



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1 ※ 프로젝트 핵심내용

2

3 1. 미니 프로젝트 명 : 캐치캐치



11

12 2. 팀원 : 오승용 , 최영권

13

14 3 프로젝트 개요(간략한 설명)

15 - 캐치캐치 (그림판과 채팅 기능을 이용한 퀴즈 프로그램)

16

17

18

19 4. 프로젝트 산출물 실용성 (활용 성과 기재)

20 1) 자바 기능 활용할 수 있다.

21

22 2) 데이터 베이스와 연동하여 값을 가져오거나 삽입할 수 있다.

23

24 3) Tcp 네트워킹과 소켓을 이용한 그림퀴즈 채팅프로그램을 구현 할 수 있다.

25

26 소스코드 : <https://github.com/osy0907/Catch-mind>

27

28

29



1. 개요

- StoryBoard

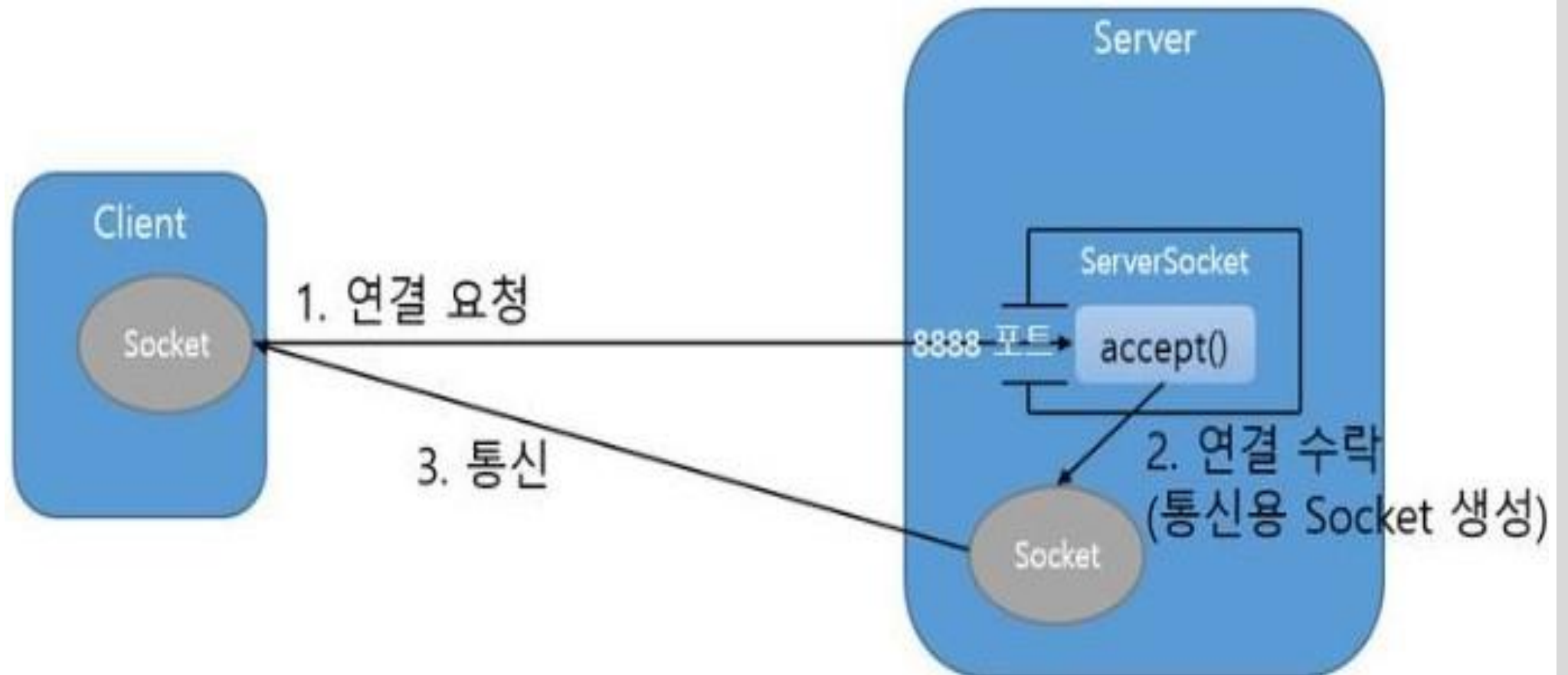
2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

Tcp 네트워킹과 소켓





1. 개요

- StoryBoard

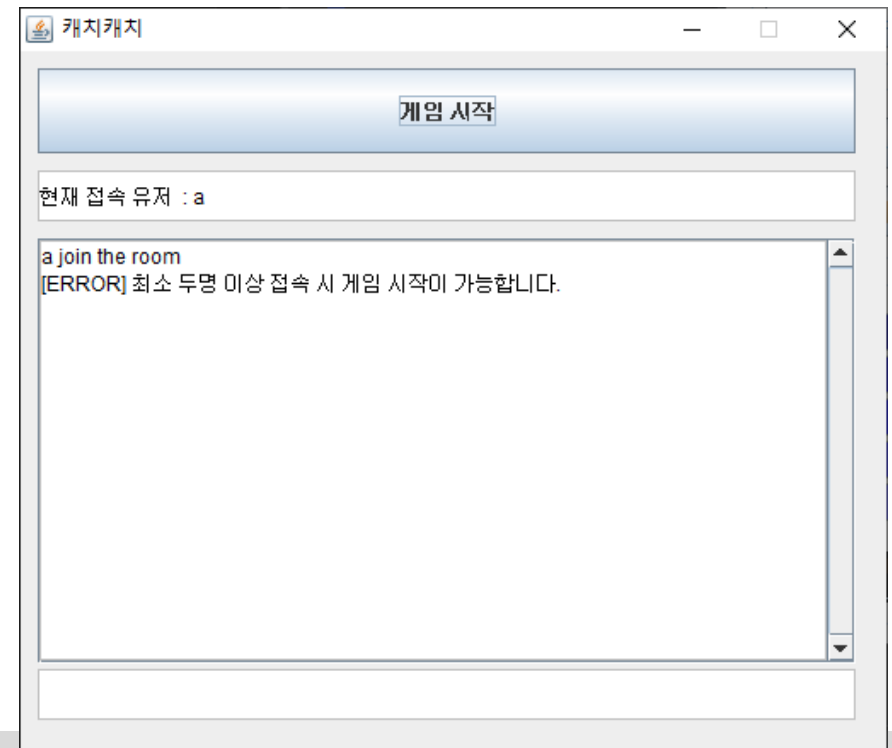
2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void StartEvent() { //StartButton 클릭시 작동하는 이벤트 처리
2     startbtn.addActionListener(new ActionListener() {
3         @Override
4         public void actionPerformed(ActionEvent e) {
5             if (checkStart()) {
6                 screen.append("[ERROR] 최소 두명 이상 접속 시 게임 시작이 가능합니다.\n");
7                 screen.setCaretPosition(screen.getDocument().getLength()); //스크린창의 스크롤을 내린다.
8             } else {
9                 gamestart();
10            }
11        }
12    });
13 }
14
15 private boolean checkStart() {
16     if (List.size() > 1)
17         return false;
18     else
19         return true;
20 }
21
22
23
24
25
26
27
28
29
```





File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



11



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 public class GameloopThread extends Thread {
2
3     private JTextArea screen;
4
5     @Override
6     public void run() { //게임시작 버튼을 눌렀을 경우 시간을 알려주는 메세지 전송
7         Game game = new Game();
8         String answer, Userstr;
9         int index = 0;
10        TimeloopThread time;
11        game.start();
12        GameController.firstStart();
13        for (int i = 3; i > 0; i--) {
14            try {
15                screen.append("CHAT:[알림] 게임시작 " + i + "초 전\n");
16                GameController.allUserMsg("CHAT:[알림] 게임시작 " + i + "초 전");
17                screen.setCaretPosition(screen.getDocument().getLength()); // 스크롤을 제일 아래로
18                Thread.sleep(1000);
19            } catch (InterruptedException e) {
20                // TODO Auto-generated catch block
21                e.printStackTrace();
22            }
23        }
24    }
25
26
27
28
29
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

캐치캐치

— □ ×

게임 시작

현재 접속 유저 : 1 yong

```
1 join the room  
[ERROR] 최소 두명 이상 접속 시 게임 시작이 가능합니다.  
yong join the room  
[SERVER] Start the game.  
CHAT:[알림] 게임시작 3초 전  
CHAT:[알림] 게임시작 2초 전  
CHAT:[알림] 게임시작 1초 전  
[알림] 1 user turn.  
[yong] 개  
[yong] 멍멍이  
[yong] 강아지  
[알림] yong user turn.
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



11



1. 개요

- StoryBoard

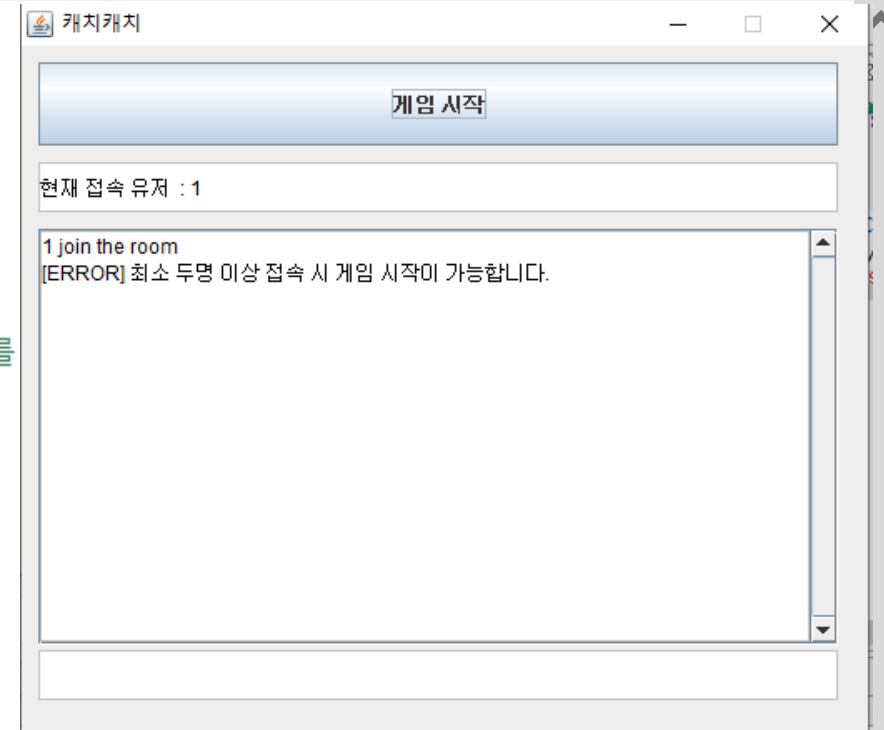
2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void joinchat() {
2     try {
3         ID = userin.readLine();
4         user.setUserID(ID);
5         JoinFieldUpdate();
6         screen.append(ID + " join the room\n");
7         userListText();// 현재 접속하는 유저 모두에게 유저목록과 점수를
8     } catch (IOException e) {
9     }
10
11
12
13
14
15
16 private void JoinFieldUpdate() {
17     String str = new String();
18     str = "현재 접속 유저 : ";
19     for (int i = 0; i < ServerController.List.size(); i++) {
20         str += ServerController.List.get(i).getUserID() + " ";
21     }
22     join.setText(str);
23 }
24
25
26
27
28
29
```





1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 package Client;
2
3 import Client.Core.socket.Controller;
4
5 public class Main {
6
7     static public void main(String[] args) {
8
9         String id = null, pw = null;
10        Controller Socket = new Controller(); // Controller의 객체 Socket 생성
11        MainDesign design = new MainDesign(); // MainDesign의 객체 design 생성
12        initDesign init = new initDesign(); // initDesign의 객체 init 생성
13        init.makeFrame(); // initDesign의 객체 init의 makeFrame() 호출
14        do {
15            id = init.getID();
16            pw = init.getPW();
17            System.out.println("");
18        } while (init.getLoginchk() == false); // db에 값과 확인해서 로그인 성공이 아닐 경우 계속 반복
19
20        Socket.setIP("127.0.0.1"); // Socket의 setIP()메서드를 호출 => init에서 얻어온 ip값을 매개변수로
21        Socket.setPort(8888); // Socket의 setPort()메서드를 호출 => 8888(서버에서 설정한 포트번호)
22        Socket.setId(id); // Socket의 setIP()메서드를 호출 => init에서 얻어온 ip값을 매개변수로
23        design.makeFrame(); // MainDesign의 객체 design의 makeFrame() 호출
24
25
26
27
28
29
```



