

# 포트폴리오

김주리

<https://github.com/zlxlnfl>

## CONTENTS

1

자기소개

2

진행 프로젝트

3

맺음말

## 자기소개

- 광운대학교 미디어영상학부/컴퓨터소프트웨어학과 졸업
- <기술스택>
  - 운영체제 : Windows / Linux Ubuntu
  - 사용 언어 : C / C++ / JAVA / Kotlin / Javascript / Python
  - 프레임워크 : SpringBoot(MVC, Webflux, Cloud) / Android Studio
  - 데이터베이스 : MySQL / MongoDB / Redis
  - 컨테이너 환경 : Docker / Kubernetes
  - 게임 엔진 : Unity
  - 그 외 : RabbitMQ / Hibernate 등
- <GIT 주소>  
<https://github.com/zlxlwnfl>

# TypingGameWorld

■ React.js/SpringBoot Webflux/MongoDB MSA 타자게임 웹사이트 (제작중)

■ <제작자>

김주리

■ <기술스택>

운영체제 : Windows 10 / Linux Ubuntu(Docker)(예정)

프로그래밍 언어 : Kotlin / JavaScript

프론트엔드 라이브러리 : React.js

백엔드 프레임워크 : SpringBoot 2.4.2 / Spring Webflux(Netty)

MSA 시스템 : Istio(예정)

데이터베이스: MongoDB

기타 : Bootstrap

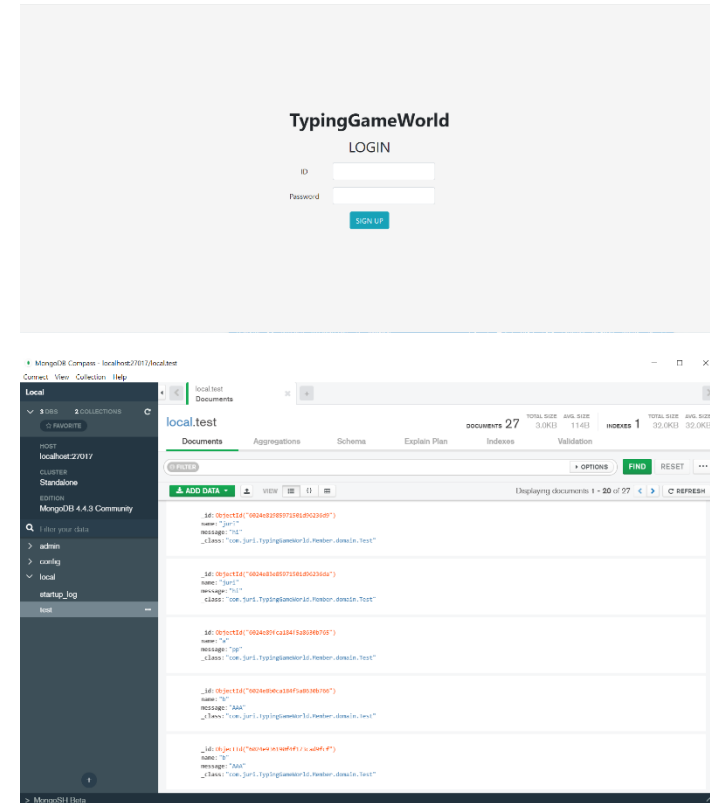
■ 로그인/회원가입 부분 구현 완료

■ [https://github.com/zlxlwnfl/TypingGameWorld\\_Frontend](https://github.com/zlxlwnfl/TypingGameWorld_Frontend)

■ [https://github.com/zlxlwnfl/TypingGameWorld\\_Backend](https://github.com/zlxlwnfl/TypingGameWorld_Backend)

■ 비동기/논블로킹 웹사이트 구현

- Reactor 기반의 비동기/논블로킹 IO 구현 연습과 Istio를 활용한 Polyglot MSA 서버 제작 연습용 웹사이트 프로젝트
- Kotlin으로 회원관리용 서비스를 구현하며 함수형 프로그래밍 연습 중
- Router-Handler-Service 레이어로 구성



# XNXGAMES

■ SpringBoot MVC/MySQL/RabbitMQ/K8s MSA 게임커뮤니티 웹사이트 (제작중)

■ <제작자>

김주리

■ <기술스택>

운영체제 : Windows 10 / Linux Ubuntu(Docker)

프로그래밍 언어 : JAVA / JavaScript

백엔드 프레임워크 : SpringBoot 2.2.2 / Spring Web MVC

MSA 시스템 : Spring Cloud(Eureka, Zuul, Ribbon 등)

보안 : Spring Security->Keycloak(예정)

데이터베이스 : MySQL

메시지 미들웨어 : RabbitMQ

컨테이너 환경 : Docker / Kubernetes(minikube)

