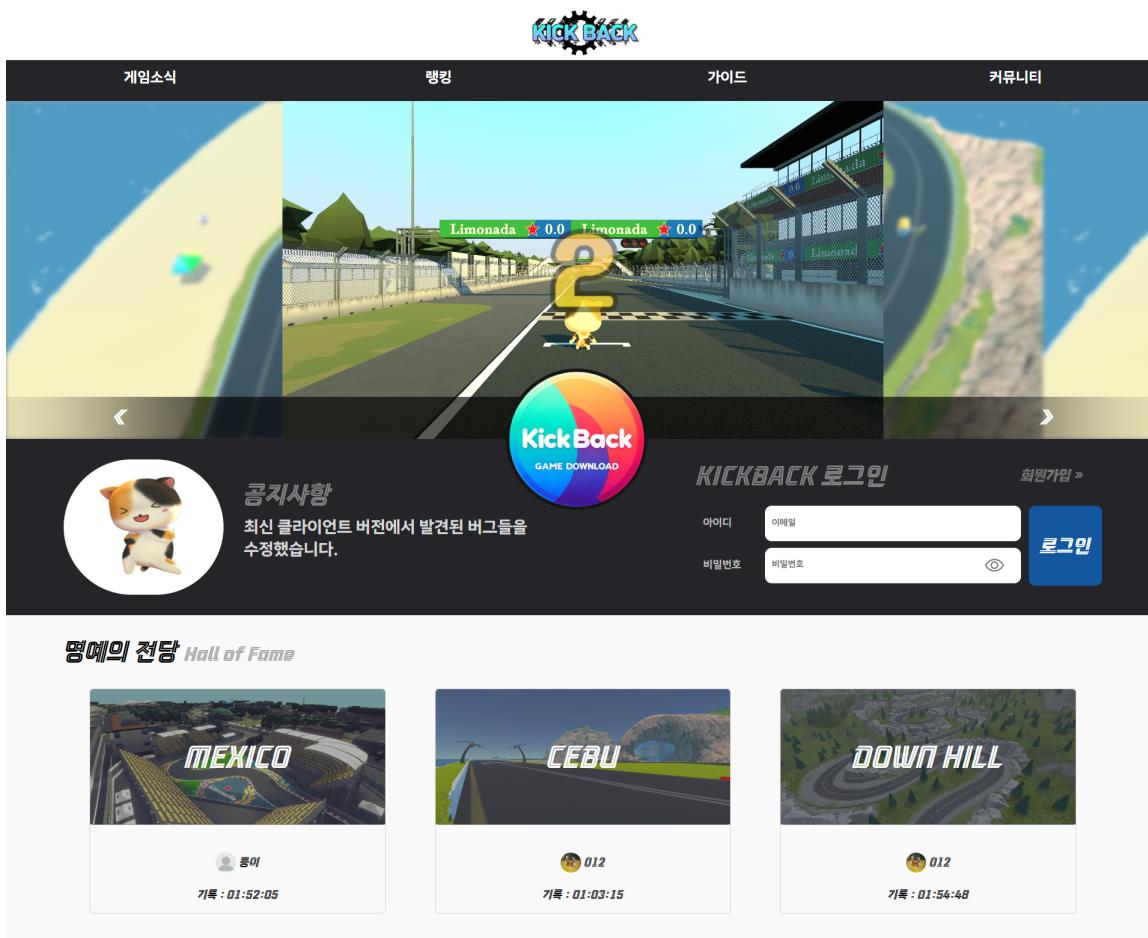
ERD



2 FRONT-END

FRONT-END 파일 구조

▼ 메인 페이지



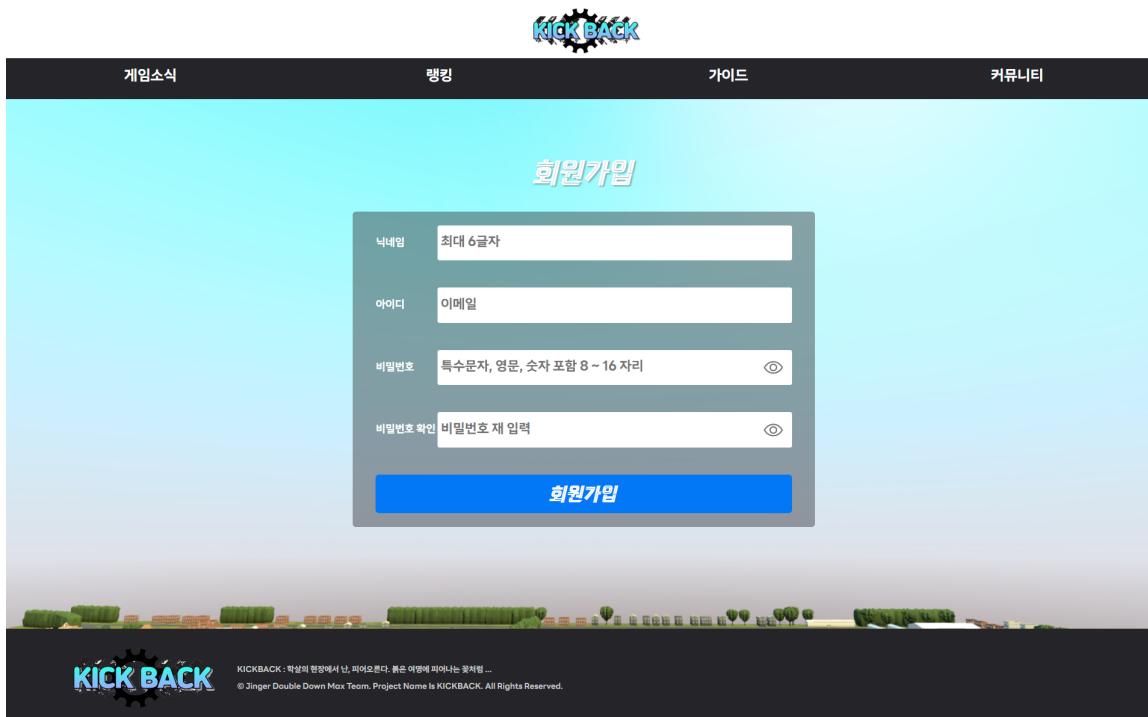
모드 소개 Mode Introduction



KICK BACK

KICKBACK : 학살의 현장에서 난, 피어오른다. 뿐은 이영에 피어나는 꽃처럼 ...
© Jinger Double Down Max Team. Project Name is KICKBACK. All Rights Reserved.

▼ 회원가입 페이지



▼ 랭킹 페이지

The screenshot shows the ranking page for the KICKBACK website. The top navigation bar includes '게임소식' (Game News), '랭킹' (Ranking), '가이드' (Guide), and '커뮤니티' (Community). The main header features the KICKBACK logo and the text 'KICKBACK : 학살의 현장에서 난, 피어오른다. 뽈은 여행에 피어나는 꽃처럼 ... © Jinger Double Down Max Team. Project Name Is KICKBACK. All Rights Reserved.' Below the header, there's a large banner with a yellow cat-like character and the text '스피드 랭킹' (Speed Ranking), '1st Me', and a timer '01:23:64'. The page displays several tabs: 'MEXICO' (highlighted in yellow), 'CEBU', 'DOWNHILL', and 'BORYEONG'. A search bar with placeholder '닉네임을 검색해보세요!' (Search nickname) and a '검색' (Search) button are located below the tabs. Two cards are shown: one for a player named '012' with a corgi profile picture, and another for a player named '메룡' showing a stadium view. A table lists the top three players with their names, icons, and times: 1st place is '동이' (01:52:05), 2nd place is '012' (02:02:22), and 3rd place is '메룡' (03:35:06). At the bottom, there are navigation arrows for the table. The footer contains the KICKBACK logo and the same copyright information as the top header.

3 WEB SERVER

소개

WEB SERVER 파일 구조

