

네트워크프로그래밍-10주

네트워크 게임

정인환교수

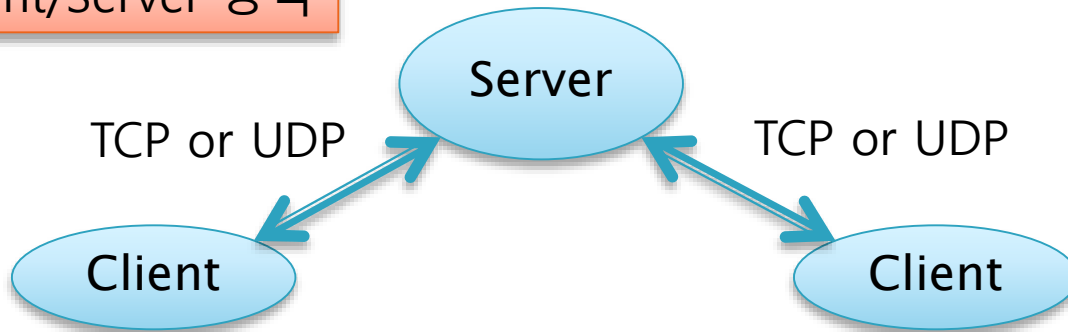
10주 네트워크 게임 프로그래밍

- ▶ Game Client Server 구조
- ▶ Game Client Server 구현 방법 - c언어 사용 예
- ▶ Java Game 예 - 슈팅게임
- ▶ JavaGameClient/Server Drawing 화면 공유
- ▶ 과제
 - c 언어 game 실습
 - JavaGameClilent/Server Drawing 공유 실습
 - JavaGameClient/Server 기능 추가
 - 텀프로젝트 설계서

