협동분산시스템 텀 프로젝트 중간보고서 - 20팀

1. 팀원 정보

|  |  |  |
| --- | --- | --- |
| **이름** | **학번** | **역할** |
| 문무현 | 201910162 | 프론트, 백 연결 |
| 김재연 | 201811168 | 프론트 구현 |
| 임지예 | 202212128 | 백엔드 구현 |

1. 구현환경

Backend - intelliJ, Java, spring boot

Frontend - intelliJ, Kotlin

1. 기능별 구현 완료 여부 체크

|  |  |  |
| --- | --- | --- |
| 3.1 | 클라이언트 2개 이상 지원 | O |
| 3.2 | 그림 객체 저장 및 로드 기능 | O |
| 3.3 | Late-comer 클라이언트 지원 | O |
| 3.4 | 동시성 지원 |  |
| 3.5 | 그리기 과정 공유 | O |

1. 전체 클래스 및 메소드 설계

**Backend**

<entity>

* Figure 클래스

도형의 기본 멤버를 정의해 놓은 클래스

<controller>

* WebSocketHandler 클래스

Springframework.web.socket.handler.WebSocketHandler를 상속받은 클래스

<service>

* FigureServiceImpl 클래스

FigureService 인터페이스를 구현한 클래스로 도형 생성, 검색, 수정, 삭제 메소드를 구현한다.

<repository>

* FigureRepositoryImpl

FigureRepository 인터페이스를 구현한 클래스로 도형 정보를 HashMap으로 저장, 삭제하는 메소드를 구현한다.

WhiteboardApplication 클래스

서버를 구동하기 위한 main 메소드를 가지고 있는 클래스이다.

**Frontend**

val canvas 그림이 그려질 캔버스로, HTML 요소이다.

val ctx 2D 렌더링 컨텍스트로서 그리기 함수를 이용하여 캔버스에 접근하기 위한 요소이다.

isDrawing 백엔드와 서로 주고받기 위한 신호. false이면 mousedown이 일어나지 않은 것이다.

linewidth 선의 굵기. 기본적으로 2로 설정되어 있다.

strokeColor 선의 색. 기본적으로 검정색으로 설정되어 있다.

fillColor 채우기 색. 기본적으로 검정색으로 설정되어 있다.

downX mousedown이 일어난 위치의 X좌표.

downY mousedown이 일어난 위치의 Y좌표.

webSocket 서버와 통신을 위한 웹소켓

connect 웹소켓 연결을 위한 메서드

EventListener circleListener

EventListener rectangleListener

EventListener lineListener

EventListener textListener

각 버튼을 누르면 동작할 이벤트 리스너이다.

receiveCircle(message: List<String>)

recieveRectangle(message: List<String>)

recieveLine(message: List<String>)

recieveText(message: List<String>)

다른 client가 그린 도형의 정보를 webSocket 형태로 받은 뒤 해당 정보를 이용해 내 캔버스에도 그리는 메서드이다.

1. 기능별 구현 방법 및 사용한 외부 라이브러리

**Backend**

스프링부트 프레임워크와 stomp-웹소켓을 사용한다.

<entity>

* Figure 클래스

도형의 기본 멤버를 정의해 놓은 클래스

private Type type //Type은 enum으로 정의되어 CIRCLE, RECTANGLE, LINE, TEXT 중 하나이다.

private String lineWidth

private String strokeColor

private String fillColor

private int startX

private int startY

private int endX

private int endY

private String msg

public String toString()

toString()메소드를 오버라이딩하여 객체의 멤버 정보들을 하나의 String으로 return하도록 한다.

<controller>

* WebSocketHandler 클래스

Springframework.web.socket.handler.WebSocketHandler를 상속받은 클래스

private List<WebSocketSession> sessionList

현재 접속한 클라이언트 세션들을 저장하기 위한 리스트

private final FigureService figureService

도형 저장 및 불러오기 메소드를 제공하는 서비스 계층 클래스인 FigureService 객체

생성자 public WebSocketHandler(FigureService figureService)

@Autowired 어노테이션으로 figureService 객체를 생성해 할당한다.

data.txt 파일로부터 그려진 도형 정보를 가져와 프로그램 상에 등록한다.

public void afterConnectionEstablished(WebSocketSession session)

클라이언트가 서버에 연결되었을 때 실행되는 메소드이다.

figureService의 getAll()메소드를 통해 현재 그려진 모든 도형 정보를 얻어와 해당 클라이언트에게 하나씩 전송한다.

헤더로부터 사용자의 이름을 가져와 현재 접속중인 모든 클라이언트에게 접속 메시지를 보낸다.

protected void handleTextMessage(WebSocketSession session, TextMessage message)

클라이언트로부터 받은 메시지를 처리하는 메소드이다.

메시지의 포맷을 통해 메시지의 종류를 구별한다.

메시지 종류는 1. 클라이언트의 접속/해제 메시지 2. 새로 그려진 도형 정보 메시지 가 있는데, 2의 경우 enrollFig()메소드를 통해 프로그램 상에 해당 도형을 저장한다.