로그 시스템 : ELK스택(Logstash, Elasticsearch, Kibana) 7.6.1

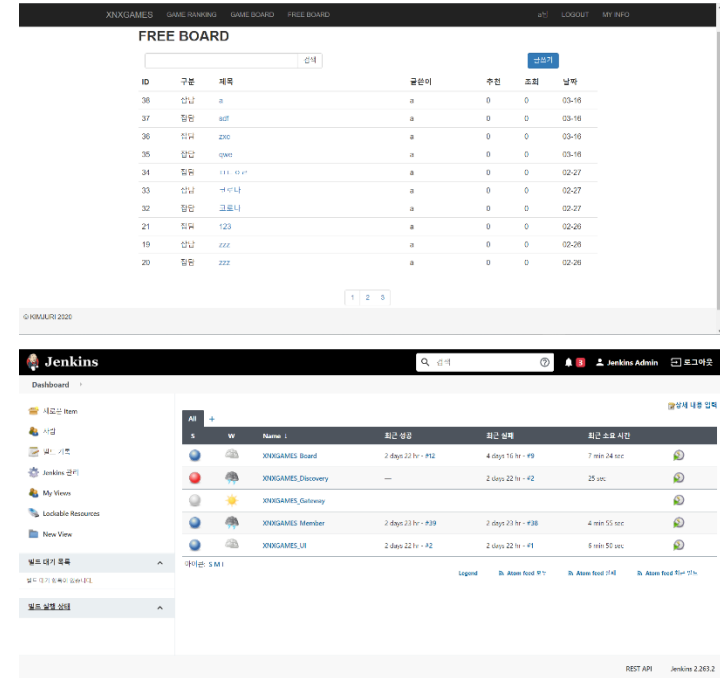
기타 : JPA(Hibernate) / Bootstrap / Jenkins

■ 로그인/회원가입/게시판(리스트, CRUD, 댓글, 검색어 추천 등) 구현 완료

■ [https://github.com/zxlwnfl/XNXGAMES\\_Web\\_Client-Server](https://github.com/zxlwnfl/XNXGAMES_Web_Client-Server)

■ MSA 시스템 구축

- Discovery, Gateway, UI, 게시판, 회원 정보 서비스 각각 구축(현재 UI에서 인증 영역까지 처리하는데, 분리 예정)
- UI만이 사용자와 직접 연결되고, 나머지 서비스들은 서비스끼리만 통신(UI에서 View 반환)
- 서비스들은 모두 Discovery에 등록되고, API 서비스는 Gateway를 통해서만 접근 가능
- 각각의 서비스들은 각자 관리하는 DB가 있으며, 그 DB는 다른 서비스에서는 접근 불가
- 서비스 간에는 RestTemplate을 이용하여 동기 통신하거나, RabbitMQ를 이용하여 비동기 통신
- Docker 위에 올려서 직접 관리하였으나 현재 K8s에 올리는 작업 진행 중(minikube 활용)
- Jenkin를 이용하여 CI/CD 구축 중(현재 Github Push -> Maven Build -> Docker Build -> DockerHub Push 까지 완료)



# 커커오톡

## ■ MFC MongoDB 윈도우 채팅 프로그램 (제작중)

### ■ <제작자>

김주리

### ■ <기술스택>

운영체제 : Windows 10 / Linux Ubuntu(VMware)

프로그래밍 언어 : C++

윈도우 프레임워크 : MFC->WinForm(예정)

