[참고] 클라이언트에서 그리는 그림을 서버에서 받아서 다시 그리는 예제.

1. Client

**public** **class** **Test** **extends** **Frame** **implements** MouseMotionListener {

**private** **Socket** socket;

**private** **PrintWriter** writer;

**private** **BufferedReader** reader;

**private** **Image** img = **null**;

**private** **Graphics** gImg = **null**;

**private** **int** x = 0;

**private** **int** y = 0;

**public** Test(**String** title) {

// **TODO** Auto-generated constructor stub

**super**(title);

addMouseMotionListener(**this**);

addWindowListener(**new** **WindowAdapter**() {

**public** **void** windowClosing(**WindowEvent** e) {

**System**.*exit*(0);

}

});

setBounds(100, 100, 500, 500);

setVisible(**true**);

img = createImage(500, 500);

gImg = img.getGraphics();

repaint();

**try** {

socket = **new** **Socket**("117.17.158.165", 10001);

writer = **new** **PrintWriter**(**new** **OutputStreamWriter**(

socket.getOutputStream()));

} **catch** (**Exception** e) {

}

}

**public** **static** **void** main(**String**[] args) {

**new** **Test**("Client");

}

**public** **void** paint(**Graphics** g) {

g.drawImage(img, 0, 0, **this**);

}

@Override

**public** **void** mouseDragged(**MouseEvent** e) {

// **TODO** Auto-generated method stub

gImg.drawLine(x, y, e.getX(), e.getY());

x = e.getX();

y = e.getY();

repaint();

writer.println("DRAW:" + x + "\_" + y);

writer.flush();

}

@Override

**public** **void** mouseMoved(**MouseEvent** e) {

// **TODO** Auto-generated method stub

x = e.getX();

y = e.getY();

writer.println("MOVE:" + x + "\_" + y);

writer.flush();

}

}

2. Server

**public** **class** **MainClass** **extends** **Frame** {

**private** **static** **ServerSocket** *serverSocket*;

**private** **static** **Socket** *socket*;

**private** **static** **BufferedReader** *reader*;

**private** **int** x = 0;

**private** **int** y = 0;

**private** **Image** img = **null**;

**private** **Graphics** gImg = **null**;

**public** MainClass(**String** title) {

// **TODO** Auto-generated constructor stub

**super**(title);

addWindowListener(**new** **WindowAdapter**() {

**public** **void** windowClosing(**WindowEvent** we) {

**System**.*exit*(0);

}

});

setBounds(100, 100, 500, 500);

setVisible(**true**);

img = createImage(500, 500);

gImg = img.getGraphics();

repaint();

**try** {

*serverSocket* = **new** **ServerSocket**(10001);

*socket* = *serverSocket*.accept();

*reader* = **new** **BufferedReader**(**new** **InputStreamReader**(

*socket*.getInputStream()));

**System**.*out*.println("연결됨");

} **catch** (**IOException** e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**new** **Thread**(**new** Runnable() {

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**String** msg = "";

**try** {

**while** ((msg = *reader*.readLine()) != **null**) {

**String**[] tokens = msg.split(":");

**if** (tokens[0].equals("DRAW")) {

**String**[] xy = tokens[1].split("\_");

gImg.drawLine(x, y, **Integer**.*parseInt*(xy[0]), **Integer**.*parseInt*(xy[1]));

x = **Integer**.*parseInt*(xy[0]);

y = **Integer**.*parseInt*(xy[1]);

repaint();

} **else** **if** (tokens[0].equals("MOVE")) {

**String**[] xy = tokens[1].split("\_");

x = **Integer**.*parseInt*(xy[0]);

y = **Integer**.*parseInt*(xy[1]);

}

}

} **catch** (**Exception** e) {

}

}

}).start();

}

**public** **void** paint(**Graphics** g) {

g.drawImage(img, 0, 0, **this**);

}

**public** **static** **void** main(**String**[] args) {

**new** **MainClass**("Server");

}

}