- Spring Security와 JWT 토큰을 사용하여 인증 및 인가 시스템을 구현했습니다.
- Redis를 활용하여 refreshToken을 저장하고 빠른 조회를 가능하게 구현했습니다.
- Jasypt를 통해 환경 변수를 암호화하여 안전하게 관리하였습니다.
- 클라우드 스토리지(파이어베이스)를 사용하여 서버 저장소 부담을 감소시켰습니다.
- JPA 및 @Query 어노테이션을 사용하여 유연한 데이터베이스 접근을 구현하였습니다.

인증/인가

spring security, jwt 토큰 사용

jwt 토큰을 활용

redis를 활용하여 accesstoken 재 발급 요청에 대한 빠른 데이터 조회 가능

중복 로그인 방지

jasypt 활용한 환경 변수 암호화

+> vault ...?

비동기 통신 활용 (async function)

JPA 및 @Query 활용 메서드 유연하게...?

파이어베이스 활용, 서버 저장소 부담 감소

4 BUSINESS SERVER

BUSINESS SERVER 파일 구조

📌 HOW TO

Netty

- 자바 네트워크 애플리케이션 프레임워크
- 비동기 이벤트 기반 네트워크 응용프로그램 프레임워크

Reactor Pattern으로 비동기 Non-Blocking TCP 서버 구현

ChannelPipeline에 Handler 등록

- LineBasedFrameDecoder
- 사용자 정의 ChannelHandlerAdapter

메서드	설명
handlerAdded	client와 최초로 연결되었을 경우 - logging
handlerRemoved	client와의 연결이 끊겼을 경우 - 연결이 끊어진 session 삭제 - logging
exceptionCaught	예외 처리 - logging

💿 데이터 처리 방법

- MessagePack(네트워크 통신 데이터 직렬화 라이브러리) 사용

- 송신 데이터 **MessageBufferPacker**

```
MessageBufferPacker packer = MessagePack.newDefaultBufferPacker();
byte[] bytes;
packer.packArrayHeader(2);      // 배열의 Header 지정
packer.packString("userList"); // 문자열 - packString , packObject
packer.packString(String.valueOf(channel.getSessionName()));
bytes = packer.toByteArray();  // 바이트 배열로 변환후 송신
```

- 수신 데이터 **MessageUnpacker**

```

MessageUnpacker unpacker = MessagePack.newDefaultUnpacker();

int arrayLength = unpacker.unpackArrayHeader(); // 배열
Type type = Type.findIndex(unpacker.unpackInt()); // 타입
String userName = unpacker.unpackString(); // 문자