1. 개요

- StoryBoard

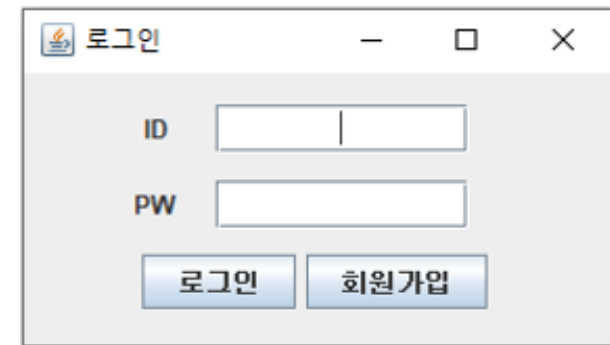
2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 package Client.Design;
2 import java.awt.Font;
3 public class initDesign {
4     private JFrame frame;
5     private JTextField textField;
6     private JPasswordField passwordField;
7     private LoginJDBC db;
8     private String id = null, pw = null;
9     private boolean loginchk;
10
11     public void makeFrame() {
12         drawFrame();
13     }
14
15     private void drawFrame() {
16         // TODO Auto-generated method stub
17         frame = new JFrame("로그인");
18         frame.setSize(295, 168);
19         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20         frame.setLocationRelativeTo(null); //윈도우를 매개변수 안의 컴포넌트에 따라 상대적인 위치를 지정할수 있다.
21         //null일 경우 윈도우 창을 화면의 가운데에 띄우는 역할을 한다.
22
23         JPanel logPane = new JPanel(); // JPanel의 객체 logPane 생성
24         logPane.setBounds(39, 5, 199, 109);
25         logPane.setLayout(new GridLayout(0, 1, 5, 5));
26
27         JPanel panel1 = new JPanel(); // JPanel의 객체 panel1 생성
28         panel1.setLayout(null);
29         JLabel lblNewLabel1 = new JLabel("ID"); // JLabel의 객체 lblNewLabel1 생성
30         lblNewLabel1.setFont(new Font("맑은고딕", Font.BOLD, 12));
31         lblNewLabel1.setBounds(0, 10, 47, 23);
32         lblNewLabel1.setHorizontalAlignment(SwingConstants.CENTER); // JLabel의 가운데정렬
33         panel1.add(lblNewLabel1);
34
35         textField = new JTextField(); // JTextField의 객체 textField 생성
36         textField.setBounds(52, 10, 120, 23);
37         textField.setHorizontalAlignment(SwingConstants.CENTER);
38         textField.setColumns(10);
39         panel1.add(textField);
40         panel1.setBounds(30, 30, 20, 20);
41         logPane.add(panel1);
```





1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 JPanel panel2 = new JPanel();
2 panel2.setLayout(null);
3 JLabel lblNewLabel2 = new JLabel("PW");
4 lblNewLabel2.setFont(new Font("맑은고딕", Font.BOLD, 12));
5 lblNewLabel2.setBounds(0, 8, 47, 23);
6 lblNewLabel2.setHorizontalAlignment(SwingConstants.CENTER);
7 panel2.add(lblNewLabel2);
8
9 passwordField = new JPasswordField(10);
10 passwordField.setBounds(52, 8, 120, 23);
11 passwordField.setHorizontalAlignment(JTextField.CENTER);
12 panel2.add(passwordField);
13 logPane.add(panel2);
14
15 JPanel panel3 = new JPanel();
16 JButton btnNewButton = new JButton("로그인");
17 btnNewButton.addActionListener(new ActionListener() {
18     @Override
19     public void actionPerformed(ActionEvent e) {
20         id = textField.getText();
21         pw = passwordField.getText();
22         db = new LoginJDBC(btnNewButton.getText(), id, pw);
23         loginchk = db.getLoginchk(); // db의 loginchk를 얻어온다.
24         if (loginchk == true) {
25             frame.dispose(); // 원하는 하나의 Frame만 종료 시키기 위해서 dispose() 메소드를 사용
26         }
27     }
28 });
29
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1      JButton btnNewButton_1 = new JButton("회원가입");
2      btnNewButton_1.addActionListener(new ActionListener() {
3
4          @Override
5          public void actionPerformed(ActionEvent e) {
6              id = textField.getText();
7              pw = passwordField.getText();
8              db = new LoginJDBC(btnNewButton_1.getText(), id, pw);
9          }
10     });
11     frame.getContentPane().setLayout(null); // 레이아웃을 설정하지 않음
12
13     panel3.add(btnNewButton);
14     panel3.add(btnNewButton_1);
15
16     logPane.add(panel3);
17     frame.getContentPane().add(logPane);
18     frame.setVisible(true);
19
20     public String getID() {
21         return this.id;
22     }
23
24     public String getPW() {
25         return this.pw;
26     }
27
28     public boolean getLoginchk() {
29         return this.loginchk;
30     }
31 }
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 package Client.Design;
2
3 import java.sql.Connection;
4
5 public class LoginJDBC extends JFrame {
6     Connection con = null;
7     Statement stmt = null;
8
9     private boolean Loginchk = false; // Loginchk의 기본 값을 false로 설정
10
11     public LoginJDBC(String loginType, String id, String pw) {
12
13         try {
14             Class.forName("oracle.jdbc.driver.OracleDriver");// JDBC 드라이버 로딩
15             con = DriverManager.getConnection("jdbc:oracle:thin:@192.168.1.34:1521:orcl");
16             stmt = con.createStatement(); // Connection 인스턴스를 이용해서 Statement 객체 생성
17             System.out.println("DB 접속 완료!");
18
19             if (loginType == "로그인") { // 앞에서 만들었던 버튼의 텍스트를 비교 로그인버튼을 눌렀을 경우
20                 String sql = "select id from member2 where Id = '" + id + "' and Pw = '" + pw + "'";
21                 ResultSet rs = stmt.executeQuery(sql); // Statement 객체의 결과를 executeQuery로 가져옴
22
23                 if (!rs.next()) { // 일치하는 아이디와 비밀번호가 없을 때
24                     if (!id.equals("")) { // 아이디가 빈칸이 아니라면
25                         JOptionPane.showMessageDialog(getParent(), "로그인 실패하였습니다.");
26                     } else {
27                         JOptionPane.showMessageDialog(getParent(), "아이디를 입력해주세요.");
28                     }
29                 } else {
30                     Loginchk = true; // 로그인 성공시에만 Loginchk가 true가 된다.
31                     JOptionPane.showMessageDialog(getParent(), "로그인 성공하였습니다.");
32                 }
33             }
34         } catch (Exception e) {
35             e.printStackTrace();
36         }
37     }
38 }
```

로그인

ID	<input type="text" value="a"/>
PW	<input type="password" value="•"/>
<input type="button" value="로그인"/> <input type="button" value="회원가입"/>	

메시지

로그인 실패하였습니다.

로그인

ID	<input type="text"/>
PW	<input type="password"/>
<input type="button" value="로그인"/> <input type="button" value="회원가입"/>	

메시지

아이디를 입력해주세요.

로그인

ID	<input type="text" value="abc"/>
PW	<input type="password" value="••••"/>
<input type="button" value="로그인"/> <input type="button" value="회원가입"/>	

메시지

로그인 성공하였습니다.



