

포트폴리오

김주리

zlxlwnfl@naver.com

CONTENTS

1

자기소개

2

진행 프로젝트

- MFC MongoDB 카카오톡 프로그램
- SpringBoot MySQL RabbitMQ
MSA 시스템 게임커뮤니티 웹사이트
- Android Firebase 인스타그램 어플
- Unity VR FPS 게임
- Unity 2D 슈팅 게임 등

3

맺음말

자기소개

■ 광운대학교 미디어영상학부/컴퓨터소프트웨어학과 4학년 재학중

■ <기술스택>

운영체제 : Windows, Linux Ubuntu

사용 언어 : C / C++ / C# / JAVA / Javascript

게임 엔진 : Unity

프레임워크 : SpringBoot / Android Studio / MFC

데이터베이스 : MySQL, MongoDB

그 외 : ELK스택, RabbitMQ, Docker

■ <진행 프로젝트>

MFC MongoDB 카카오톡 프로그램 (프론트엔드, 백엔드)

SpringBoot MySQL RabbitMQ MSA 시스템 게임커뮤니티 웹사이트 (프론트엔드, 백엔드)

Android Firebase 인스타그램 (프론트엔드, 백엔드)

Unity VR FPS 게임 (프론트엔드) 등

■ <GIT 주소>

<https://github.com/zlxlwnfl>

MFC MongoDB 카카오톡 프로그램

■ MFC MongoDB 카카오톡 프로그램 (제작중)

■ <제작자>

김주리

■ <기술스택>

운영체제 : 윈도우 10(클라이언트), 리눅스 우분투 18.04(서버)

프로그래밍 언어 : C++

윈도우 프레임워크 : MFC

데이터베이스 : MongoDB, Redis

포맷 변환 라이브러리 : JsonCpp

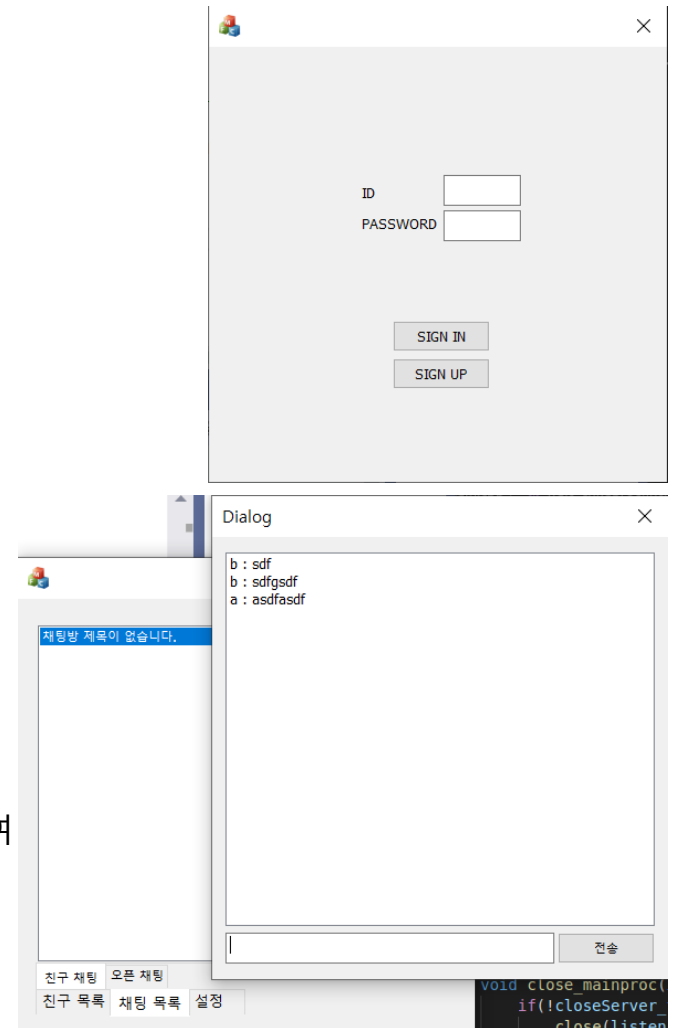
■ 회원가입/로그인/유저 채팅 리스트/채팅방 통신 구현 완료

■ 클라이언트 : https://github.com/zlxlwnfl/MFC_Chatter_Client

서버 : https://github.com/zlxlwnfl/MFC_Chatter_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