데이터베이스 : MongoDB / Redis

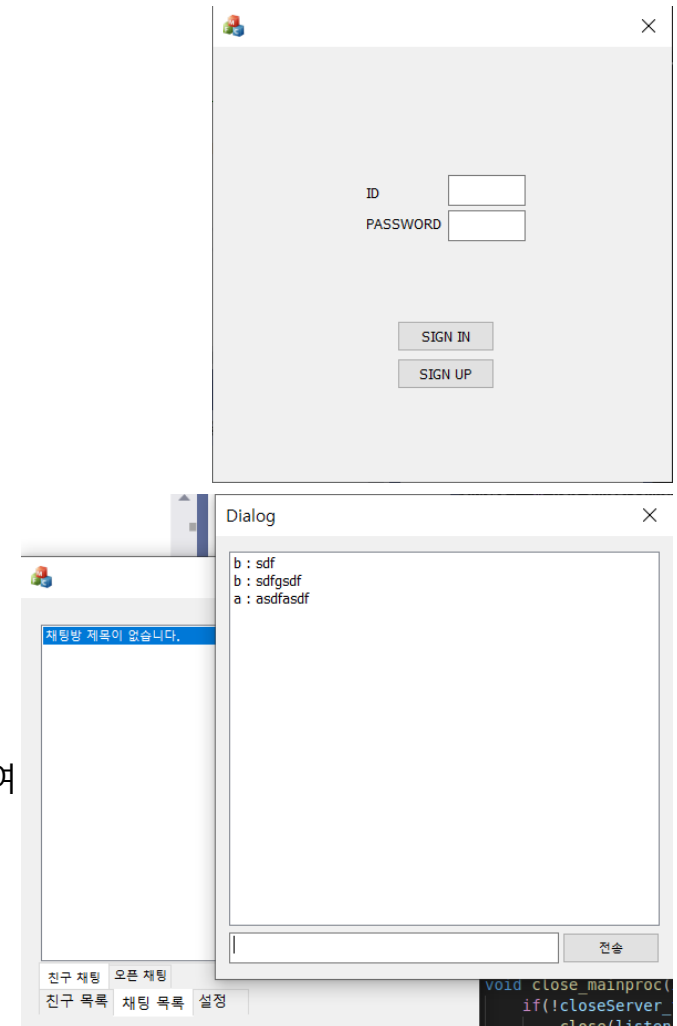
## ■ 회원가입/로그인/유저 채팅 리스트/채팅방 통신 구현 완료

■ [https://github.com/zlxlwnfl/MFC\\_Chat\\_Client](https://github.com/zlxlwnfl/MFC_Chat_Client)

■ [https://github.com/zlxlwnfl/MFC\\_Chat\\_Server](https://github.com/zlxlwnfl/MFC_Chat_Server)

## ■ TCP 서버

- 서버에 클라이언트가 접속할 때마다 새로운 스레드를 생성하고 로직을 처리할 Controller 객체 생성
- 다중 채팅 구현을 위해 멀티룸 구조 설계(RoomManager 객체가 관리)
- 현재 접속한 유저와 활성화된 채팅룸 데이터들은 Redis를 이용하여 관리
- 한 유저가 채팅방에 메시지를 보내면 해당 채팅방의 모든 유저들에게 메시지 전달(Controller 객체 이용)
- 받은 메시지와 보낼 메시지는 Controller에서 각각 수신 큐, 전송 큐를 활용하여 이벤트 큐 방식으로 관리(이벤트 처리는 순서유지를 위해 싱글 스레드)
- 수신 큐, 전송 큐의 메시지 전달, 처리 로직을 나누어  
Producer, Consumer 패턴 적용(Consumer는 큐가 비어 있지 않으면 계속 처리)
- Producer, Consumer 패턴을 위해 mutex, condition\_variable 활용
- epoll을 활용한 비동기/논블로킹 IO로 변경 + 구체적인 패킷 설계 예정



# SPACE WARS

■ 광운대학교 2018년 1학기 응용소프트웨어실습 프로젝트

■ Unity 2D 슈팅 게임 'SPACE WARS'

■ <제작자>

기획 : 김주리, 김성준, 김광남

디자인 : 김주리

프로그래밍 : 김주리, 김성준, 김광남

■ <기술스택>

운영체제 : 윈도우 10

프로그래밍 언어 : C#

게임엔진 : 유니티

오픈소스 : Fungus(유니티 대화창 구현)

디자인 툴 : 포토샵

■ 오프닝 / 스테이지 1 / 스테이지 2 / 스테이지 3 / 엔딩

■ [https://github.com/zlxlwnfl/SpaceWars\\_Unity](https://github.com/zlxlwnfl/SpaceWars_Unity)

■ 게임 구현

- InvokeRepeating 함수로 SpecialPlus 실행
- SpecialCurrent 100되면  
보호막우주선으로 이미지 바뀌게함  
(GetComponent, Resources.Load 이용)
- SpecialCurrent 100 초과 시  
CancelInvoke 함수로 SpecialPlus 실행중지
- 필살기 활성화시 충돌해도  
decreaseHP 함수 실행되지 않게 조정



# DREAM LAB

## ■ 게임제작 연합동아리 'GameMakers' 팀 프로젝트

## ■ Unity 2D 호러 러닝 게임 'DREAM LAB'

### ■ <제작자>

기획 : 김주리, 김찬양, 박정은

디자인 : 김찬양, 박정은

프로그래밍 : 김주리

### ■ <기술스택>

운영체제 : 윈도우 10

프로그래밍 언어 : C#

게임엔진 : 유니티

오픈소스 : Fungus(유니티 대화창 구현)

디자인 툴 : 포토샵

### ■ 오프닝 / 스테이지 1 / 엔딩

### ■ [https://github.com/zlxlwnfl/DREAM-LAB\\_Unty](https://github.com/zlxlwnfl/DREAM-LAB_Unty)

### ■ 게임 구현

- 입력키에 따른 점프/슬라이드/일시정지 구현
- 점프/슬라이드 시 애니메이션 동작
- 장애물 생성 시 Random 함수 사용
- 하트 이미지로 목숨 구현
- HP가 하나씩 깎일 때마다 heart 이미지도 왼쪽부터 순서대로 SetActive(false) 시킴





맺음말

배우고 싶은 것이 많습니다.  
익히고 싶은 것이 많습니다.  
열심히 배우고 익히겠습니다.  
감사합니다.