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1      } else if (loginType == "회원가입") {
2          String sql = "select Id from member2 where Id = '" + id + "'";
3          ResultSet rs = stmt.executeQuery(sql);
4
5          if (!rs.next()) {
6              if (!id.equals("")) {
7                  String sql2 = "insert into member2(Id,Pw) values('" + id + "', '" + pw.toString() + "')";
8                  System.out.println(sql2);
9                  stmt.executeUpdate(sql2);
10                 JOptionPane.showMessageDialog(getParent(), "회원가입에 성공했습니다.");
11             } else {
12                 JOptionPane.showMessageDialog(getParent(), "아이디를 입력해주세요.");
13             }
14         } else {
15             JOptionPane.showMessageDialog(getParent(), "이미 있는 아이디입니다.");
16         }
17     }
18
19     } catch (ClassNotFoundException e) {
20         // TODO: handle exception
21         System.out.println(e.getMessage());
22     } catch (SQLException e) {
23         System.out.println(e.getMessage());
24     }
25
26     }
27
28     public boolean getLoginchk() {
29         return Loginchk;
30     }
31 }
```

로그인 폼: ID는 'abc', PW는 '....'로 입력되어 있음. 로그인 버튼과 회원가입 버튼이 있음.

메시지 박스: 회원가입에 성공했습니다. 확인 버튼이 있음.

로그인 폼: ID와 PW는 비어 있음. 로그인 버튼과 회원가입 버튼이 있음.

메시지 박스: 아이디를 입력해주세요. 확인 버튼이 있음.

로그인 폼: ID는 'a', PW는 '....'로 입력되어 있음. 로그인 버튼과 회원가입 버튼이 있음.

메시지 박스: 이미 있는 아이디입니다. 확인 버튼이 있음.



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

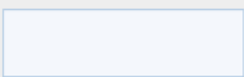
- Client

- Server

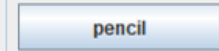
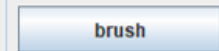
3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

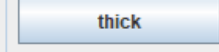
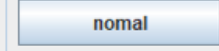
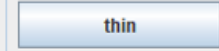
캐치캐치



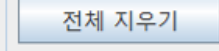
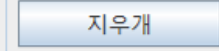
그리기 도구



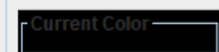
펜 두께



지우개



Color



ID / Score

전송



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void drawImgSection() { // 이미지를 그려낼 그림판 객체를 프레임에 생성
2     imgbuff = new BufferedImage(panelWidth, panelHeight, BufferedImage.TYPE_INT_ARGB); // 가로 세로는 현재 패널의 크기만큼 그리고 이미지
3                                                     // 타입을 지정
4
5     // private final int panelWidth = 840; // 그림판 너비
6     // private final int panelHeight = 671; // 그림판 높이
7     imgpanel = new JLabel(new ImageIcon(imgbuff));
8     imgpanel.setBounds(180, 0, panelWidth, panelHeight);
9     imgpanel.setBorder(BorderFactory.createLineBorder(Color.LIGHT_GRAY, 1));
10    frame.getContentPane().add(imgpanel);
11 }
12
13 private void makeBrush() { // 그림을 그릴때 좌표를 공유할 브러쉬 객체를 프레임에 생성
14     brush = new Brush();
15     brush.setBounds(180, 0, panelWidth, panelHeight);
16     brush.repaint();
17     brush.printAll(imgbuff.getGraphics()); // 이미지 버퍼에 있는 그래픽을 얻어와서 컴포넌트 모두 출력함
18     frame.getContentPane().add(brush);
19 }
20
21
22
23
24
25
26
27
28
29
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1
2 private void makeMouseEvent() {
3     imgpanel.addMouseMotionListener(new MouseMotionListener() {
4         @Override
5         public void mouseDragged(MouseEvent e) { // 마우스 드래그시
6             if (GameController.turnflag == true) { // 게임의 차례가 내순서일 경우(혹은 게임이 시작이 아직 안되었을때)
7                 SendMessage.send.println("Position:" + e.getX() + "," + e.getY()); // 서버로 좌표값 전송 마우스 드래그 시 해당 좌표값 보냄
8                 SendMessage.send.flush();|
9             }
10        }
11
12        public void mouseMoved(MouseEvent e) {
13        }
14    });
15 }
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1
2 private void getMsg() {
3     while (true) {
4         try {
5             msg = msgbuff.readLine(); // sever.getInputStream()을 얻어와서 입력받은 걸 한줄씩 msg에 담는다.
6             if (msg.contains(":")) { // msg 에 ':'이 포함되어있다면 실행
7                 String[] pars = msg.split(":"); // String 배열 pars에 msg를 ':'문자 기준으로 나눈 문자열을 담는다.
8                 if (pars[0].equals("Position")) { // Position : x,y ->좌표 값을 지정해주는 구문
9                     // mainDesign 에서 (GameController.turnflag == true)일때 서버로 출력후
10                    pars = pars[1].split(",");
11                    // ':' 기준으로 나눈 문자열 인덱스 1번을 ',' 문자 기준으로 나눠서 pars배열에 다시 담음 (이전 문자배열 사라짐)
12                    x = Integer.parseInt(pars[0]); // ',' 문자 기준으로 pars[0] == x좌표값
13                    y = Integer.parseInt(pars[1]); // ',' 문자 기준으로 pars[1] == y좌표값
14                    brush.setX(x); // brush 객체에 setX() 메서드를 호출하여 x좌표값을 지정
15                    brush.setY(y); // brush 객체에 setY() 메서드를 호출하여 y좌표값을 지정
16                    brush.repaint(); // brush객체의 repaint()메서드를 호출하여 초기화(repaint는 기본정의된 메서드)
17                    brush.printAll(imgbuff.getGraphics()); // setImgbuff()로 설정한 imgbuff의 Graphics을 얻어옴
18                }
19            }
20        }
21    }
22
23
24
25
26
27
28
29
```

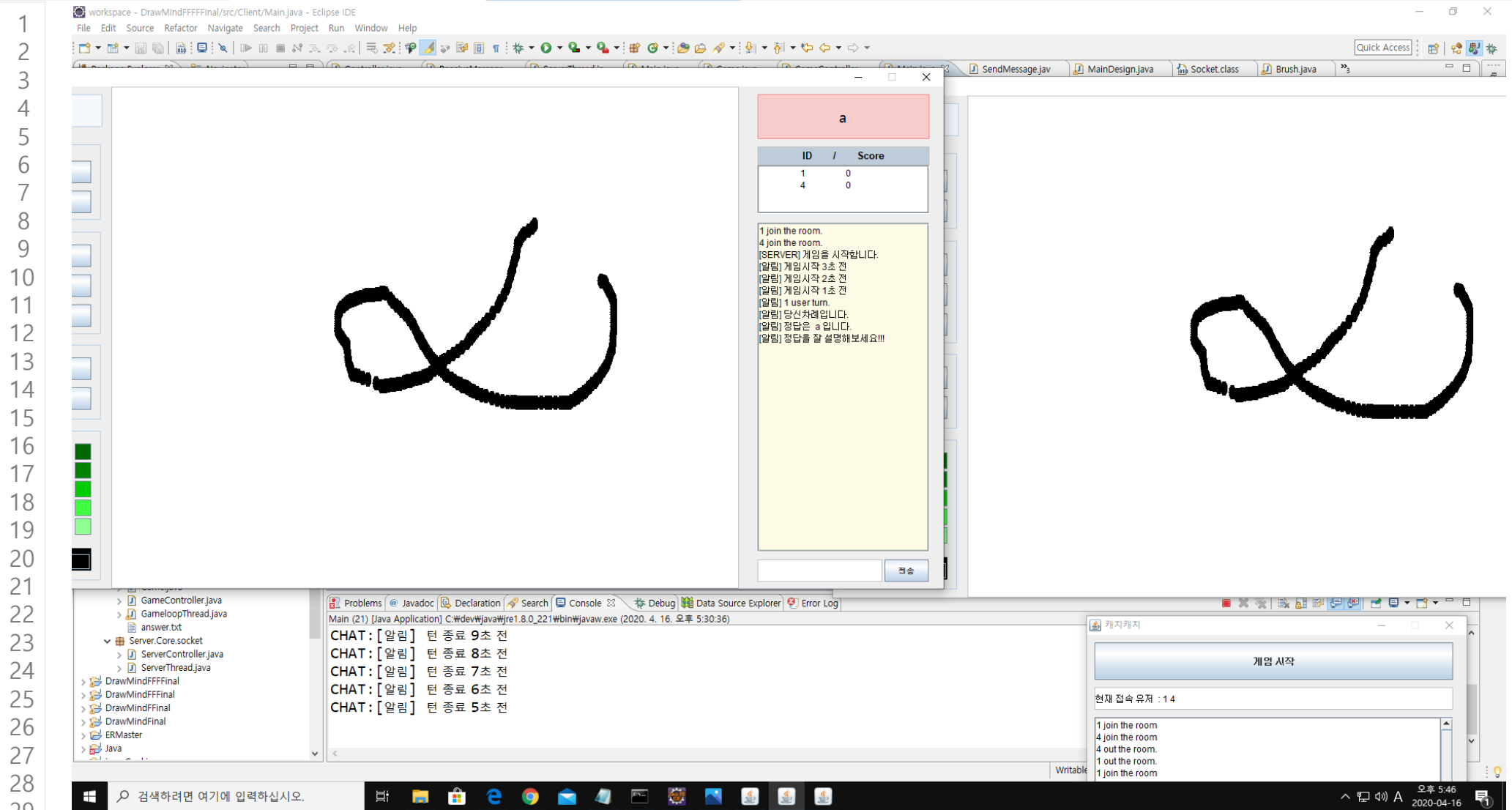

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행





File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

캐치캐치

그리기 도구

brush

pencil

펜 두께

thin

nomal

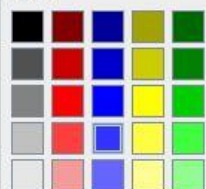
thick

지우개

지우개

전체 지우기

Color



Current Color

ID / Score

b

0

b join the room.

전송



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void brushSelect() { // 그리기 도구 설정
2
3     brushJPanel = new JPanel();
4     brushJPanel.setLayout(null);
5     brushJPanel.setBounds(13, 69, 155, 111);
6     brushJPanel.setBorder(new TitledBorder("그리기 도구"));
7
8     penbtn = new Penbtn();
9     penbtn.setBrush(brush);
10    penbtn.makePenbtn();
11
12    brushJPanel.add(penbtn.penButton()); // 버튼을 패널에 추가
13    brushJPanel.add(penbtn.brushButton()); // 버튼을 패널에 추가
14
15    frame.getContentPane().add(brushJPanel); // 패널을 프레임에 추가
16
17 private void drawButtons() { // 컬러선택 버튼
18     btns = new Colorbtns();
19     btns.setBrush(brush);
20     btns.setCurrentColor(currentColor);
21     btns.makebutton();
22
23     currentColor.setBorder(new TitledBorder("Current Color"));
24     currentColor.setBounds(10, 165, 130, 30);
25
26     colorbtnPanel = new JPanel();
27     colorbtnPanel.setLayout(null);
28     colorbtnPanel.setBounds(13, 452, 155, 210);
29     colorbtnPanel.setBorder(new TitledBorder("Color"));
30     colorbtnPanel.setBackground(brush.color);
31
32     colorbtnPanel.add(btns.getSelectedColor());
33
34     colorbtnPanel.add(currentColor);
35     frame.getContentPane().add(colorbtnPanel);
36
37     makeEraserButton();
```

```
private void lineSelect() { // 선 두께 설정
    linePanel = new JPanel();
    linePanel.setLayout(null);
    linePanel.setBounds(13, 186, 155, 147);
    linePanel.setBorder(new TitledBorder("펜 두께"));
    thicknessbtns = new Thicknessbtns();
    thicknessbtns.setBrush(brush);
    thicknessbtns.makeThicknessbtn();
    linePanel.add(thicknessbtns.thin()); // 버튼을 패널에 추가
    linePanel.add(thicknessbtns.nomal()); // 버튼을 패널에 추가
    linePanel.add(thicknessbtns.thick()); // 버튼을 패널에 추가
    frame.getContentPane().add(linePanel); // 패널을 프레임에 추가
}

private void makeEraserButton() { // 지우개버튼
    panel = new JPanel();
    panel.setLayout(null);
    panel.setBounds(13, 337, 155, 110);
    panel.setBorder(new TitledBorder("지우개"));
    eraserbtn = new Eraserbtn();
    eraserbtn.setBounds(10, 25, 130, 30);
    eraserbtn.setBrush(brush);
    panel.add(eraserbtn);
    makeClearButton();
    frame.getContentPane().add(panel);
}
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