이후 1, 2의 경우 모두에 대해 받은 메시지를 모든 클라이언트에 그대로 전송한다.

public void afterConnectionClosed(WebSocketSession session, CloseStatus status)

클라이언트의 접속이 해제되었을 때 실행되는 메소드로, 헤더로부터 사용자의 이름을 가져와 현재 접속중인 모든 클라이언트에게 접속 해제 메시지를 보낸다.

그리고 data.txt 파일에 현재 프로그램에 저장된 모든 도형 정보를 저장한다. (파일 업데이트는 클라이언트 하나가 disconnect될 때마다 이루어짐)

<service>

* FigureServiceImpl 클래스

FigureService 인터페이스를 구현한 클래스로 도형 생성, 검색, 수정, 삭제, 모든 도형 정보 불러오기 메소드를 구현한다.

Public String[] getAll()

현재 저장된 모든 도형 정보를 불러오는 메소드로, figureRepository의 figs()메소드를 호출하여 결과값을 return한다.

<repository>

* FigureRepositoryImpl 클래스

FigureRepository 인터페이스를 구현한 클래스로 도형 정보를 HashMap으로 저장, 삭제하는 메소드, 그리고 현재 저장된 모든 도형 정보를 return하는 메소드를 구현한다.

private Map<Integer, Figure> drawn

현재 그려져 있는 도형 정보를 저장하는 Map으로 HashMap을 이용한다. Key 값으로 각 도형의 시작점 x좌표를 이용하고, Value 값으로 Figure을 갖는다.

Public String[] figs()

현재 저장된 모든 도형 정보를 hashMap으로부터 하나씩 가져와 String배열에 저장해 return한다.

WhiteboardApplication 클래스

서버를 구동하기 위한 main 메소드를 가지고 있는 클래스이다.

**Frontend**

원 그리기 – mousedown 위치와 mouseup 위치의 정 가운데를 중심으로 하는 원을 그린다.

사각형 그리기 – mousedown 위치와 mouseup 위치의 차이를 각 가로, 세로로 하는 사각형을 그린다.

선 그리기 – mousedown 위치와 mouseup 위치를 양 끝단으로 하는 선을 그린다.

텍스트 그리기 – mousedown의 위치와 mouseup 위치의 사각형 안에 textInput의 내용을 그린다.

소켓 통신 – 각 도형을 그릴 때 mousedown의 x, y좌표, mouseup의 x, y 좌표, 선의 색, 굵기, 채우기 색, 텍스트 정보를 메시지 형태로 통신한다.

도형 공유 – 메시지 형태로 받은 소켓을 “ “ 단위로 잘라서 해당 정보를 이용하여 원, 사각형, 선, 텍스트 형태에 맞게 그린다.

1. 클라이언트와 서버 동작 프로토콜(기능 항목별 메시지 포맷, 메시지 송신 조건, 메시지 수신 동작 정의)

기본적으로 WebSocket으로 통신하고, STOMP 프로토콜을 이용한다.

<클라이언트 접속/해제 시>

* 메시지 포맷

Session.getId()님께서 입장/퇴장하셨습니다.

* 서버 측 메시지 송/수신 동작

새로운 클라이언트 접속 시 현재까지 그려진 모든 도형 정보를 전송한다.

메시지를 받으면 연결된 세션들에게 메시지를 전송한다.

* 클라이언트 측 메시지 송/수신 동작

접속/해제 메시지를 클라이언트에게 console.log로 알린다.

<새로운 도형 생성 시>

* 메시지 포맷 : 공백으로 구분되는 문자열

Figure 클래스의 멤버와 일대일 대응하도록 도형 type, lineWidth, startX 등을 공백으로 구분하여 클라이언트와 서버가 통신한다.

예시

circle 2 #000000 #000000 323 495 244 562

* 서버 측 메시지 송/수신 동작

새로운 클라이언트가 접속하면 현재까지 그려진 모든 도형 정보를 전송한다.

한 클라이언트로부터 메시지를 받으면 현재 접속중인 모든 클라이언트에 받은 메시지를 그대로 다시 전송한다.

* 클라이언트 측 메시지 송/수신 동작

서버로부터 메시지를 받은 정보를 통해 도형을 그린다.

1. 컴파일 및 실행 방법

IntelliJ IDEA 2021.3.3

프론트엔드 Kotlin/JS Browser Application 프로젝트 환경에서 실행한다.

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

해당 프로젝트 내 파일을 삭제 후

git remote <https://github.com/MoonMoohyeon/CDS_term-project>

git pull front을 통해 받아서 컴파일 및 실행한다.

백엔드 git clone <https://github.com/MoonMoohyeon/CDS_term-project>을 통해 받아서 컴파일 및 실행한다.

1. 소스코드 포함 프로젝트 Github URL

<https://github.com/MoonMoohyeon/CDS_term-project>

https://github.com/MoonMoohyeon/CDS\_term-project/tree/front

1. 기능별 시연 영상 youtube 링크