```

- Type 지정 ENUM

```

public enum Type {
    LIVESERVER( idx: 0 ), // 최초 연결 (라이브 서버)
    CLIENT( idx: 1 ), // 최초 연결 (유니티)
    CREATE( idx: 2 ), // 방 생성
    JOIN( idx: 3 ), // 방 참가
    LEAVE( idx: 4 ), // 방 나가기
    READY( idx: 5 ), // 게임 준비
    START( idx: 6 ), // 게임 시작
    ITEM( idx: 7 ), // 아이템 사용
    END( idx: 8 ), // 게임 끝
    Map( idx: 9 ), // 맵 바꾸기
    POSITION( idx: 10 ),
    SPOSITION( idx: 11 ),
    TEAMCHANGE( idx: 12 ), // 팀 바꾸기
    CHARCHANGE( idx: 13 ); // 캐릭터 바꾸기

    private final int idx;

    Type(int idx) { this.idx = idx; }

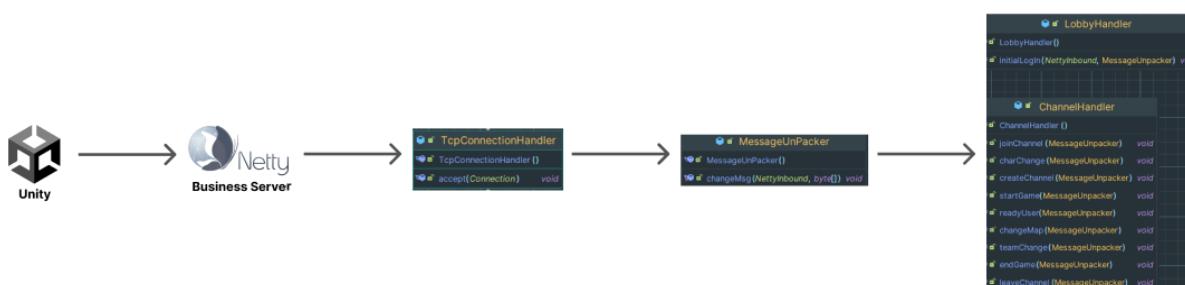
    public static Type findByIndex(int index) {
        for (Type type : Type.values()) {
            if (type.idx == index) {
                return type;
            }
        }
        throw new IllegalArgumentException("Unknown index: " + index);
    }
}

```

CLIENT - BUSINESS SERVER 데이터 흐름

① LOBBY & CHANNEL

- Unity 로그인 후 Business Server 최초접속
- Business Server에서 Type 확인 후 case 별 handler 호출
- 데이터 확인 후 각 Channel에 Broadcasting