[Brush 클래스]

```
1
2
3
4
5
6 public Color color = Color.black;
7 private static int brushSize = 10;
8 private static String brushType = "brush";
9
10 @Override
11 public void paint(Graphics g) {
12     // paint 메서드 호출 시점 : GUI Application이 실행 될때, damage를 입은 영역을 복구할때 비활성화 되어 다른 창에
13     // 가려진 활성화된 다음 다시 보여질 때
14
15     // 1) Component.repaint() 메소드 호출 (update() 메소드 호출)
16     // 2) Component.update(g) 메소드 호출 (영역을 clear 한 후 paint 메소드 호출)
17     // 3) Component.paint(g) 메소드 호출
18     if (flag == true) {
19         g.setColor(color);
20         g.fillOval(x - 2, y - 2, brushSize, brushSize * 2);
21
22
23
24
25
26
27
28
29
```




1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

[Penbtn 클래스]

```
1 private ActionListener brushAction = new ActionListener() {  
2  
3  
4  
5  
6  
7     @Override  
8     public void actionPerformed(ActionEvent e) {  
9         if (GameController.turnflag == true) {  
10             brushChange(e.getActionCommand());  
11             brush.setBrushSize(size);  
12             SendMessage.send.println("Line:" + brush.getBrushType());  
13             SendMessage.send.println("Think:" + size);  
14             SendMessage.send.flush();  
15         }  
16     }  
17 };
```

```
18 private void brushChange(String type) {  
19     if (type == "brush") {  
20         size = 10;  
21         brush.setBrushType("brush");  
22     } else if (type == "pencil") {  
23         size = 2;  
24         brush.setBrushType("pencil");  
25     }  
26 };
```

```
27 public void setBrush(Brush brush) {  
28     this.brush = brush;  
29 }
```

[Thicknessbtns 클래스]

```
1 @Override  
2 public void actionPerformed(ActionEvent e) {  
3     if (GameController.turnflag == true) {  
4         thinkChange(e.getActionCommand());  
5         System.out.println(brush.getBrushType());  
6         if (brush.getBrushType().equals("brush")) {  
7             size *= 5;  
8             brush.setBrushSize(size);  
9         } else if (brush.getBrushType().equals("brush")) {  
10             brush.setBrushSize(size);  
11         }  
12         SendMessage.send.println("Think:" + size);  
13         SendMessage.send.flush();  
14     }  
15 };
```

```
16 private void thinkChange(String type) {  
17     if (type == "thin") {  
18         size = 1;  
19     } else if (type == "nomal") {  
20         size = 2;  
21     } else if (type == "thick") {  
22         size = 3;  
23     }  
24 };
```

```
25 public void setBrush(Brush brush) {  
26     this.brush = brush;  
27 }
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1 캐치캐치

2

3

4

5 그리기 도구

6 brush

7 pencil

8

9 펜 두께

10 thin

11

12 nomal

13

14 thick

15

16 지우개

17 지우개

18

19 전체 지우기

20

21 Color

22

23

24

25

26

27

28 Current Color

29

[thin]

brush

pencil

[normal]

[thick]

b join the room.

전송



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

[Eraserbtn 클래스]

```
1 private void makeEvent() {  
2     this.addActionListener(new ActionListener() {  
3         public void actionPerformed(ActionEvent e) {  
4             if (GameController.turnflag == true) {  
5                 SendMessage.send.println("Color:WHITE");  
6                 SendMessage.send.flush();  
7                 brush.setColor(Color.WHITE);  
8             }  
9         }  
10    }  
11 }  
12 }  
13 }  
14 }  
15 }  
16 }  
17 }  
18 }  
19 }  
20 }  
21 }  
22 }  
23 }  
24 }  
25 }  
26 }  
27 }  
28 }  
29 }
```

```
public void setBrush(Brush brush) {  
    this.brush = brush;  
    makeEvent();  
}
```





1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

[전체지우기]

```
1
2
3
4 private void makeClearButton() {
5     drawClearButton();
6     ClearButtonEvent();
7 }
8
9 private void drawClearButton() { // 전체지우기 버튼을 프레임에 추가하는 메서드
10     clearbtn = new JButton("전체 지우기");
11     clearbtn.setFont(new Font("맑은 고딕", Font.PLAIN, 14));
12     clearbtn.setBounds(10, 65, 130, 30);
13     panel.add(clearbtn);
14 }
15
16 private void ClearButtonEvent() { // 전체지우기 버튼 액션 메서드
17     clearbtn.addActionListener(new ActionListener() {
18         public void actionPerformed(ActionEvent e) {
19             if (GameController.turnflag == true) {
20                 SendMessage.send.println("MODE: CLEAR");
21                 SendMessage.send.flush();
22             }
23         }
24     });
25
26 private void ClearScreen() { // X표 = 전체지우기를 눌렀을때 실행 메서드
27     brush.setFlag(false); // brush 객체의 flag를 false로 설정
28     brush.repaint(); // brush객체의 repaint()메서드를 호출하여 초기화(repaint는 기본정의된 메서드)
29     brush.printAll(imgbuff.getGraphics()); // setImgbuff()로 설정한 imgbuff의 Graphics를 얻어옴
30 }
31
32 } else if (flag == false) {
33     g.setColor(Color.WHITE);
34     g.fillRect(0, 0, 1000, 1000);
35     flag = true;
36 }
```




1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

[Colorbtns 클래스]

```
public void makebutton() {
```

```
    currentColor.setBackground(Color.BLACK); // 시작색은 검정이니 현재색은 검정색으로 설정한다.
```

```
    selectColor.setLayout(null); // 버튼들의 크기와 위치를 지정해주기위해 Layout 은 null
```

```
    selectColor.setBounds(10, 25, 130, 140);
```

```
    colorButton handler = new colorButton(); // colorButton 에 Mouse Event 설정을 위한 객체 생성
```

```
    for (int i = 0; i < colorButton.length; i++) { // 버튼들의 설정을 위한 이중 for 문
```

```
        for (int j = 0; j < colorButton[i].length; j++) {
```

```
            colorButton[i][j] = new JButton(); // 버튼의 각 원소를 초기화해준다.
```

```
            colorButton[i][j].setBounds((j * 27), (i * 25), 22, 22); // 버튼의 위치를 잡아준다. 버튼 크기는 20x10
```

```
            colorButton[i][j].addActionListener(handler); // 버튼 각원소를 ActionListener에 handler 추가
```

```
            selectColor.add(colorButton[i][j]); // 버튼을 select 에 담는다.
```

```
        }
```

```
    }
```

```
class colorButton implements ActionListener {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        JButton action = (JButton) e.getSource();
```

```
        /*---검은색 열 부분의 액션이 발생했을 때 currentColor색과 변수에color에 색을 지정---*/
```

```
        if (action == colorButton[0][0]) {
```

```
            if (GameController.turnflag == true) {
```

```
                SendMessage.send.println("Color:BLACK");
```

```
                SendMessage.send.flush();
```

```
                brush.setColor(Color.BLACK);
```

```
                currentColor.setBackground(brush.color);
```

```
            }
```

```
        }
```

```
        else if (action == colorButton[1][0]) {
```

```
            if (GameController.turnflag == true) {
```

```
                SendMessage.send.println("Color:AUTOGRAYDEEP");
```

```
                SendMessage.send.flush();
```

```
                brush.setColor(new Color(81, 81, 81));
```

```
                currentColor.setBackground(brush.color);
```

```
            }
```

```
        }
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



11



1. 개요

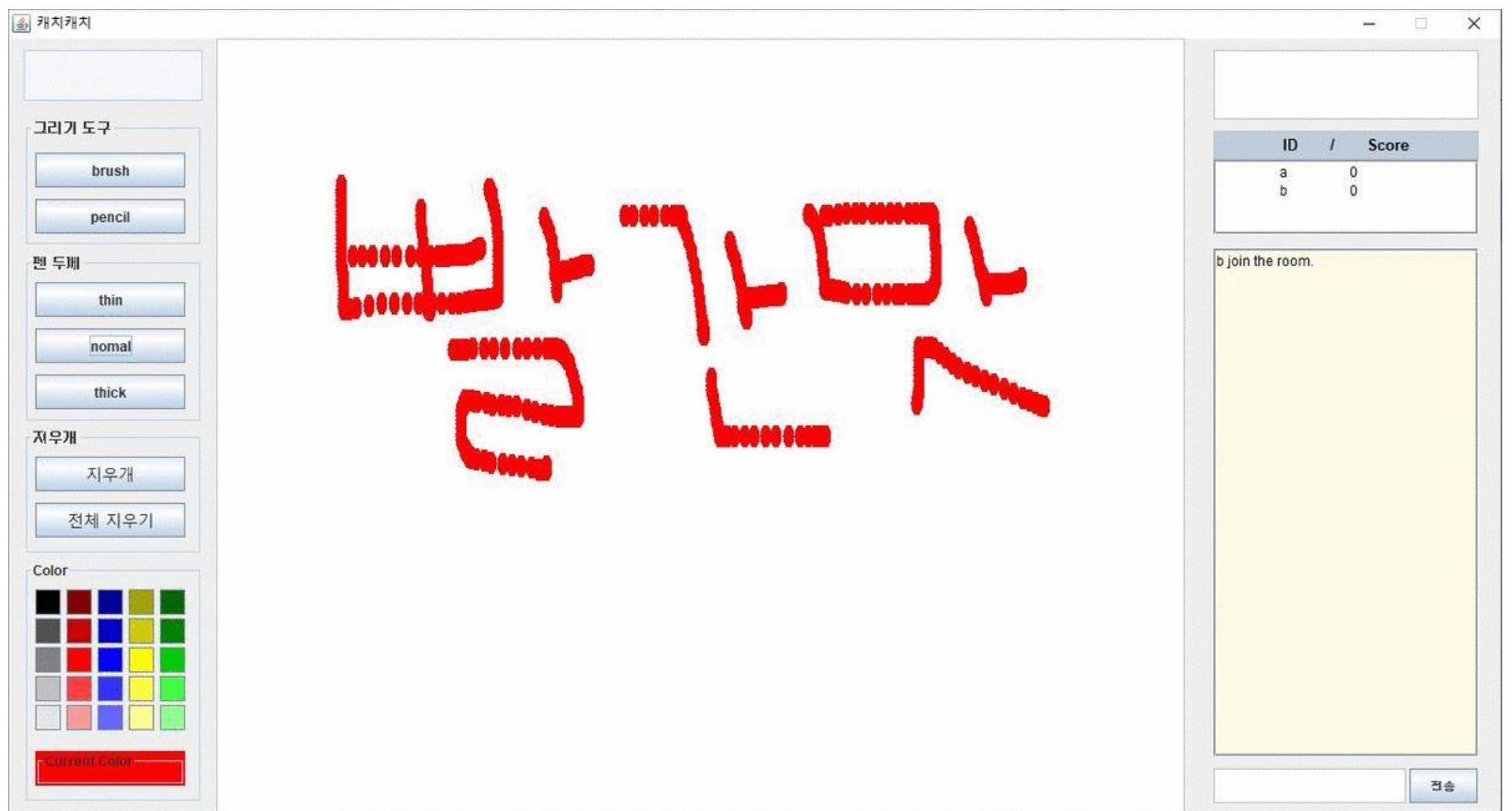
- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29