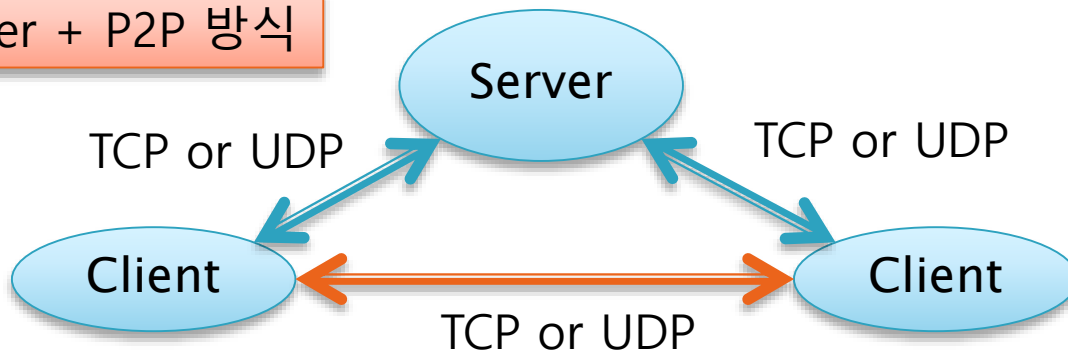
Game Client Server 구조

- ▶ TCP or UDP
- ▶ Client/Server or Client/Server + P2P or P2P

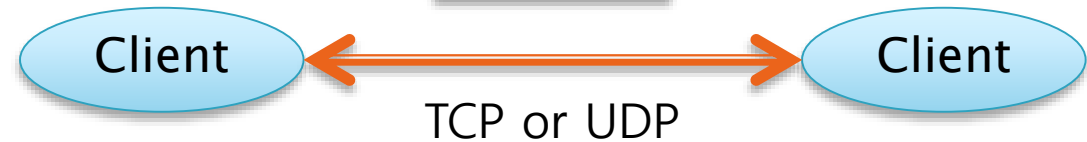
Client/Server 방식



Client/Server + P2P 방식



P2P 방식



네트워크 게임 Client / Server 구현 방법

▶ 게임서버

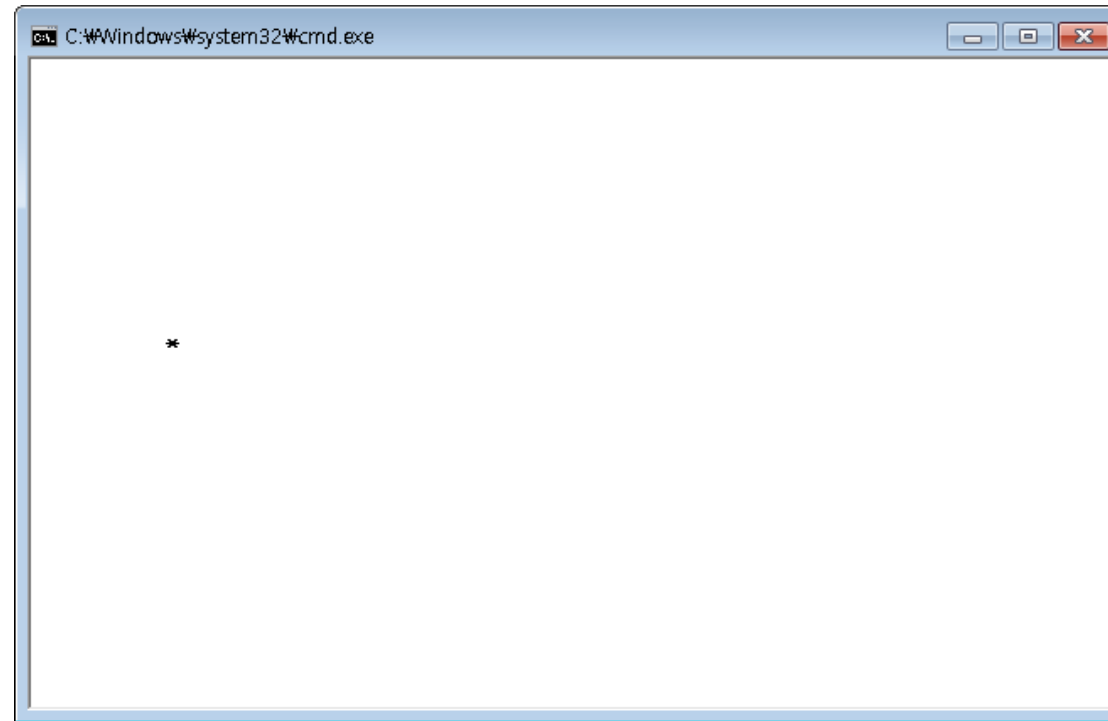
- 1단계: 채팅 서버를 그대로 응용
- 2단계: 게임에 따라 변형
 - 사용자 관리
 - Session 관리
 - 상태관리 (점수, 위치, 아이템, 최종상태, ..)
 - 기타

