TERM PROJECT : Server-Client RMI Whiteboard

1. 과제 개요

A. 개괄

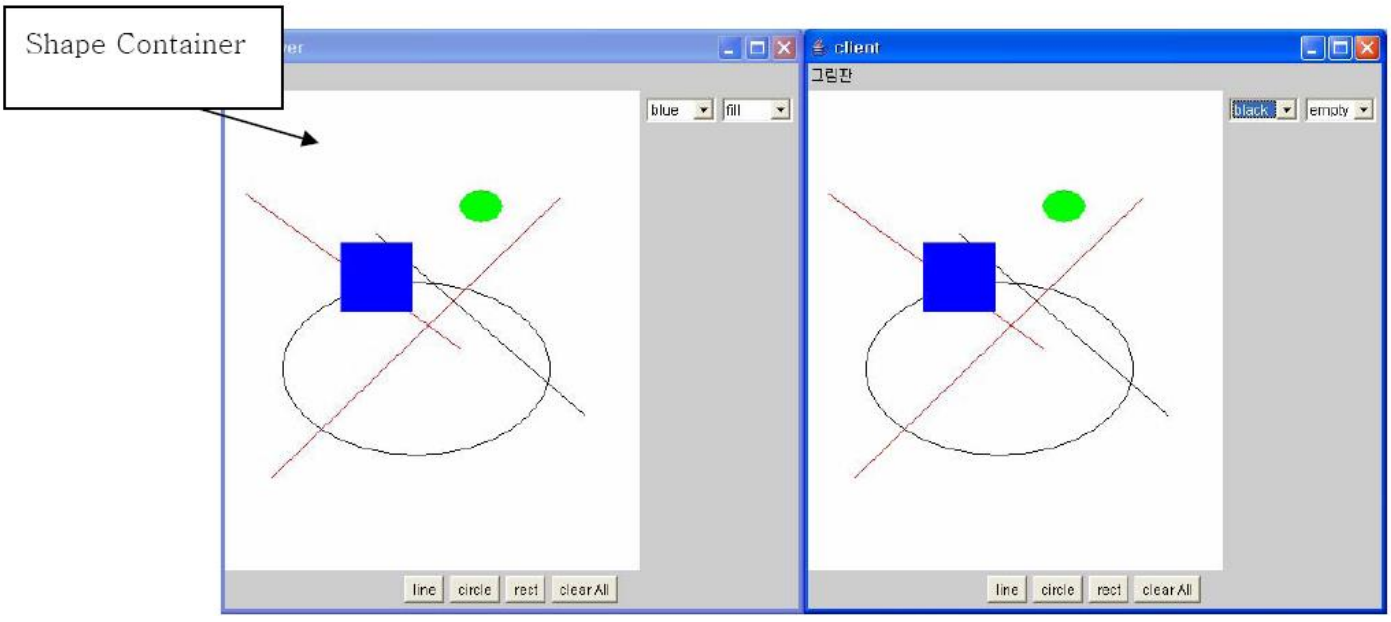
이번 과제를 통해 컴퓨터 공학부 컴퓨터프로그래밍 과목 수강 학생들이 GUI 프로그래밍을 익힐 수 있도록 한다. 과제의 내용은 그림판을 구현하되, 이를 네트워크로 연결하여 서로 연동, 즉 같은 결과를 공유할 수 있도록 구현하도록 한다.

2. 구현 요소

주어진 클래스들을 활용하여 과제를 수행하도록 한다. 편의를 위해 클래스와 Method들을 미리 정해놓았으므로, 과제를 수행하는 학생들은 미완성인 Constructor와 Method들을 채우면 된다. RMI 모듈은 이미 구현이 되어 있다.

3. 구현 기능

Line/Circle/Rectangular 도형을 선택할 수 있는 Button과 Black/Red/Green/Blue 색깔을 선택할 수 있는 Choice-menu 및 Fill/Empty를 선택할 수 있는 Choice-menu를 제공하여 그림을 그릴 수 있는 ShapeContainer를 구현한다.



4. 실행 방법

먼저 cmd 창을 띄우고, 해당 클래스들이 들어있는 폴더로 이동하여 다음과 같은

명령어를 호출한다: rmic WhiteBoardServiceImpl

그 후, 실행은 다음과 같다:
Java WhiteBoardServiceImpl server localhost
Java WhiteBoardServiceImpl client localhost