5 CHATTING SERVER

CHATTING SERVER 파일 구조

써주세요

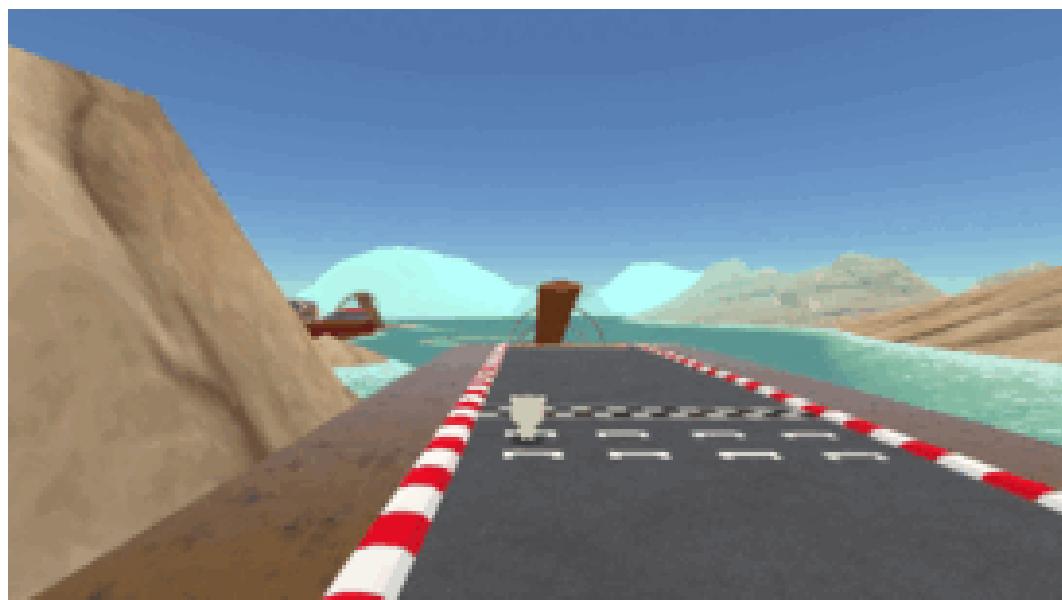
6 LIVE SERVER

써주세요

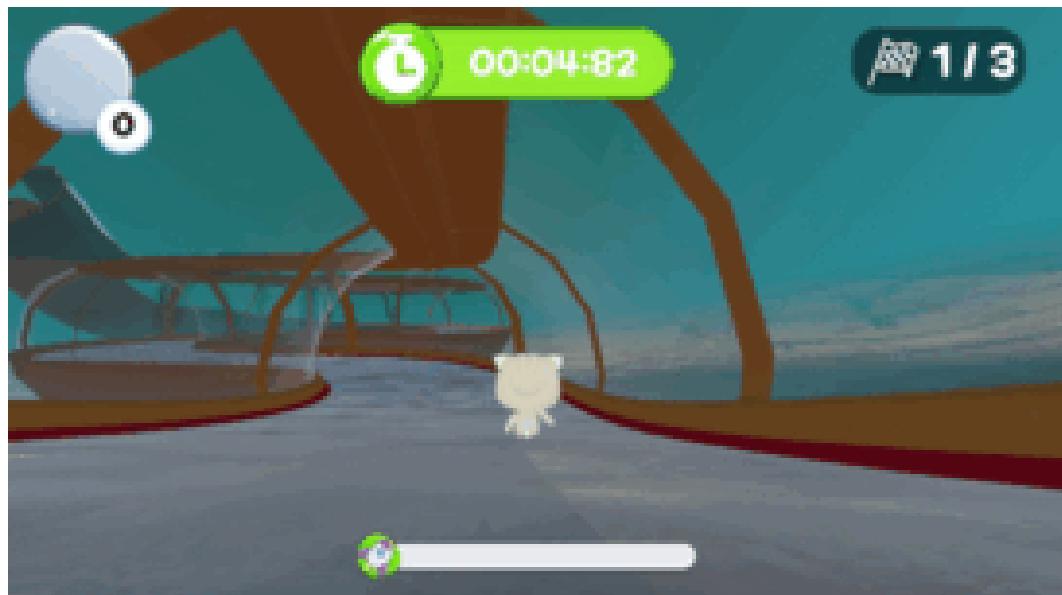
7 UNITY

- 회원 가입
- 로그인
- 로비
- 방 만들기
- 대기방

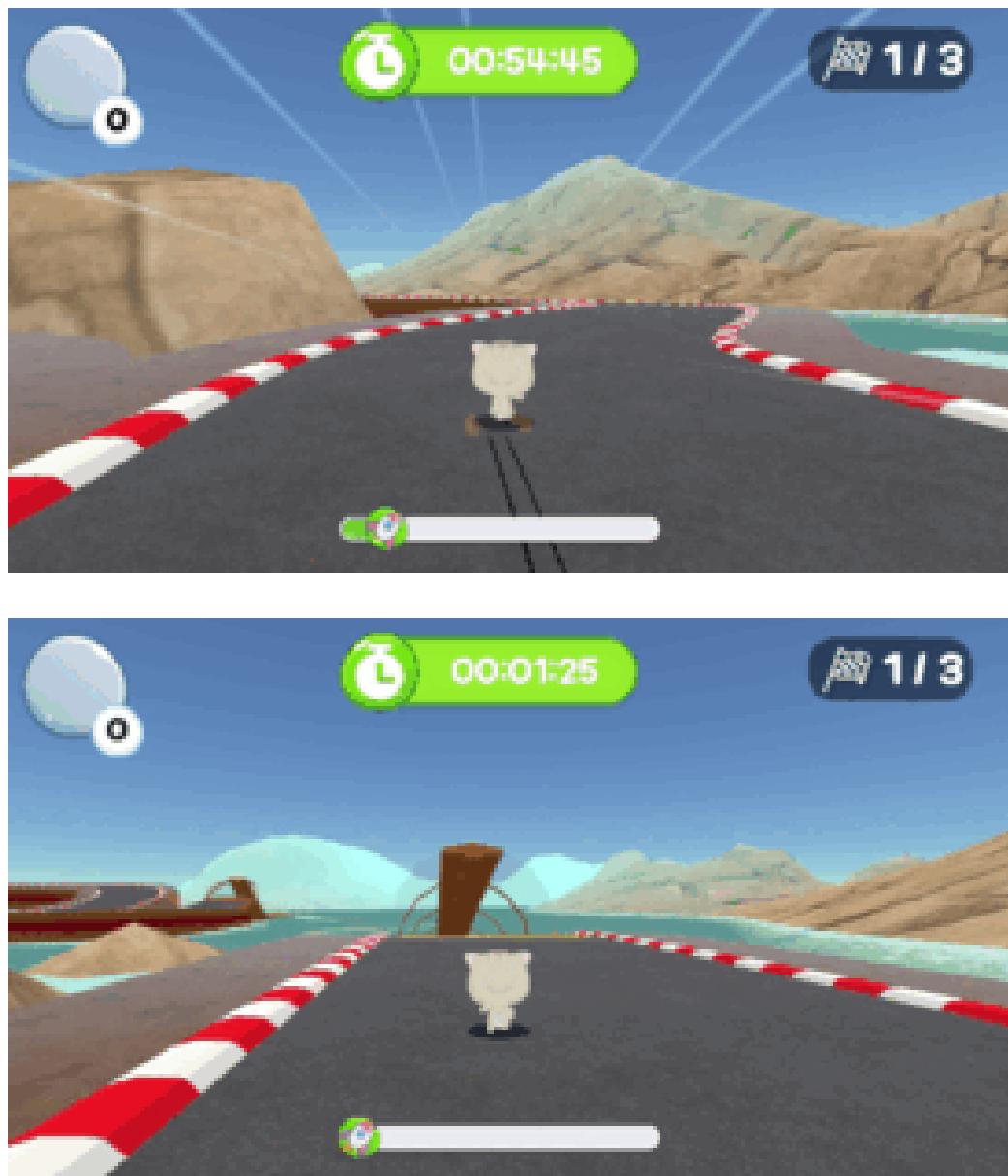
- PC 버전
 - 인트로



- 게임 시작 시 Main Camera의 Transform을 변환 시켜 게임 인트로 및 카운트다운 구현
- 인트로 및 카운트다운 진행 중 움직이지 못하도록 제한
 - 게임 플레이