1. 개요

- StoryBoard

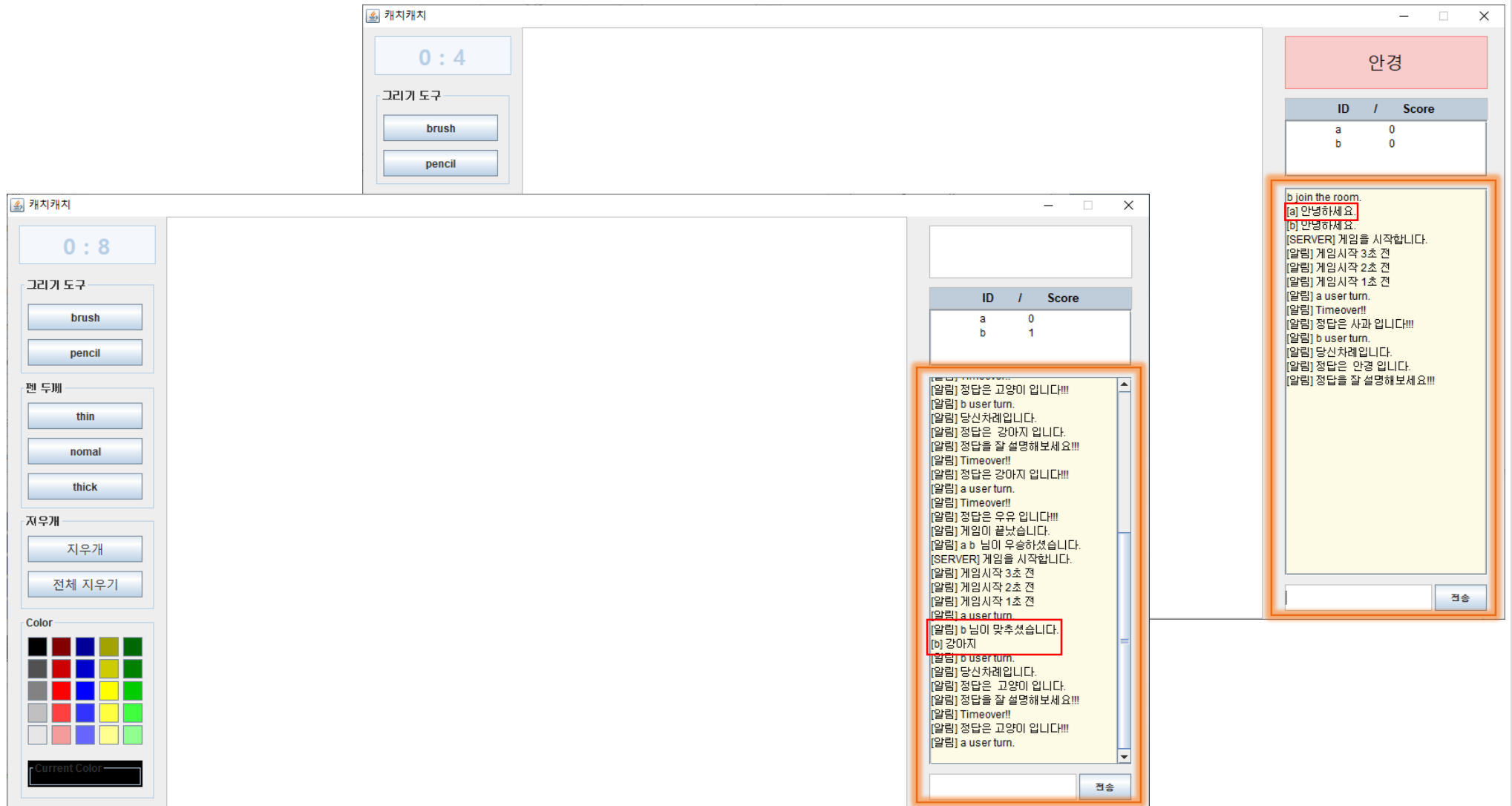
2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29





1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void [redacted] { // 대화내용을 띄울 대화창
2     screen = new JTextArea(); // JTextArea 로 대화창을 구현
3     screen.setBackground(new Color(253, 250, 230));
4     JScrollPane scroll = new JScrollPane(screen); // 대화창에 스크롤바 생성
5     scroll.setBounds(1045, 182, 230, 440);
6     Border screenborder = BorderFactory.createLineBorder(Color.LIGHT_GRAY, 1);
7     screen.setEnabled(false); // 화면 내에서는 수정할수 없도록 enable(false)
8     screen.setDisabledTextColor(Color.BLACK);
9     screen.setFont(screen.getFont().deriveFont(10)); // 현재의 Font 객체를 복제해 새로운 스타일을 적용
10    screen.setBorder(screenborder);
11    screen.setCaretPosition(screen.getDocument().getLength()); // 스크롤바를 제일 아래로 위치
12    frame.getContentPane().add(scroll);
13 }
14 private void [redacted] { // 대화 내용 입력할 텍스트필드
15     input = new JTextField(); // 대화내용을 입력해서 보낼 텍스트필드 생성
16     input.setBounds(1045, 632, 167, 30);
17     Border inputborder = BorderFactory.createLineBorder(Color.LIGHT_GRAY, 1); // 테두리 설정
18     input.setBorder(inputborder);
19     input.addKeyListener(new KeyAdapter() { // 텍스트필드에 키액션 추가
20         @Override
21         public void keyPressed(KeyEvent e) { // 키를 눌렀을때 이벤트 발생
22             if (e.getKeyCode() == KeyEvent.VK_ENTER) { // 엔터기를 눌렀다면 실행
23                 SendMessage.send.println("CHAT:" + input.getText()); // 서버로 메시지("CHAT:" + input.getText()) 전송
24                 SendMessage.send.flush(); // 버퍼를 비움
25                 input.setText(""); // 전송후에 텍스트 필드는 다시 reset(아무값도 없게)
26             }
27         }
28     });
29     sendbtn = new JButton("전송");
30     sendbtn.setBounds(1215, 632, 59, 30);
31
32     sendbtn.setFont(new Font("돋움", Font.BOLD, 10));
33     sendbtn.addActionListener(new ActionListener() { // 전송버튼에도 엔터를 눌렀을때와 같이 서버로 메시지 전송
34         @Override
35         public void actionPerformed(ActionEvent e) {
36             SendMessage.send.println("CHAT:" + input.getText());
37             SendMessage.send.flush();
38             input.setText(""); // 서버로 전송후 텍스트필드 초기화
39         }
40     });
41     frame.getContentPane().add(input);
42     frame.getContentPane().add(sendbtn);
43 }
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 1. 서버쪽에서 처리
2 private void waitMsg() {
3     while (true) {
4         try {
5             msg = userin.readLine();
6             if (msg.contains("CHAT:")) {
7                 msg += " "; // 아무것도 입력하지않고 엔터눌렀을시 멈출방지
8                 String[] pars = msg.split(":");
9                 System.out.println(pars[1]);
10
11                 if (pars[0].equals("CHAT")) { // 입력받은 스트림이 채팅일때
12                     pars[1] += " ";
13                 }
14             }
15         }
16     }
17     allUserSendMsg();
```

```
private void allUserSendMsg() {
    for (int i = 0; i < ServerController.List.size(); i++) {
        ServerController.List.get(i).sendMessage(msg);
    }
    if (msg.contains("CHAT:")) {
        String pars[] = msg.split(":");
        screen.append(pars[1] + "\n");
        screen.setCaretPosition(screen.getDocument().getLength());
    }
}
```

```
17 2. 클라이언트에서 다시 받아와서 처리
18 private void getMsg() {
19     while (true) {
20         try {
21             msg = msgbuff.readLine(); // 서버에서 보내온 메시지를 한줄씩 msg에 담는다.
22             if (msg.contains(":")) { // msg 에 ':'이 포함되어있다면 실행
23                 String[] pars = msg.split(":"); // String 배열 pars에 msg를 ':' 문자 기준으로 나눈 문자열을 담는다.
24                 if (pars[0].equals("CHAT")) { // mainDesign에서 drawChat() 이 호출한 drawInputText() 메서드에서
25                     // keyPressed 액션, 전송버튼 클릭 발생시 [(VK_ENTER) --> 엔터키를 눌렀을때 실행됨] 서버로 "CHAT:" +
26                     // input.getText()" 전송된 메시지, 혹은 서버쪽에서 바로 보내는 메시지를 처리
27                     screen.append(pars[1] + "\n"); // : 문자 기준으로 나눈 뒤의 getText(입력값)을 screen(JTextArea)에 덧붙인다.
28                     screen.setCaretPosition(screen.getDocument().getLength()); // 맨 아래로 스크롤 한다.
29                 }
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

캐치캐치

0 : 8

그리기 도구

brush

pencil

펜 두께

thin

nomal

thick

지우개

지우개

전체 지우기

Color



Current Color

ID / Score

a 0
b 1

[알림] 정답은 고양이 입니다!!!
[알림] b user turn.
[알림] 당신 차례입니다.
[알림] 정답은 강아지 입니다.
[알림] 정답을 잘 설명해보세요!!!
[알림] Timeover!!
[알림] 정답은 강아지 입니다!!!
[알림] a user turn.
[알림] Timeover!!
[알림] 정답은 우유 입니다!!!
[알림] 게임이 끝났습니다.
[알림] a b 님이 우승하셨습니다.
[SERVER] 게임을 시작합니다.
[알림] 게임시작 3초 전
[알림] 게임시작 2초 전
[알림] 게임시작 1초 전
[알림] a user turn.
[알림] b 님이 맞추셨습니다.
[b] 강아지
[알림] b user turn.
[알림] 당신 차례입니다.
[알림] 정답은 고양이 입니다.
[알림] 정답을 잘 설명해보세요!!!
[알림] Timeover!!
[알림] 정답은 고양이 입니다!!!
[알림] a user turn.