▶ 게임 클라이언트

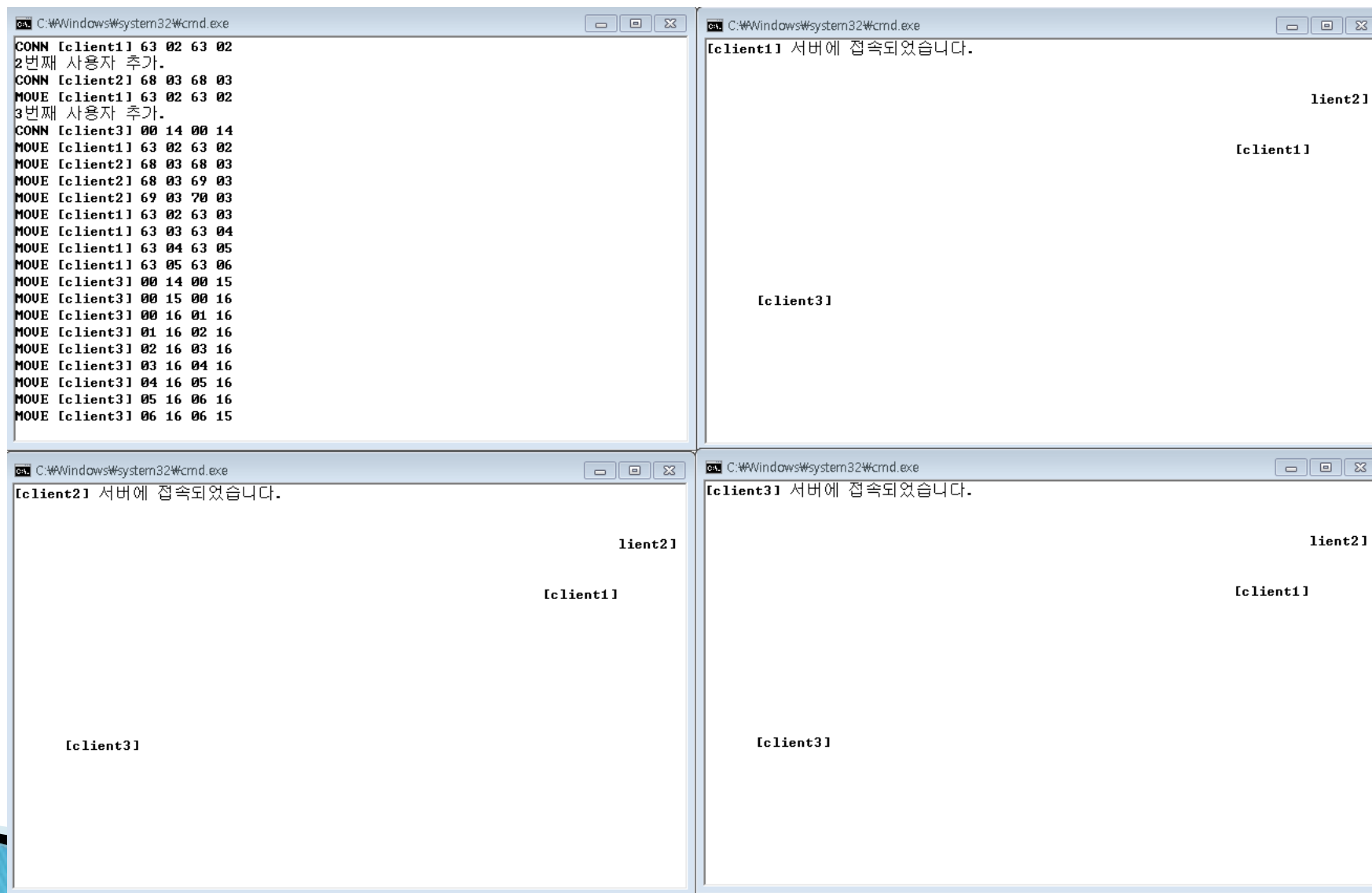
- 1단계: 채팅 서버를 그대로 이용
- 2단계: 게임서버 수정 + 클라이언트 수정
 - 게임을 위한 네트워크 프로토콜 정의 필요

sample1 : 게임 화면 제어 예

- ▶ single user
- ▶ 방향키 또는 AWSD 로 이동
- ▶ gotoxy() 이용



game_client1 : 채팅구조 그대로 이용
sample1.c 이용 자신의 위치를 MOVE oldx oldy newx newy 로 전달.



game_client1 : 게임 프로토콜 전달

- ▶ client 의 이동을 채팅 메시지로 전달
 - CONN NickName oldx oldy newx newy
 - 게임 서버에 접속하면 최초 한번 전달
 - MOVE NickName oldx oldy newx newy
 - 이동할 때 마다 내 위치를 전달
 - client 들은 다른 player의 위치를 표시
- ▶ `void sendmypos(int sd, char *name, int oldx, int oldy, int newx, int newy)`

```
{  
    char buf[512];  
    sprintf(buf, "%s %s %02d %02d %02d %02d", MOVE, name, oldx, oldy, newx, newy);  
    send(sd, buf, BUF_LEN, 0); // NULL 포함 전송.  
}
```
- ▶ `void movename(char *name, int oldx, int oldy, int newx, int newy)`

```
{  
    erasename(oldx, oldy, name);  
    showname(newx, newy, name);  
    fflush(stdout);  
}
```