- 키보드 버튼 클릭에 따라 Horizontal, Vertical 값을 Raw 형태로 받아 -1, 1로 입력값으로 받음
- Rigidbody를 통해 AddForce 등을 통해 Transform 변화를 통해 움직임 구현
- Left Shift를 눌렀을 때 Drift 발동, Player의 회전 속도를 상승 시켜 빠른 회전을 통해 드리프트 효과 구현

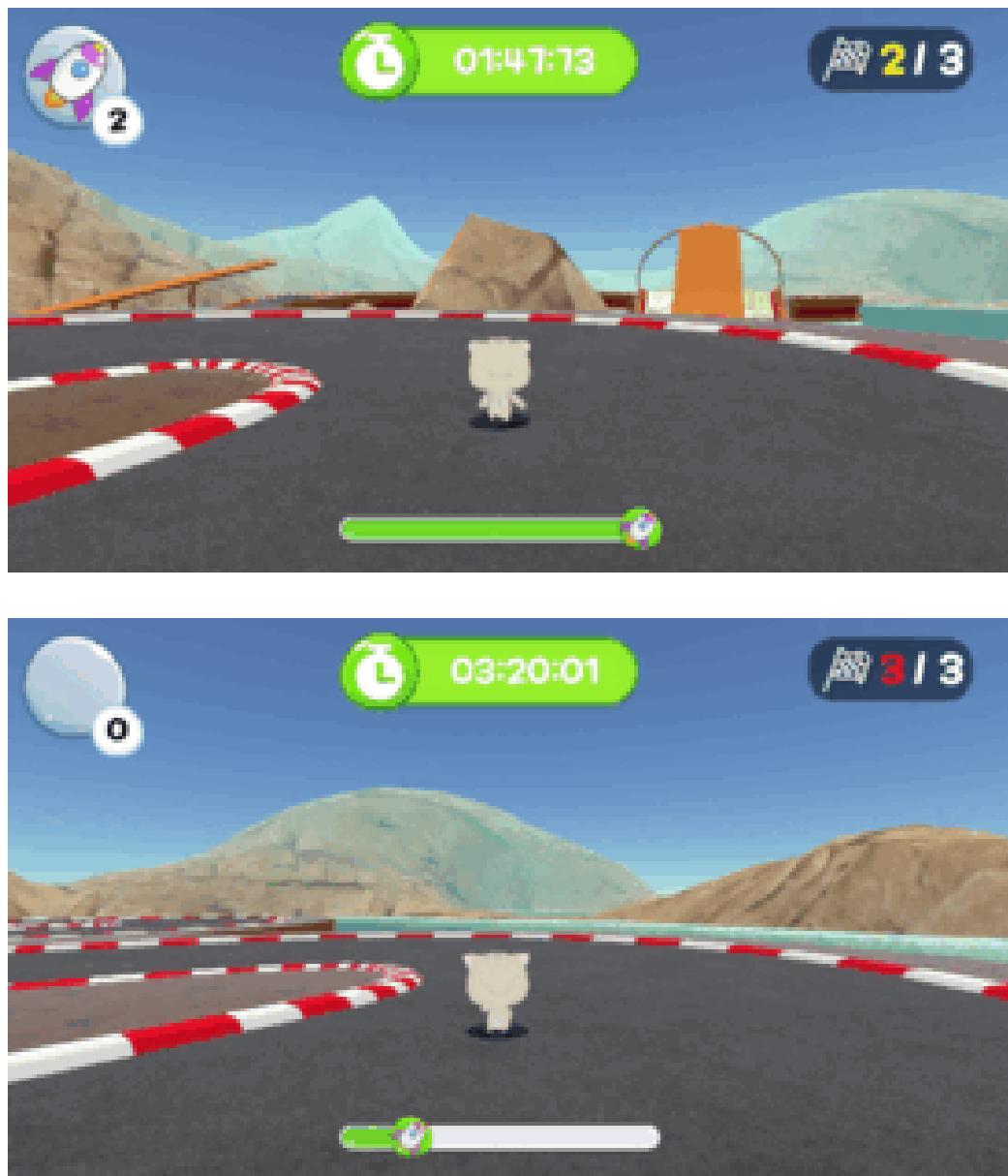


- 드리프트 시 부스터 게이지 충전
- 드리프트 시 TrailRender 생성을 통해 스키드 마크 구현
- 부스터 최대 2개 까지 충전
- 부스터 사용 동안 중첩 부스터 사용 제한
- 부스터 사용 시 ParticleSystem 재생, 불꽃 효과 및 스피드 라인을 통해 속도 감 있는 부스터 구현
- OnTrigger 이벤트를 통해 부스터 패드를 밟았을 시에도 부스터 발동