전송



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1
2 private void drawUserList() { // 접속중인 유저 목록을 띄울 영역
3     userJPanel = new JPanel();
4     userJPanel.setBounds(1045, 80, 230, 90);
5     userList = new JTextArea();
6     userList.setTabSize(5);
7     JLabel userLabel = new JLabel("ID / Score");
8     userLabel.setOpaque(true); // JLabel에 배경색을 설정하기 위해 사용
9     userLabel.setBackground(SystemColor.inactiveCaption); // 활성 창 제목 표시줄의 텍스트 색
10    userLabel.setHorizontalAlignment(SwingConstants.CENTER);
11    userLabel.setFont(new Font("맑은고딕", Font.BOLD, 13));
12    userLabel.setBorder(BorderFactory.createLineBorder(Color.LIGHT_GRAY, 1));
13    userLabel.setBounds(0, 0, 230, 25);
14    JScrollPane scrol2 = new JScrollPane(userList);
15    userJPanel.setLayout(null);
16    userJPanel.add(userLabel);
17    scrol2.setBounds(0, 25, 230, 65);
18    userList.setEnabled(false);
19    Border screenborder = BorderFactory.createLineBorder(Color.LIGHT_GRAY, 1);
20    userList.setDisabledTextColor(Color.BLACK);
21    userList.setFont(userList.getFont().deriveFont(10));
22    userList.setBorder(screenborder);
23    userList.setCaretPosition(userList.getDocument().getLength());
24
25    userJPanel.add(scrol2);
26    frame.getContentPane().add(userJPanel);
27
28
29 }
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 1. 서버쪽에서 보내줄 메시지
2 private void userListText() { // 유저목록에 띄울 아이디 전송
3     String str = new String();
4     for (int i = 0; i < ServerController.List.size(); i++) { // 유저목록의 사이즈 만큼 반복문 실행
5         str += 
6         ; // String str에 유저 목록에 있는 순서대로 "아이디 점수 ," 형식으로 계속 추가
7     }
8     GameController.allUserMsg("READY:" + str);
9     // GameController.allUserMsg 메서드를 사용해 모든 유저에게 메시지 전송
10 }
11 static public void allUserMsg(String msg) {
12     for (int i = 0; i < ServerController.List.size(); i++) {
13         ServerController.List.get(i).sendMessage(msg);
14     }
15 }
16
17 2. 클라이언트에서 받아온 메시지 처리
18 private void getMsg() {
19     while (true) {
20         try {
21             msg = msgbuff.readLine(); // 서버에서 보내온 메시지를 한줄씩 msg에 담는다.
22             if (msg.contains(":")) { // msg 에 ':'이 포함되어있다면 실행
23                 String[] pars = msg.split(":"); // String 배열 pars에 msg를 ':' 문자 기준으로 나눈 문자열을 담는다.
24                 if (pars[0].equals("READY")) {
25                     pars = pars[1].split(" , ");
26                     userList.setText("");
27                     for (int i = 0; i < pars.length; i++) {
28                         userList.append("\t" + pars[i] + "\n");
29                     }
30                 }
31             }
32         } catch (IOException e) {
33             e.printStackTrace();
34         }
35     }
36 }
```


정답저장

```
Teamproject1 ▾ 📁 src ▾ 📁 Server.Core.game ▾ 🟢 FileRead ▾ • file : File
1 package Server.Core.game;
2
3 import java.io.BufferedReader;
4
5
6
7
8
9
10 public class FileRead { /// 정답을 읽어오는 클래스
11
12     private File file; //파일을 저장하는 클래스
13     private final String dir = "src" + File.separator + "Server" + File.separator + "Core" + File.separator + "Game"
14         + File.separator + "answer.txt"; //정답의 경로를 dir 변수에 저장
15     private ArrayList<String> list; // 정답을 리스트에 저장
16
17     public void read() { // 이 메소드가 실행되면 파일을 읽어서 리스트에 저장 ㅋㅋ
18         makeList();
19         readstart();
20     }
21
22     private void makeList() {
23         list = new ArrayList<String>();
24     }
25
26     private void readstart() {
27         try {
28             file = new File(dir);
29
30             BufferedReader bufReader = new BufferedReader(new FileReader(file));
31             String line = "";
32             while ((line = bufReader.readLine()) != null) {
33                 list.add(line);
34             }
35             bufReader.close();
36         } catch (FileNotFoundException e) {
37             System.out.println("Working Directory = " + System.getProperty("user.dir"));
38         } catch (IOException e) {
39             System.out.println(e);
40         }
41     }
42
43     public ArrayList<String> getAnswer() {
44         return this.list;
45     }
46 }
```

```
1 package Server.Core.game;
2
3 import java.util.ArrayList;
4
5 public class Game {
6
7     private FileRead file; // 파일리드 클래스 변수
8     private static ArrayList<String> answerList; // static변수로 설정
9     private String answer;
10
11     public void start() { |
12         readFile();
13         saveAnswer();
14     }
15
16     private void readFile() { // 파일리드 클래스를 정의해서 정답을 읽어들임
17         file = new FileRead();
18         file.read();
19     }
20
21     private void saveAnswer() { // 읽은 파일을 static변수 정답목록에 저장
22         answerList = file.getAnswer();
23     }
24
25     public void print() { // 정답을 순서대로 출력하는 메소드 // 여기서 실행 시키지 않음
26         for (int i = 0; i < answerList.size(); i++) {
27             System.out.println(answerList.get(i));
28         }
29     }
30
31     public boolean hasMoreAnswer() { // 정답리스트에서 정답을 랜덤으로 추출하고 지우는 메소드 // 여기서 실행시키지 않음
32         if (answerList.size() != 0) {
33             int index = (int) (Math.random() * answerList.size());
34             answer = answerList.get(index);
35             answerList.remove(index);
36             return true;
37         } else
38             return false;
39     }
40
41     public String getAnswer() {
42         return this.answer;
43     }
44 }
```

Teamproject1 ▸ src ▸ Server.Core.game ▸ GameloopThread ▸ run() : void

```
27 }
28 }
29
30 public class GameloopThread extends Thread {
31
32     private JTextArea screen;
33
34     @Override
35     public void run() {
36         Game game = new Game(); // 게임 객체를 생성하고
37
38         String answer, Userstr;
39         int index = 0; // 인덱스 0
40         TimeloopThread time;
41         game.start(); // 게임스타트 메소드를 실행시킨다. 이게 실행되면 정답을 읽어들인다.
42         GameController.firstStart();
43     }
```

```
75     }
76 }
77
78 private void StartEvent() { // 시작 버튼을 눌렀을때 (주호)
79     startbtn.addActionListener(new ActionListener() {
80         @Override
81         public void actionPerformed(ActionEvent e) {
82             if (checkStart()) {
83                 screen.append("[ERROR] 최소 두명 이상 접속 시 게임 시작이 가능합니다.\n");
84                 screen.setCaretPosition(screen.getDocument().getLength());
85             } else {
86                 gamestart(); // 게임 루핑스레드를 실행 시킨다.
87             }
88         }
89     });
90 }
91
92 private boolean checkStart() {
93     if (List.size() > 1)
94         return false;
95     else
96         return true;
97 }
98
99 private void gamestart() {
100     screen.append("[SERVER] Start the game.\n");
101     screen.setCaretPosition(screen.getDocument().getLength());
102     GameloopThread game = new GameloopThread(); // 정답을 읽어온다.
103     game.setScreen(screen);
104     if (GameController.gameflag == false) { // 게임 시작이 안되었을때만 게임시작 가능하게 함
105         game.start();
106         GameController.gameflag = true;
107     } else {
108         JOptionPane.showMessageDialog(null, "게임이 시작되었습니다.");
109     }
```

턴에 대한 설명과 권한

Teamproject1 ▾ src ▾ Client.Core ▾ GameController ▾

```
1 package Client.Core;
2
3 public class GameController {
4     static public boolean gamestate = true; // 게임시작여부
5     static public boolean turnflag = true; // 이 값의 초기값이 TURE이기 때문에 모든 유저가 그림을 그릴 수 있다(주호중요)
6 }
7 |
```

Teamproject1 ▾ src ▾ Client.Design ▾ MainDesign ▾ makeMouseEvent() : void ▾ new MouseMotionListener() {...} ▾ mouseDragged(MouseEvent) : void

```
109 //주호 중요
110 private void makeMouseEvent() {
111     imgpanel.addMouseMotionListener(new MouseMotionListener() {
112         @Override
113         public void mouseDragged(MouseEvent e) { // 마우스 드래그시//유저게임컨트롤러의 값이
114             if (GameController.turnflag == true) { // 만약 내턴이라면 서버에 보낸다.(주호 중요)
115                 SendMessage.send.println("Position:" + e.getX() + "," + e.getY()); // 서버로 좌표값 전송
116                 SendMessage.send.flush();
117             }
118         }
119
120     public void mouseMoved(MouseEvent e) {
121     }
122     });
123 }
124 }
```

Teamproject1 ▸ src ▸ Client.Core.socket ▸ ReceiveMessage ▸ getMsg0 : void

```

}
    }

private void getMsg() {
    while (true) {
        try {
            msg = msgbuff.readLine(); // sever.getInputStream()을 얻어와서 입력받은 걸 한줄씩 msg에 담는다.
            if (msg.contains(":")) { // msg 에 ':'이 포함되어있다면 실행
                String[] pars = msg.split(":"); // pars배열에 ':'문자 기준으로 나눈 문자열을 담는다.