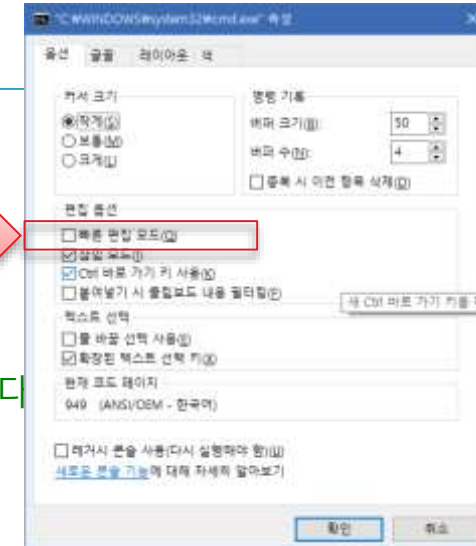
game_client2 : keyboard, mouse 동시 처리

▶ game_client1 의 프로토콜은 그대로 이용

- CONN Nickname oldx oldy newx newy
- MOVE Nickname oldx oldy newx newy

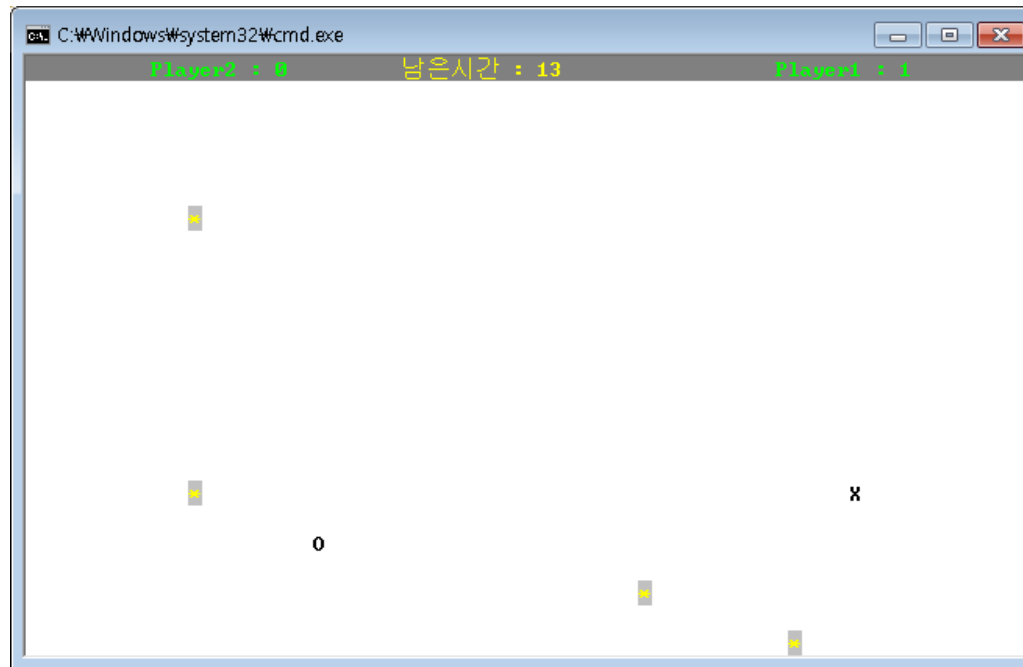
▶ keyboard / mouse event 처리

- `GetNumberOfConsoleInputEvents(hIn, (LPDWORD)&nevents);`
- `if (nevents==0) // Event 가 없으면 읽으려 들어갈 필요가 없다`
- `continue;`
- `ReadConsoleInput(hIn,&rec,1,&dwNOER);`
- `if (rec.EventType == MOUSE_EVENT)`
 - `if (rec.Event.MouseEvent.dwButtonState & FROM_LEFT_1ST_BUTTON_PRESSED)`
 - {
 - `newx = rec.Event.MouseEvent.dwMousePosition.X;`
 - `newy = rec.Event.MouseEvent.dwMousePosition.Y;`
 - `move_flag = 1;`
 - }
- `if (rec.EventType == KEY_EVENT && rec.Event.KeyEvent.bKeyDown == TRUE) {`
- `// 내가 방향키 누르면.`
 - `switch(rec.Event.KeyEvent.wVirtualKeyCode) {`
 - `case VK_UP:`
 - `case VK_DOWN:`
 - `case VK_LEFT:`
 - `case VK_RIGHT:`



게임예 1 – 네트워크 없이 실행

- ▶ game_nonet.c
 - Player 1: 방향키로 이동
 - Player 2: ASWD 로 이동
 - 한번 움직이면 계속 움직인다.
 - 벽을 만나면 반대로 방향 전환을 한다.