XNXGAMES

■ 스프링부트 MySQL RabbitMQ MSA 시스템 게임커뮤니티 웹사이트 (제작중)

■ <제작자>

김주리

■ <기술스택>

운영체제 : 윈도우 10, 리눅스(Docker)

프로그래밍 언어 : JAVA / JavaScript

백엔드 프레임워크 : SpringBoot 2.2.2

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

보안 : Spring Security

데이터베이스: MySQL

메시지 미들웨어 : RabbitMQ

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

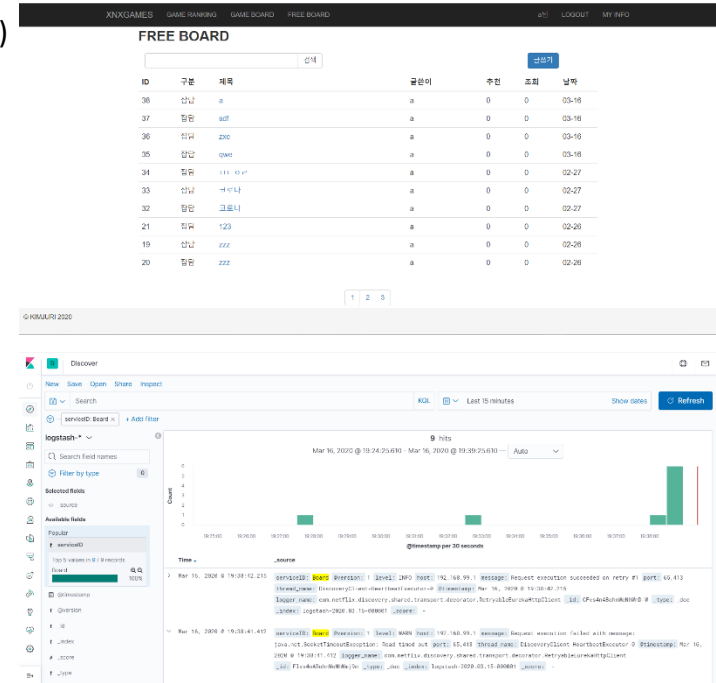
기타 : BootStrap, JPA(Hibernate)

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

■ https://github.com/zlxlwfnl/XNXGAMES_Web_Client-Server

■ MSA 시스템 구축

- Discovery, Gateway, UI, 게시판, 회원 정보 마이크로서비스 각각 구축(현재 UI에서 인증 영역까지 처리하는데, 분리 예정)
- UI만이 사용자와 직접 연결되고, 나머지 서비스들은 서비스끼리만 통신(UI에서 view 반환)
- 서비스들은 모두 Discovery 에 등록되고, API 서비스는 Gateway를 통해서만 접근 가능
- 각각의 서비스들은 각자 관리하는 DB가 있으며, 그 DB는 다른 서비스에서는 접근 불가
- 다른 서비스의 데이터에 접근하기 위해서는 그 데이터를 관리하는 서비스의 인터페이스를 활용하여 따로 요청해야 함
- 서비스 간에는 RestTemplate을 이용하여 동기 통신하거나, RabbitMQ를 이용하여 비동기 통신



Android Firebase 인스타그램

■ Android Firebase 인스타그램 (제작중)