                //-----
                if (pars[0].equals("Position")) { // Position : x,y ->좌표 값을 지정해주는 구문
                    // mainDesign 에서 (GameController.turnflag == true)일때
                    pars = pars[1].split(",");
                    // ':' 기준으로 나눈 문자열 인덱스 1번을 ',' 문자 기준으로 나눠서 pars배열에 다시 담은(이전 문자배열 사라짐)
                    x = Integer.parseInt(pars[0]); // ',' 문자 기준으로 pars[0] == x좌표값
                    y = Integer.parseInt(pars[1]); // ',' 문자 기준으로 pars[1] == y좌표값
                    brush.setX(x); // brush 객체에 setX() 메서드를 호출하여 x좌표값을 지정
                    brush.setY(y); // brush 객체에 setY() 메서드를 호출하여 y좌표값을 지정
                    brush.repaint(); // brush객체의 repaint()메서드를 호출하여 초기화(repaint는 기본정의된 메서드)
                    // 그린다.(주호)
                    brush.printAll(imgbuff.getGraphics()); // imgbuff의 Graphics를모두 그린다.(주호)
                }

                //-----
                else if (pars[0].equals("Color")) {
                    if (pars[1].equals("BLACK")) // : 로 나눈 인덱스 1번이 brack일 경우
                        brush.setColor(Color.BLACK); // brush클래스 setColor(Color color) 메소드로 색을 설정
                    else if (pars[1].equals("AUTOGRAYDEEP"))
                        brush.setColor(new Color(81, 81, 81));
                    else if (pars[1].equals("GRAY"))
                        brush.setColor(new Color(130, 130, 130));
                    else if (pars[1].equals("SILVER"))
                        brush.setColor(new Color(192, 192, 192));
                    else if (pars[1].equals("BRIGHTSILVER"))
                        brush.setColor(new Color(230, 230, 230));
                    else if (pars[1].equals("DARKREDBTN"))
                        brush.setColor(new Color(194, 0, 0));
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Teamproject1 ▸ src ▸ Client.Core.socket ▸ ReceiveMessage ▸ getMsg() : void

```
156         if (pars[1].equals("CLEAR")) // "MODE: CLEAR"를 처리하여
157             ClearScreen(); // ClearScreen() 메서드 실행
158 //-----
159     } else if (pars[0].equals("SET")) { // (주호 중요) , 읽은 값이
160         if (pars[1].equals("FALSE")) { // 서버에서 받아와서 set : false 라면
161             GameController.turnFlag = false; // 턴 플레그를 FALSE로 설정한다.
162             answerfield.setBackground(new Color(255, 255, 255));
163             answerfield.setBorder(BorderFactory.createLineBorder(Color.LIGHT_GRAY, 1));
164
165         } else if (pars[1].equals("TRUE")) { // 서버에서 받아와서 set : TURE 라면
166             GameController.turnFlag = true; // TURNFLAG를 TURE로 설정한다.
167         }
168         if (pars[1].equals("ANSWERON")) { // 서버에서 받아와서 set : ANSWERON 이라면
169             answerfield.setBackground(new Color(249, 204, 204));
170             answerfield.setBorder(BorderFactory.createLineBorder(new Color(249, 137, 137), 1));
171         }
172 //-----
173     } else if (pars[0].equals("ANSWER")) { // 서버에서 정답 받아옴(주호)
174         answerfield.setText(pars[1]); // mainDesign - drawAnswerField 오른쪽 위에 저장
175         // 받아온 정답을 화면에 출력
176 //-----
```

```

54
55 }
56 while (game.hasMoreAnswer()) { // 정답이 리스트에 아직 있다면 //정답이 있을때 까지 돌기
57     answer = game.getAnswer(); // 여기서 정답을 가져오는 것을 실행을 시킨다. (주호) answer 변수에 저장
58     time = new TimeloopThread(); // 타임 루프스레드 생성
59     time.timeChk = false;
60     time.start(); // 타임루프 스레드 실행
61     GameController.answer = answer; // 게임컨트롤러에 있는 answer변수에 저장하여 정답을 비교
62     System.out.println(answer); // 정답을 출력
63     GameController.gameflag = true; // 게임을 시작하는지 여부
64     GameController.answerflag = false; // 정답 flag를 false로 설정
65     GameController.ID = ServerController.List.get(index).getUserID(); // 서버컨트롤러에 있는유저들 중 0번을 게임 컨트롤러 id에 저장
66     GameController.allUserMsg("CHAT:[알림] " + GameController.ID + " user turn."); // 모든 유저에게 턴을 알려줌 (주호)
67                                     // ServerController.List.get(i).sendMessage(msg);
68     screen.append("[알림] " + GameController.ID + " user turn.\n"); // 서버 채팅창에 턴을 알려줌
69     // 게임컨트롤러가 모든유저에게 보내는 메소드들을 정의해 놓음
70     screen.setCaretPosition(screen.getDocument().getLength()); // 서버 스크롤을 제일 아래로
71     // 0번 유저에게 매세지 출력
72     ServerController.List.get(index).sendMessage("SET:TRUE"); // 턴을 넘겨준다. 이것으로 출제자를 정함 (주호중요)
73     ServerController.List.get(index).sendMessage("SET : ANSWERON");
74     ServerController.List.get(index).sendMessage("CHAT:[알림] " + "당신차례입니다.");
75     ServerController.List.get(index).sendMessage("CHAT:[알림] " + "정답은 " + answer + " 입니다.");
76     ServerController.List.get(index).sendMessage("CHAT:[알림] " + "정답을 잘 설명해보세요!!!");
77     ServerController.List.get(index).sendMessage("ANSWER:" + answer);
78     // 중첩while문
79     while (true) { // 무한 반복문 정답이라면 빠져나와서 정답 리정답이 아니라면 점수표시와
80         if (GameController.answerflag == true) { // 정답이라면 나오기
81             Userstr = new String();
82             for (int j = 0; j < ServerController.List.size(); j++) { // userstr에 모든유저 아이디와 점수 + , 를 출력하고
83                 Userstr += ServerController.List.get(j).getUserID() + "\t"
84                     + ServerController.List.get(j).getScore() + " , ";
85             }
86             GameController.allUserMsg("READY:" + Userstr); // 게임컨트롤러에 모든 유저메세지로 ready : userstr 출력
87             time.stop();
88             break; //

```


정답체크

Teamproject1 ▾ src ▾ Server.Core.socket ▾ ServerThread ▾ waitMsg() : void

```
44 private void waitMsg() {
45     //무한 반복문(주회)
46     while (true) {
47         try {
48             msg = userin.readLine(); //입력된 값을 읽어온다.(주회)
49             //-----
50             if (msg.contains("CHAT:")) {
51                 msg += " "; // 아무것도 입력하지않고 엔터눌렀을시 멈춤방지
52                 String[] pars = msg.split(":");
53                 System.out.println(pars[1]); // CHAT : 뒤에 부분을 읽어온다.
54                 // 만약 게임컨트롤의 정답과 같다면// 뒤쪽에 다른 조건들은 버그를 방지하기 위해서
55                 // 엔서플레그와 아이디의 유효성검사를 한다.
56                 if (pars[1].equals(GameController.answer + " ") && GameController.answerflag == false
57                     && (!(this.ID).equals(GameController.ID))) {
58                     GameController.answerflag = true; // 정답을 맞출으로 표시
59                     GameController.rightAnswer(ID); // right메소드는 모든 사용자에게 모든 메시지를 보낸다.
60                     for (int i = 0; i < ServerController.List.size(); i++) {
61                         if (ServerController.List.get(i).getUserID().equals(this.ID)) { // 현재아이뒤와 중복되는 유저에게 점수를 주는 메소드
62                             ServerController.List.get(i).setScore(ServerController.List.get(i).getScore() + 1);
63                         }
64                     }
65                 }
66                 //아무 것도 입력하지 않았을 때 표시
67                 if (pars[0].equals("CHAT")) { // 입력받은 스트림이 채팅일때
68                     pars[1] += " ";
69                     msg = "CHAT:" + "[" + ID + " ] " + pars[1];
70                 }
71             }
72         }
73     }
74 }
```

Teamproject1 ▸ src ▸ Server.Core.game ▸ GameController ▸ rightAnswer(String) : void

```
25     }
26
27     static public void allUserPermissionFalse() {
28         for (int i = 0; i < ServerController.List.size(); i++) {
29             ServerController.List.get(i).sendMessage("SET:FALSE");
30             ServerController.List.get(i).sendMessage("MODE:CLEAR");
31         }
32     }
33
34     static public void rightAnswer(String id) { // 정답을 맞추면 모든 사용자에게 보내는 메세지 (주호)
35         answerflag = true; // 정답표시를 true로 수정
36         answer = ""; // (주호) aaa있던걸 수정
37         for (int i = 0; i < ServerController.List.size(); i++) {
38             ServerController.List.get(i).sendMessage("CHAT:[알림] " + id + " 님이 맞추셨습니다.");
39         }
40     }
41
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

캐치캐치

0 : 9

그리기 도구

brush

pencil

펜 두께

thin

nomal

thick

지우개

지우개

전체 지우기

Color

Current Color

a

ID	/	Score
1	2	
4	0	

[SERVER] 게임을 시작합니다.
[알림] 게임시작 3초 전
[알림] 게임시작 2초 전
[알림] 게임시작 1초 전
[알림] 1 user turn.
[알림] 당신차례입니다.
[알림] 정답은 a 입니다.
[알림] 정답을 잘 설명해보세요!!!
[알림] Timeover!!
[알림] 정답은 a 입니다!!!
[알림] 4 user turn.
[알림] 1 님이 맞추셨습니다.
[1] a
[알림] 1 user turn.
[알림] 당신차례입니다.
[알림] 정답은 a 입니다.
[알림] 정답을 잘 설명해보세요!!!
[알림] Timeover!!
[알림] 정답은 a 입니다!!!
[알림] 4 user turn.
[알림] 1 님이 맞추셨습니다.
[1] a
[알림] 1 user turn.
[알림] 당신차례입니다.
[알림] 정답은 a 입니다.
[알림] 정답을 잘 설명해보세요!!!

전송



1. 개요

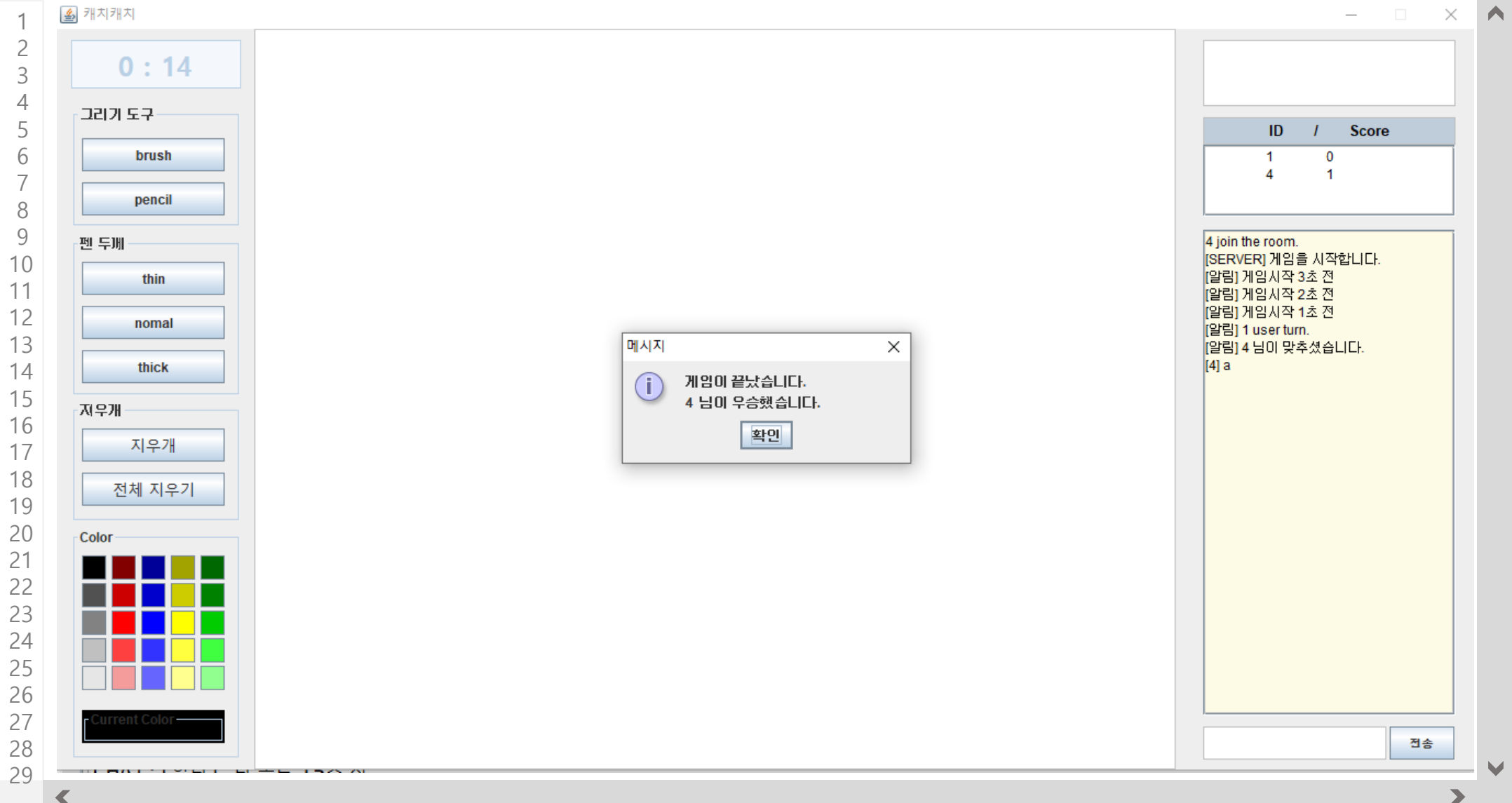
- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행





File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void StartEvent() {
2     startbtn.addActionListener(new ActionListener() {
3         @Override
4         public void actionPerformed(ActionEvent e) {
5             if (checkStart()) {
6                 screen.append("[ERROR] 최소 두명 이상 접속 시 게임 시작이 가능합니다.\n");
7                 screen.setCaretPosition(screen.getDocument().getLength());
8             } else {
9                 gamestart();
10            }
11        }
12    });
13 }
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 private void gamestart() {
2     screen.append("[SERVER] Start the game.\n");
3     screen.setCaretPosition(screen.getDocument().getLength());
4     GameloopThread game = new GameloopThread();
5     game.setScreen(screen);
6     game.start();
7 }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1
2
3  @Override
4  public void run() {
5      Game game = new Game();
6      String answer, Userstr;
7      int index = 0;
8      TimeLoopThread time;
9
10     game.start(); // 정답읽어서 어레이리스트에 저장
11     GameController.firstStart(); // 처음 게임 시작하면 점수 및 승자 초기화
12     for (int i = 3; i > 0; i--) { // 3초뒤 시작
13         try {
14             screen.append("CHAT:[알림] 게임시작 " + i + "초 전\n");
15             GameController.allUserMsg("CHAT:[알림] 게임시작 " + i + "초 전");
16             screen.setCaretPosition(screen.getDocument().getLength()); // 스크롤을 제일 아래로
17             Thread.sleep(1000);
18         } catch (InterruptedException e) {
19             // TODO Auto-generated catch block
20             e.printStackTrace();
21         }
22     }
23
24     ...
25
26 }
27
28
29
```