game_client3 : 1:1 게임 구조 예

- ▶ game_nonet 의 network 버전
- ▶ 채팅서버를 그대로 이용
 - 1:1 게임 (상대편은 Player2, 나는 Player1)
 - Player 의 keyboard 방향 전환만 message를 보낸다.
 - `CONN ID oldx oldy newx newy`
 - 새로운 연결
 - `MOVE ID oldx oldy newx newy`
 - 현재 위치 통보
 - `MOVEUP ID oldx oldy newx newy`
 - `MOVEDOWN ID oldx oldy newx newy`
 - `MOVELEFT ID oldx oldy newx newy`
 - `MOVERIGHT ID oldx oldy newx newy`

game_client3 : client1의 1:1 모드

```
C:\Windows\system32\cmd.exe
게임 서버 실행중.
1번째 사용자 추가.
CONN ID 101 64 05 64 05
2번째 사용자 추가.
CONN ID 102 66 02 66 02
MOVE ID 101 64 05 64 05
MOVEDOWN ID 101 64 05 64 05
MOVEDOWN ID 102 66 02 66 02
MOVELEFT ID 102 66 13 66 13
```

▶ 게임 메시지

- CONN ID oldx oldy newx newy
 - 새로운 연결
- MOVE ID oldx oldy newx newy
 - 현재 위치 통보
- MOVEUP ID oldx oldy newx newy
- MOVEDOWN ID oldx oldy newx newy
- MOVELEFT ID oldx oldy newx newy
- MOVERIGHT ID oldx oldy newx newy

```
C:\Windows\system32\cmd.exe
101 서버에 접속되었습니다.

101

101
```

```
C:\Windows\system32\cmd.exe
101 서버에 접속되었습니다.

101

101
```

Java Game 예 - 슈팅 게임

▶ Yscramble - 1Player

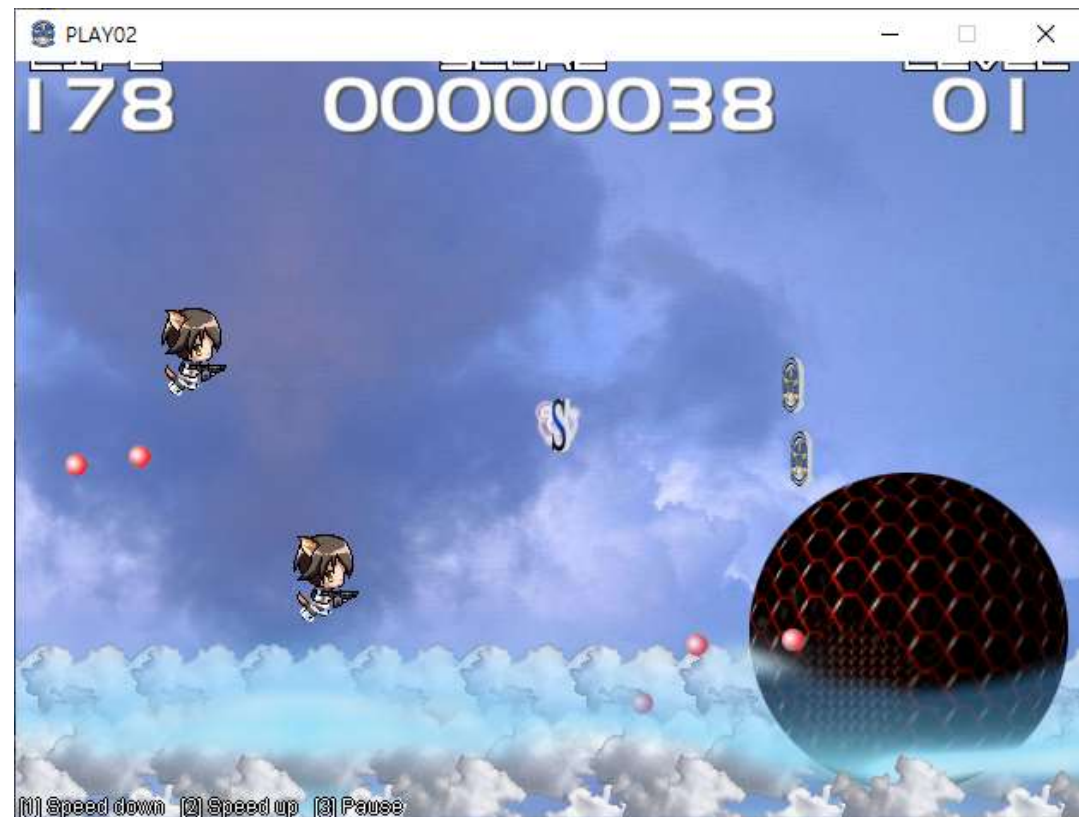


Java Game – 1인용을 2인용으로 확장

- ▶ Yscaramble-2player-nonet
- ▶ Player1 – 방향키 + Space
- ▶ Player2 – ASWD + F



Java Game – 2player – network version



Game Object 화면 처리, 속도, 이동, 이벤트 제어

▶ 화면 처리 (Double Buffering)