5. 구현 과정

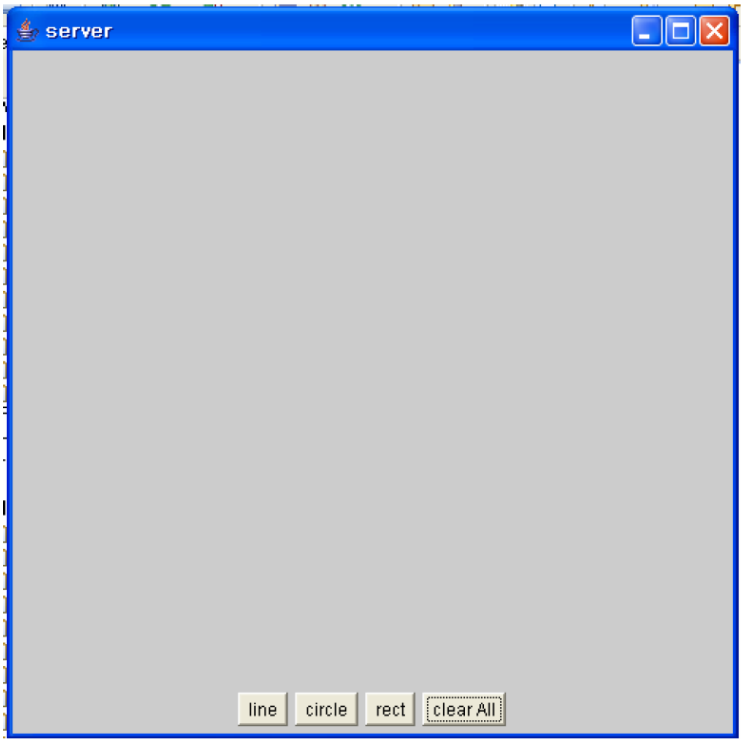
A. 버튼 추가하기

Java GUI 프로그래밍을 처음 접하는 학생들이 많을 것으로 예상되므로, 버튼을 추가하는 작업을 자세히 알려주기로 한다. 버튼 외의 나머지 부분들은 버튼을 구현하는 과정을 잘 살펴보고 비슷한 방법으로 유추하면 된다.

1. 먼저 Panel 클래스를 선언하여 DrawingFrame Class의 ButtonPanel와 연결시킨다
   buttonPanel = new Panel();을 DrawingFrame의 Constructor에 추가
2. Line, Circle, Rectangular, Clear ALL 이렇게 4가지 버튼을 ButtonPanel에 집어넣는다.
   new ButtonController(this);을 DrawingFrame의 Constructor에 추가.
   Button b1 = new Button("line"); 및
   jF.getButtonPanel().add(b1);을 ButtonController의 Constructor에 추가.
3. 이를 main frame의 contentPane의 남쪽에 배치시킨다
   getContentPane().add("South",buttonPanel); 을 DrawingFrame의 constructor에 추가
4. LocalButtonHandler를 이용하여 각 버튼을 눌렀을 때 수행되어야 할 action을 프로그래밍 한다.

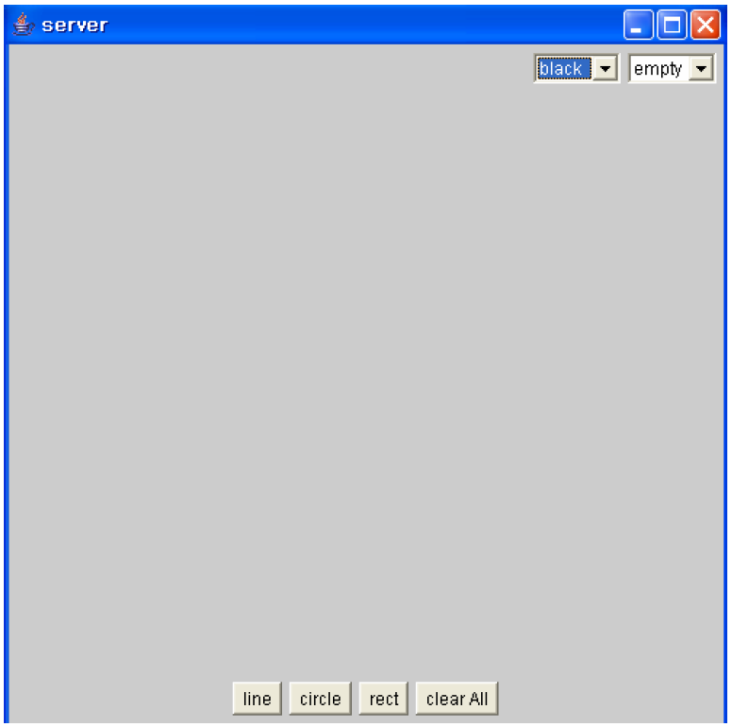
```
if(ae.getActionCommand().equals("line"))
{


// System.out.println("line");
bT.setShape(new Shape().LINE);
} 을 LocalButtonHandler의 actionPerformed에 추가.
```

버튼을 추가한 후 결과 화면은 다음과 같다:



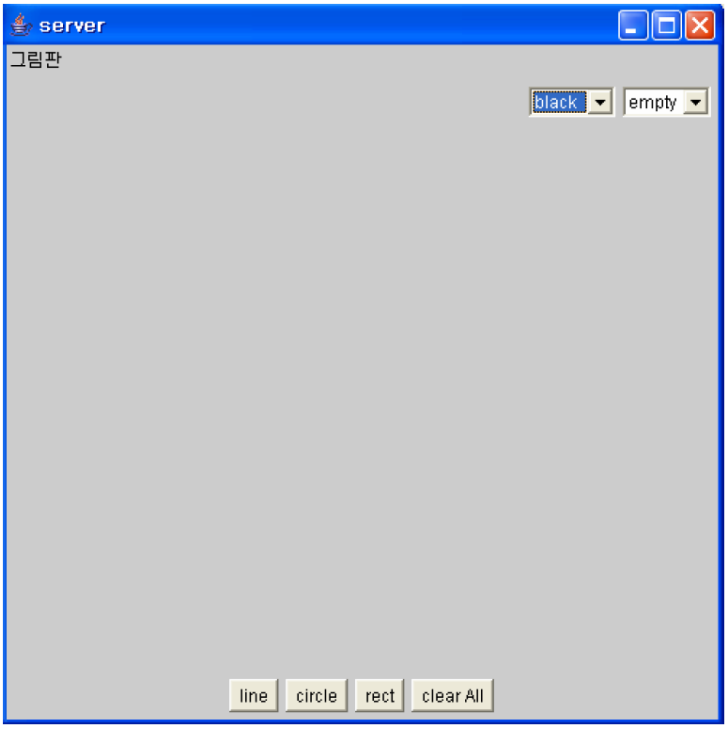
B. 선택 창 추가하기

버튼과 비슷한 방법으로 선택 창을 다음 결과가 나오도록 추가한다.



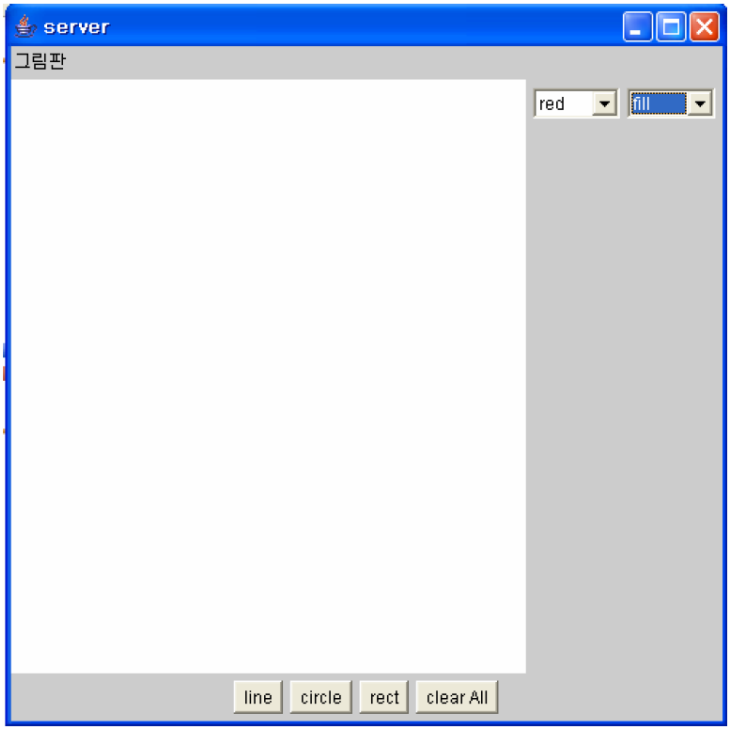
C. 이름 추가하기

프레임의 북쪽에 "그림판" 이라는 제목을 집어넣도록 한다.



D. ShapeContainer 추가하기

이제 실제로 그림을 그릴 부분을 추가하도록 한다. 이를 위해서는 ShapeContainer 클래스를 선언하여 DrawingFrame에 붙여주면 된다. 결과는 다음과 같다.