1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 while (game.hasMoreAnswer()) { // 가져올 문제가 있으면 true 없으면 false
2     // 정답목록에서 출제할 문제를 랜덤으로 하나 가져와서 가져온 정답은 리스트에서 삭제
3
4     answer = game.getAnswer(); // 문제가져옴
5     time = new TimeloopThread(); // TimeloopThread 객체 생성
6     time.timeChk = false; // 초깃값 시간이 끝나지 않음을 의미
7     time.start(); // 시간 카운트다운 시작
8     GameController.answer = answer;
9
10    // System.out.println(answer);
11
12    GameController.gameflag = true; // 게임시작
13    GameController.answerflag = false; // 정답자없음(게임시작 초깃값)
14
15    GameController.ID = ServerController.List.get(index).getUserID(); // 순서대로 아이디(출제자) 가져옴
16    GameController.allUserMsg("CHAT:[알림] " + GameController.ID + " user turn."); // 출제자 알림
17    screen.append("[알림] " + GameController.ID + " user turn.\n");
18    screen.setCaretPosition(screen.getDocument().getLength()); // 스크롤을 제일 아래로
19    ServerController.List.get(index).sendMessage("SET:TRUE"); // index에 해당하는 유저만 그림 권한 부여받음
20    ServerController.List.get(index).sendMessage("SET:ANSWERON"); // 정답란 빨간표시
21    ServerController.List.get(index).sendMessage("CHAT:[알림] " + "당신차례입니다."); // 알림
22    ServerController.List.get(index).sendMessage("CHAT:[알림] " + "정답은 " + answer + " 입니다.");
23    ServerController.List.get(index).sendMessage("CHAT:[알림] " + "정답을 잘 설명해보세요!!!");
24    ServerController.List.get(index).sendMessage("ANSWER:" + answer);
25
26
27
28
29
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



11



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 public boolean hasMoreAnswer() { // 가져올 문제가 있는지 검사 있으면 true 없으면 false
2     if (answerList.size() != 0) { // 남은 문제 목록의 사이즈가 0이 아닌경우 (있을때)
3         int index = (int) (Math.random() * answerList.size()); // 0~목록의 크기중 랜덤으로 인덱스 선택
4         answer = answerList.get(index); // 랜덤으로 선택한 인덱스를 정답으로
5         answerList.remove(index); // 해당 인덱스에 위치한 정답 목록 삭제
6         return true; // 있으면 true
7     } else
8         return false; // 출제할 문제 없으면 false
9 }
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1  class TimeloopThread extends Thread {
2      static boolean timeChk = false;
3
4
5      @Override
6      public void run() {
7          String Time;
8          for (int i = 90; i >= 0; i--) { // i 초깃값이 게임중 턴에대한 시간제한(초단위)
9              try {
10                  Time = i / 60 + ":" + i % 60; // 분과 초 를 나타냄
11                  System.out.println("CHAT:[알림] 턴 종료 " + i + "초 전");
12                  GameController.allUserMsg("TIME:" + Time); // 모든 유저에게 시간이 얼마 남았는지 알려줌
13                  if (i == 0) {
14                      timeChk = true; // 시간 카운트 다운이 끝났을때 timeChk true로 바꿈
15                  } else {
16                      Thread.sleep(1000); // 1초씩 쉼
17                  }
18              } catch (InterruptedException e) {
19                  e.printStackTrace();
20              }
21          }
22      }
23  }
24
25
26
27
28 }
```



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 while (true) { // 정답 혹은 시간이 끝날때까지 무한루프
2     if (GameController.answerflag == true) { // 정답이 나오면 ( 정답 검사는 ServerThread의 waitMsg에서 판단 후 점수 더함)
3         Userstr = new String();
4         for (int j = 0; j < ServerController.List.size(); j++) {
5             Userstr += ServerController.List.get(j).getUserID() + "\t"
6                 + ServerController.List.get(j).getScore() + " , "; // 유저 아이디와 점수목록을 가져옴
7         }
8         GameController.allUserMsg("READY:" + Userstr); // 모든 유저에게 다시 모두 뿌림
9         time.stop(); // 시간 카운트다운 (스레드) 멈춤
10        break; // 무한루프 나감
11    } else {
12        try {
13            if (time.timeChk == true) { // 시간 카운트제한이 끝났을때 무한루프 끝냄
14                GameController.allUserMsg("CHAT:[알림] Timeover!!");
15                GameController.allUserMsg("CHAT:[알림] 정답은 " + GameController.answer + " 입니다!!!");
16                screen.append("CHAT:[알림] Timeover!!\n");
17                screen.append("CHAT:[알림] 정답은 " + GameController.answer + " 입니다!!!\n");
18                screen.setCaretPosition(screen.getDocument().getLength()); // 스크롤을 제일 아래로
19                break; // 무한루프 나감
20            } else {
21                Thread.sleep(1000);
22            }
23        } catch (InterruptedException e) {
24        }
25    }
26    ServerController.List.get(index).sendMessage("ANSWER:" + " "); // 전 출제자였던 사람 정답란 초기화
27    GameController.allUserPermissionFalse(); // 그림판 초기화
28    ++index; // 다음 사람
29    if (index == ServerController.List.size()) { // 다음사람에 대한 인덱스가 유저 크기와 같으면
30        index = 0; // 다시 처음으로
31    }
32    // while(game.hasMoreAnswer()) 정답목록 없음
33    GameController.gameRank(); // 문제가 모두 출제 후 게임끝 우승자 누구인지 알려주는 메서드
34    screen.append("[알림] 게임이 끝났습니다.\n"); // 서버창에 메시지 출력
35    screen.append("[알림] " + GameController.winner + "님이 우승하셨습니다.\n");
36    screen.setCaretPosition(screen.getDocument().getLength()); // 스크롤을 제일 아래로
37    // 스레드 끝
```



File Edit Source Refactor Navigate Search Project Run Window Help

Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

```
1 static public void gameRank() {
2
3     temp = ServerController.List.get(0).getScore(); // 첫 번째 유저 점수 저장
4     for (int i = 1; i < ServerController.List.size(); i++) { // 유저 모두 비교
5         temp = temp > ServerController.List.get(i).getScore() ? temp : ServerController.List.get(i).getScore();
6     }
7
8     for (int i = 0; i < ServerController.List.size(); i++) {
9         if (temp == ServerController.List.get(i).getScore()) { // 가장 높은 점수인 temp와 유저 점수가 같으면
10             Winner += ServerController.List.get(i).getUserID() + " "; // 해당 점수를 가진 유저의 아이디 가져옴
11         }
12     }
13
14     allUserMsg("Winner:" + GameController.Winner); // Winner: id 와 같은 형태로 모든 유저에게 메시지 보냄 (우승자가 누군지
15                                     // SHOWMESSAGEIALOG형태로 알려줌)
16
17     allUserMsg("CHAT:[알림] 게임이 끝났습니다."); // 게임 끝 알림 메시지
18     allUserMsg("CHAT:[알림] " + GameController.Winner + " 님이 우승하셨습니다."); // 우승 알림메시지
19     GameController.gameflag = false; // 게임중이 아님을 설정(게임끝)
20 }
21
22
23
24
25
26
27
28
29
```



Contents



Storyboard



Client



Server



Start



1. 개요

- StoryBoard

2. 프로그램 구성도

- Client

- Server

3. 프로그램 실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

Q&A



Contents



Storyboard



Client



Server



Start



11



1. 개요
- StoryBoard

2. 프로그램
구성도
- Client
- Server

3. 프로그램
실행

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

게임 실행!!