■ <제작자>

김주리

■ <기술스택>

운영체제 : 윈도우 10

프로그래밍 언어 : JAVA

클라이언트 : 안드로이드 스튜디오

서버 : Firebase

■ 회원 가입/로그인 기능/사진 업로드 기능/

프로필 기능/홈 화면 부분구현 완료

■ <https://github.com/zlxlwnfl/Sample-Instagram>

■ 시작 화면/메인 화면/사용자 화면/홈 화면 구현

■ 세 개의 프래그먼트로 화면 설계

■ 글쓰기 완료 시, 이미지는 Firebase 저장소에, 이미지와 내용 데이터는 Firebase DB에 저장되도록 구현

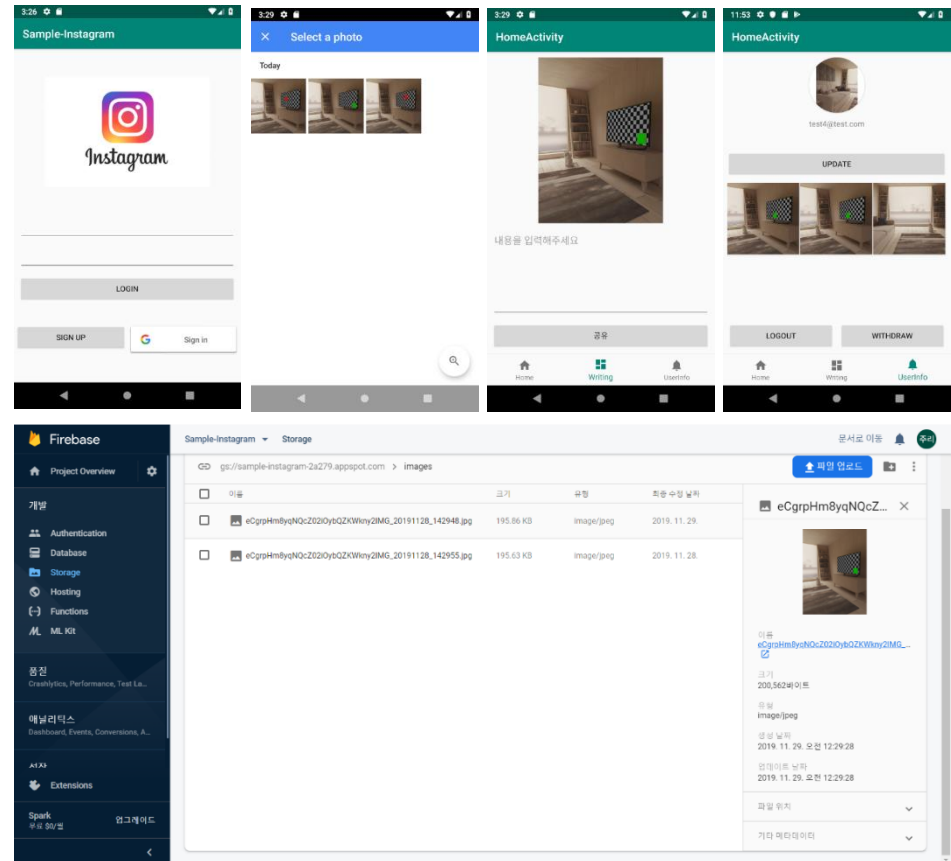
■ RecyclerView 활용

■ 사진 표시할 때, 사이즈 조절을 위하여

Glide 라이브러리 활용

■ RecyclerView로 데이터들을 보여주려 하는데, 아무 데이터도 보이지 않는 오류 발생(비동기 오류)

- Firebase에 데이터가 올라갔을 때(리스너를 통해 확인),
RecyclerView에 데이터가 로딩되게 수정



Unity VR FPS 게임

- 광운대학교 2019년 1학기 HCI와UX 실험 프로젝트
- Unity VR FPS 게임
- <제작자>
 - 기획 : 김주리, 김규리, 김경배, 김우찬
 - 프로그래밍 : 김주리, 김경배
- <기술스택>
 - 운영체제 : 윈도우 10
 - 프로그래밍 언어 : C#
 - 게임엔진 : 유니티
 - VR 기기 : HTC VIVE
- 게임 메인 시스템/튜토리얼 구현완료
- https://github.com/zlxlwnfl/UnityVR_FPS_sample
(용량 문제로 git에 전부 업로드하지는 못함)



- 고정위치에서 고개 회전과
컨트롤러 조작만 인식하는 VR모드로 구현
- 플레이어 Shot, Reload 구현
(알고리즘, 애니메이션, 소리 등)
- 적 Shot 구현
- HP바 UI 구현(고정 위치)
- 탄약 수 UI 구현(AR형식)

Unity 2D 슈팅 게임

- 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
- 일정시간 지날 시 필살기 발동
- 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



Unity 2D 공포 러닝 게임

- 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/zlxlwngl/DREAM-LAB_Untity
- 입력키에 따른 점프/슬라이드/일시정지 구현
- 점프/슬라이드 시 애니메이션 동작
- 장애물 생성 시 Random 함수 사용
- 시간 표현
- Time.deltaTime 이용
- 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.