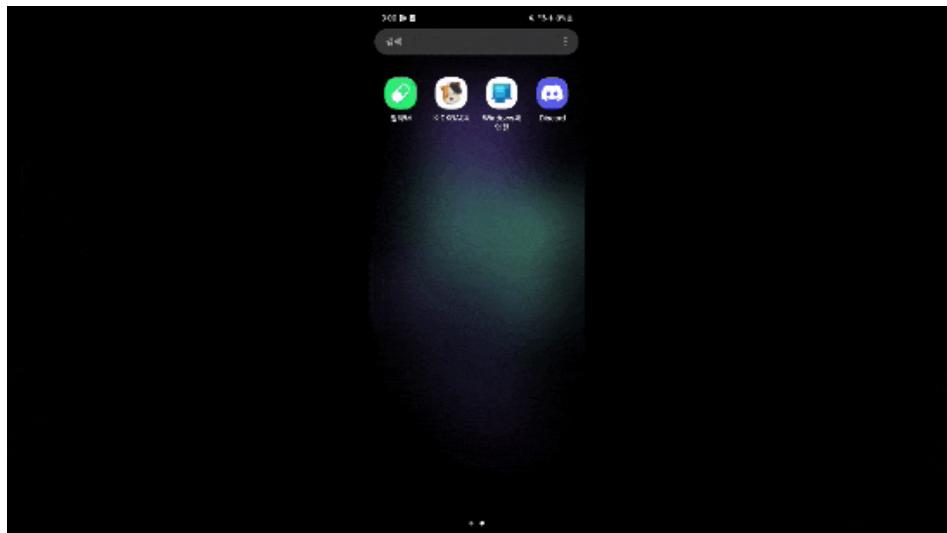


- RayCast를 통해 땅 감지 및 점프 기능 구현





- 레벨 디자인에 투명한 체크 포인트 배치
 - 체크 포인트를 지날 때 리스폰 Transform 초기화
 - 맵에 끼거나 맵 밖으로 나가졌을 때 데드존에 떨어졌을 때 리스폰
 - 각 체크 포인트 및 데드존은 OnTrigger 이벤트를 통해 초기화
 - 체크 포인트를 순서대로 통과했을 때 랩 측정
 - 올바르게 체크 포인트를 통과하지 못할 경우 경고 메세지 출력 후 강제 리스폰
 - 각 맵 별 정해진 랩을 모두 통과 했을 시 게임 종료
- 안드로이드 버전
 - 앱 다운로드 후 실행