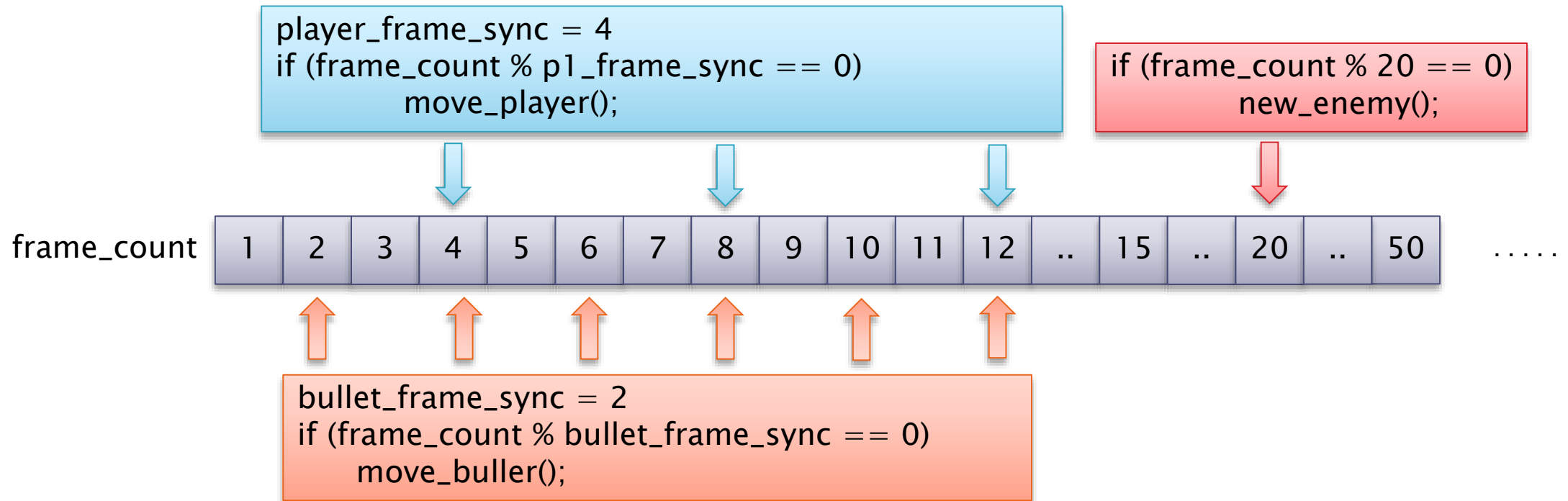
- 2개의 화면이 있다는 가정
 - 1. 보여지는 화면 = main
 - 2. 화면 크기와 같은 Image = dblbuffer
- While (true) {
 - // 배경, object, 총알, 효과 등등을 dblbuffer 에 그린다.
 - // main.drawImage(dblbuffer) 로 한번에 main을 update 한다.
 - // 각종 process 처리 (모든 object들의 이동, 활동, 이벤트 처리)
 - // Keyboard 처리
 - frame_count++;
 - }

▶ Object들의 속도/Event 제어

- 각각의 object 마다 frame_sync 값을 갖는다.
- frame_count % frame_sync == 0 이면 object의 이동/활성화 처리
- 예) 총알이 player보다 2배 빠르게 움직이게 한다면
 - if (frame_count % 4 == 0) move_player() // 기본 움직임
 - if (frame_count % 2 == 0) move_bullet() // 총알 속도는 2배
 - if (frame_count % 20 == 0) new_enemy() // 10 frame마다 적 등장