E. MouseController 클래스 정의하기

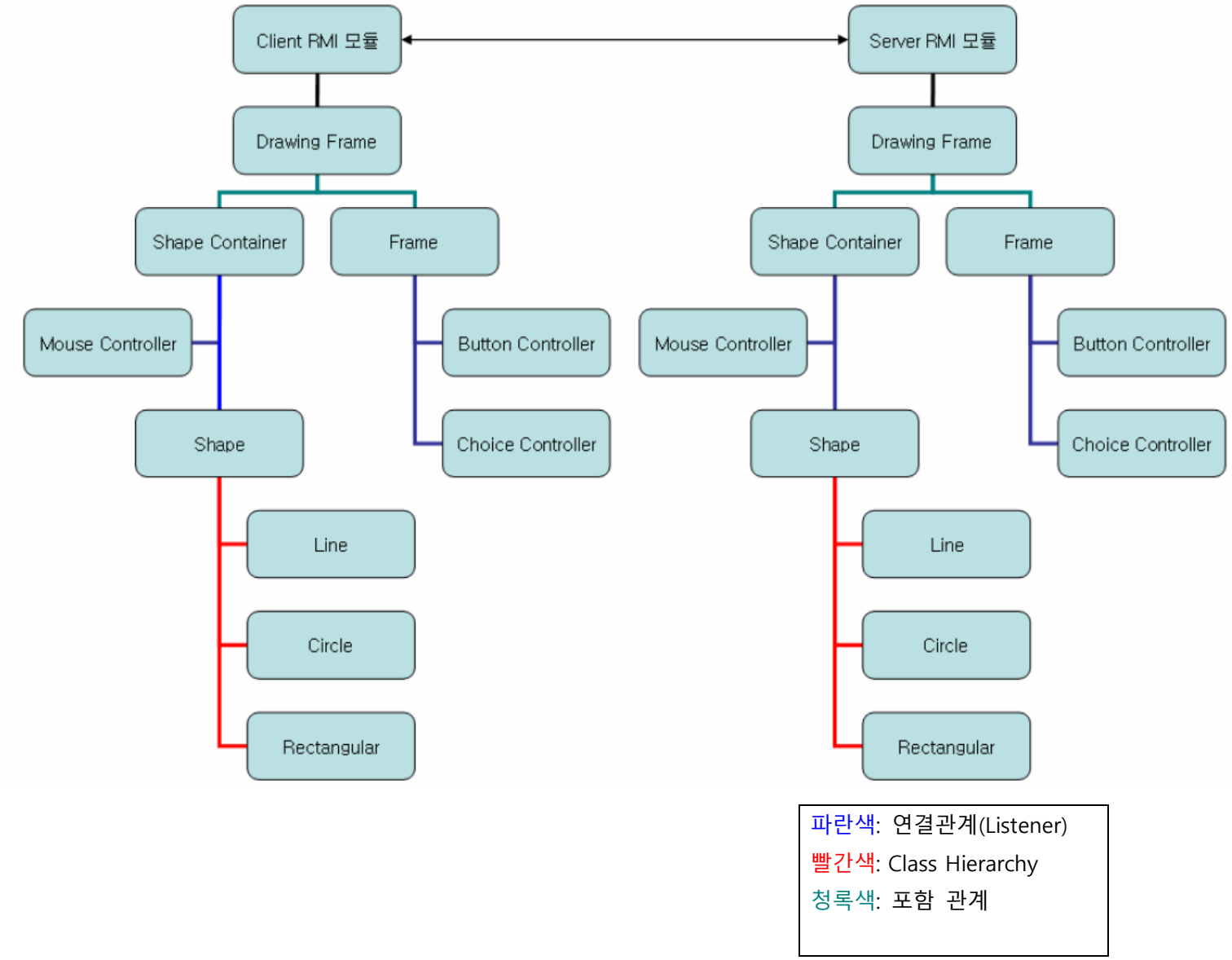
이제 그림을 그릴 부분에 마우스 이벤트를 주었을 때, 이를 처리하는 부분을 구현해 보도록 하자. 먼저 DrawingFrame의 생성자에서 CanvasMouseController를 선언해주고, CanvasMouseController의 Constructor및, 내부 Method들을 정의 하도록 한다. 이 때, CanvasMouseController의 모든 Method를 전부 정의 할 필요는 없고, 필요한 것만 선택해서 구현해도 된다는 점을 유의하도록 한다.

힌트: DrawingFrame 클래스의 removeFrontShape(), addShape() Method를 잘 활용하도록 한다.

F. Line, Circle, Rectangular 그리기

바로 전 단계에서 마우스 이벤트에 따른 도형을 ShapeContainer에 추가하였다면, 이를 실제 그림판 부분에 그려주는 작업해 보자. 먼저 ShapeContainer의 Paint Method에서 Container가 현재 갖고 있는 도형들의 Paint Method를 모두 순차적으로 호출하는 부분을 추가해야 한다. 또한 Line/Circle/Rectangular의 Paint Method 부분에

적절한 정의를 해주도록 한다.



6. Submission – Compressed file that source code and report

Name : PROJ_STUDENT ID_NAME.zip

1. Source Code
   - It is recommended that you create a batch file for compiling such Compile.bat

2. Report
   - Free-form
   - Detailed description of code
   - Need screen capture running program
   - PDF, DOC, HWP

3. How to submit

   snu.comp2015@gmail.com
   - Deadline **6 월 18 일 23 시 59 분**
   - Mail on the subject ""[COMP-PROJ]STUDENT ID_NAME"