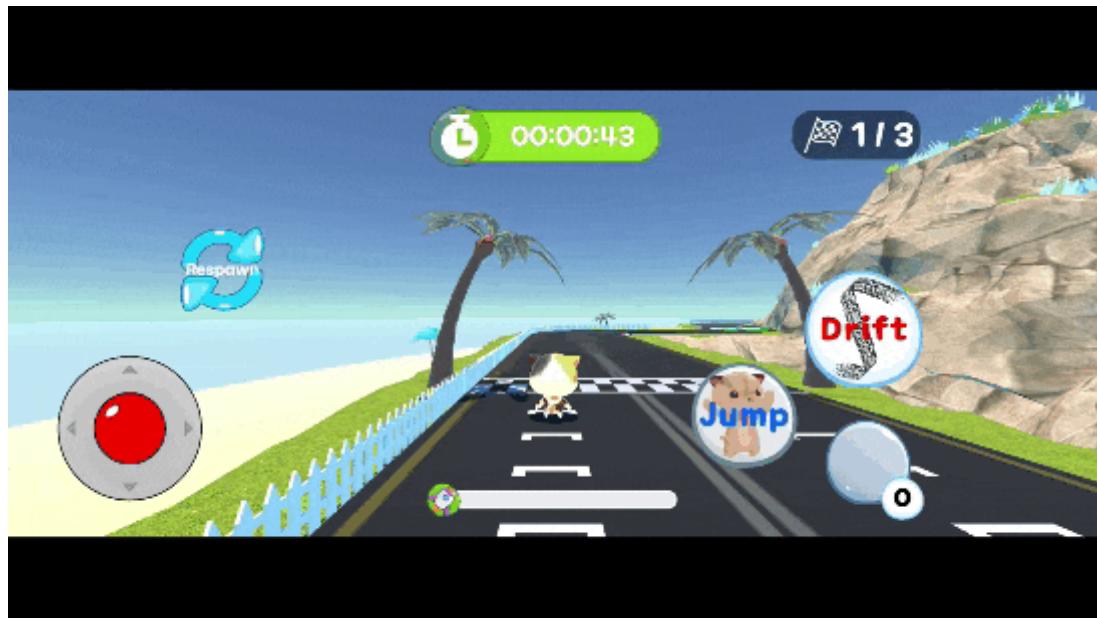


- 인트로

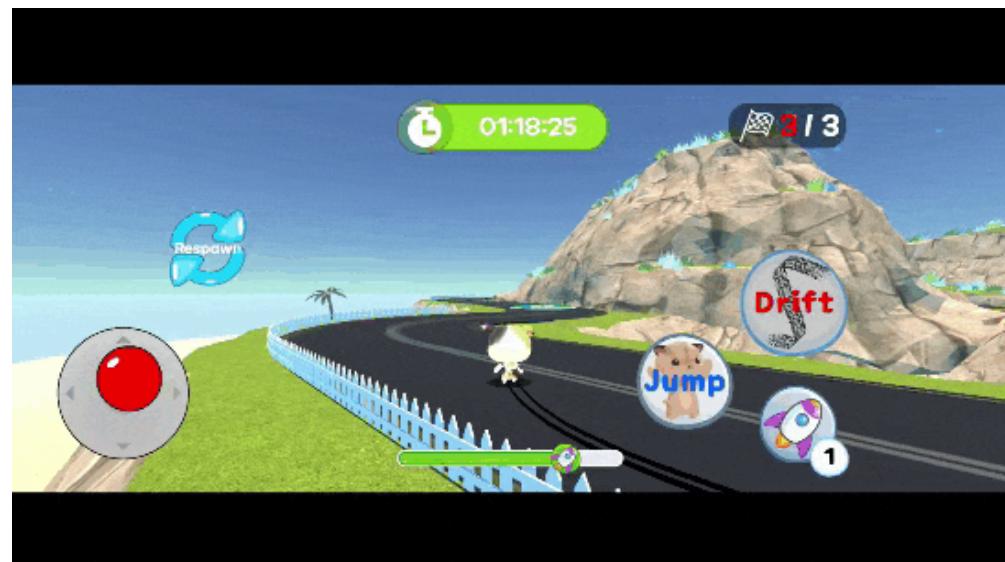


- 게임 시작 시 Main Camera의 Transform을 변환 시켜 게임 인트로 및 카운트 다운 구현
- 인트로 및 카운트 다운 진행 중 움직이지 못하도록 제한

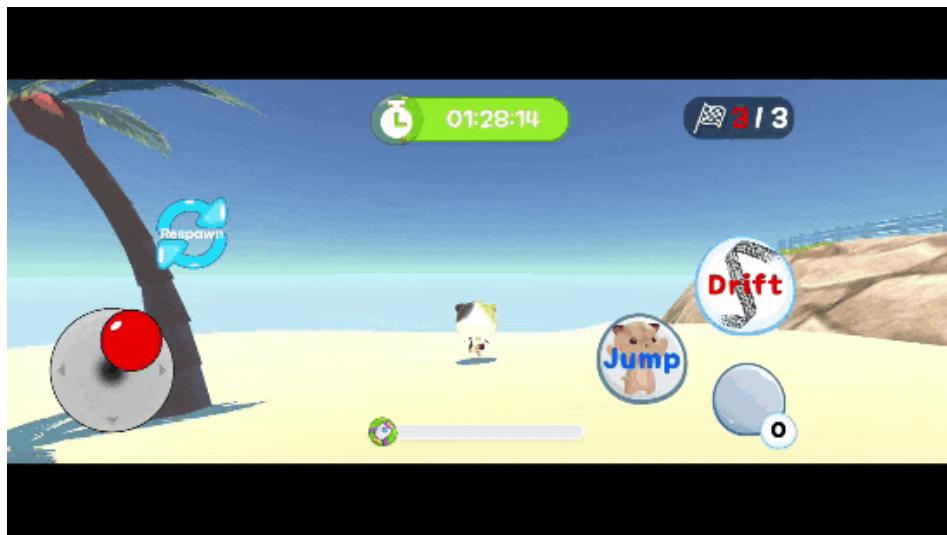
- 게임 플레이



- 조이스틱의 움직임에 따라 Horizontal, Vertical 값을 Raw 형태로 받아 -1, 1로 입력값으로 받음
- Rigidbody를 통해 AddForce 등을 통해 Transform 변화를 통해 움직임 구현
- 각각의 기능들을 OnClick 혹은 Event Trigger를 통해 버튼에 담아 버튼 클릭 시 기능이 구현
- Drift를 눌렀을 때 Player의 회전 속도를 상승 시켜 빠른 회전을 통해 드리프트 효과 구현
- 드리프트 시 부스터 게이지 충전
- 드리프트 시 TrailRender 생성을 통해 스키드 마크 구현

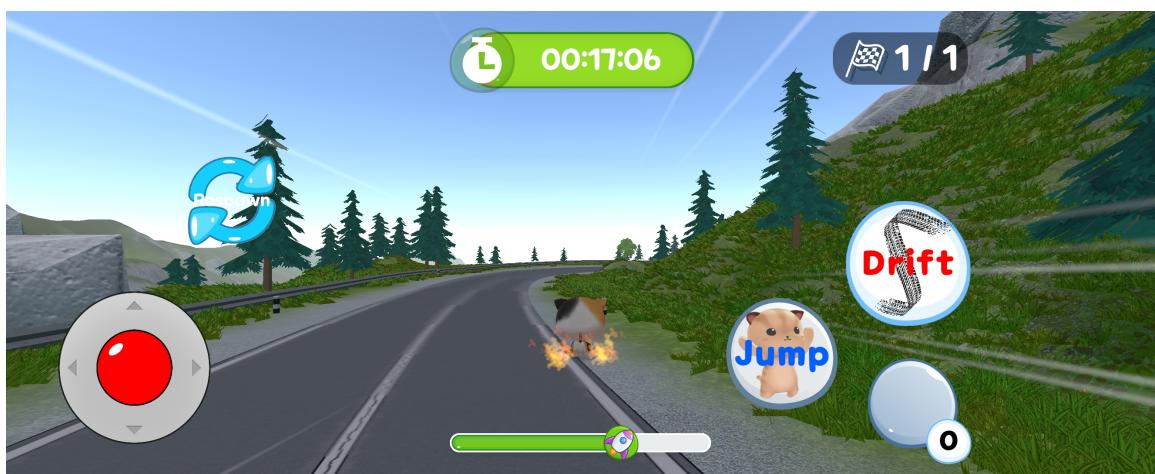
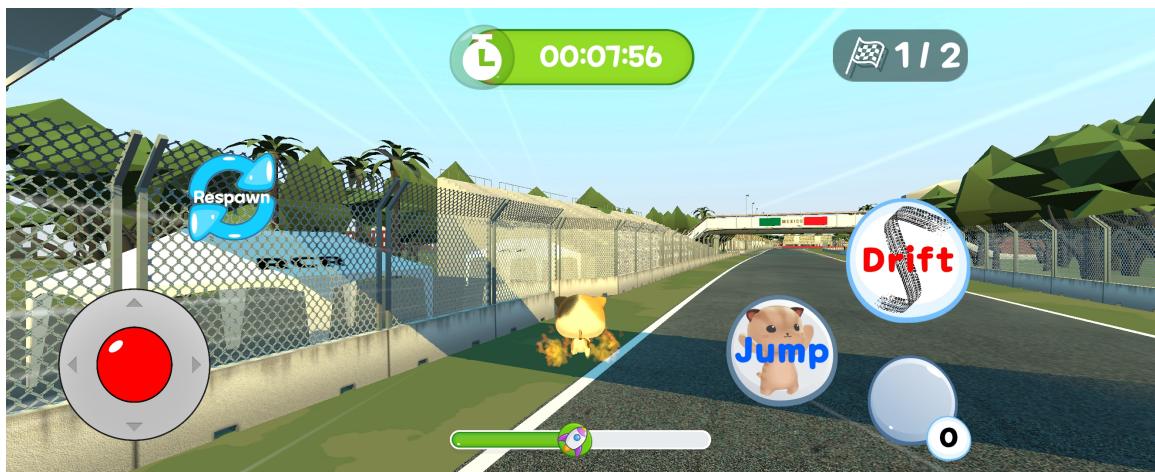


- 부스터 최대 2개 까지 충전
- 부스터 사용 동안 중첩 부스터 사용 제한
- 부스터 사용 시 ParticleSystem 재생, 불꽃 효과 및 스피드 라인을 통해 속도감 있는 부스터 구현
- OnTrigger 이벤트를 통해 부스터 패드를 밟았을 시에도 부스터 발동



- 다양한 장애물 배치를 통해 색다른 레이싱 구현

- RayCast를 통해 땅 감지 및 점프 기능 구현
- 레벨 디자인에 투명한 체크 포인트 배치
- 체크 포인트를 지날 때 리스폰 Transform 초기화
- 맵에 끼거나 맵 밖으로 나가졌을 때 데드존에 떨어졌을 때 리스폰
- 각 체크 포인트 및 데드존은 OnTrigger 이벤트를 통해 초기화
- 체크 포인트를 모두 통과했을 때 랩 측정
- 각 맵 별 정해진 랩을 모두 통과 했을 시 게임 종료



- 다양한 맵 구성
- 비탈길에서 Physics Raycast를 통해 지면과 붙어 있는지 확인 후 자연스러운 움직임 구현

- 축구 모드
- 랭킹 페이지

랭킹 시스템

- 맵 별 랩타임 기록



MEXICO CEBU DOWNHILL BORYEONG

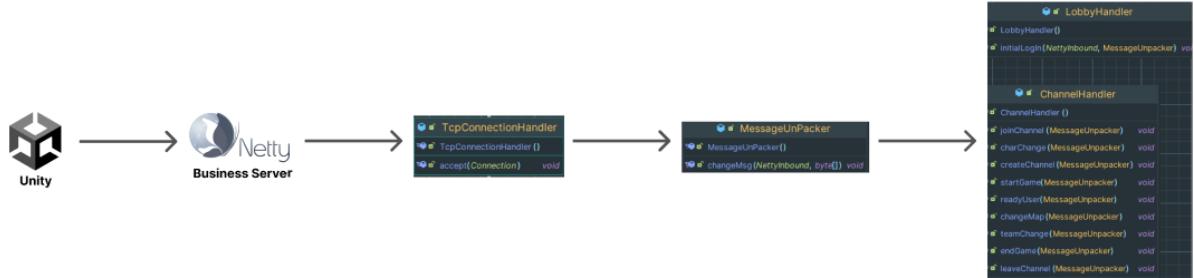
닉네임을 검색해보세요!

검색

닉네임 : 012
랭킹 : 1
기록 : 01:03:15

등수	닉네임	기록
	012	01:03:15
	qq	01:04:25

- 게임 종료 후 자신이 기록한 랭 타임은 HTTP 프로토콜 통신을 통해 requestUrl에 PUT 요청 보냄
- 갱신된 랭 타임은 웹페이지 랭킹 페이지에서 확인



8 팀원 별 담당 역할 및 업무

이름	담당 주 역할 및 업무	부 역할 및 업무
김선욱	Unity 인게임 개발	인게임 데이터 관리
김영일	비즈니스 서버 개발	Unity 인게임 개발
서민주	웹 백엔드 개발	웹 프론트엔드 개발
이항우	서버 개발, Unity	CI/CD, 데이터 관리
장동재	브로드캐스트 서버 개발	비즈니스 서버 개발
조현호	웹(인증) 서버 개발	서버 모니터링 및 관리