Game Object 별 속도제어 원리 예

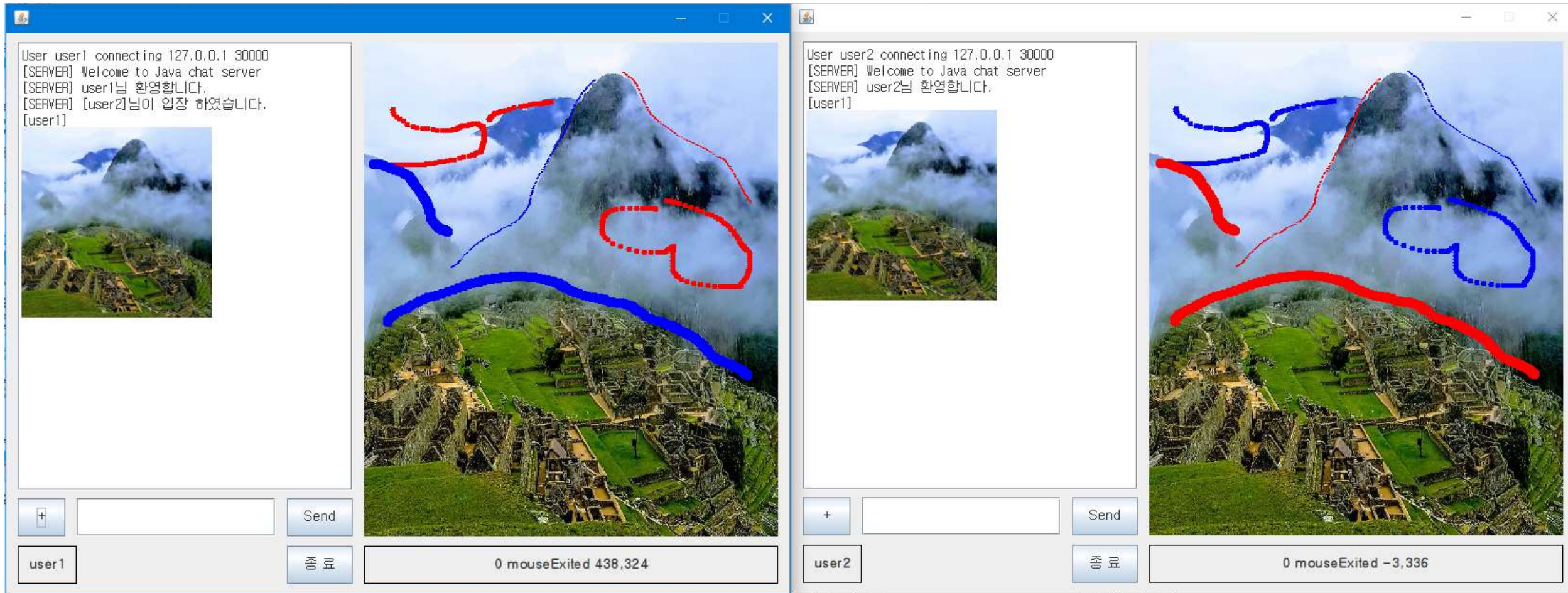
- ▶ Player의 속도 느리게/빠르게
 - `player_frame_sync++` / `player_frame_sync--`
- ▶ bullet 속도 느리게/빠르게
 - `bullet_frame_sync++` / `bullet_frame_sync--`
- ▶ 20 frame에 새로운 적군 등장



JavaGameClient/Server – Drawing 공유

- ▶ JavaObjClient/Server ObjectOutputStream 이용
- ▶ Mouse Event 처리, 공유
 - Panel 에 mouse drag 하면 dot 표시(내 것은 Blue, 다른 user는 Red)
 - Mouse Event 처리 함수
 - `class MyMouseEvent implements MouseListener, MouseMotionListener {`
 - `public void mouseDragged(MouseEvent e) {`
 - `SendMouseEvent(e);`
 - `}`
- ▶ Mouse Event 전송
 - ChatMsg object 사용, code = "500"
 - `public void SendMouseEvent(MouseEvent e) {`
 - `ChatMsg cm = new ChatMsg(UserName, "500", "MOUSE");`
 - `cm.mouse_e = e;`
 - `cm.pen_size = pen_size; // mouse wheel 로 pen 크기 조정`
 - `SendObject(cm);`
 - `}`
- ▶ Image 첨부하면 Drawing Area 에 Image 표시

JavaGameClient/Server



JavaGameClient/Server Image 사라지는 현상

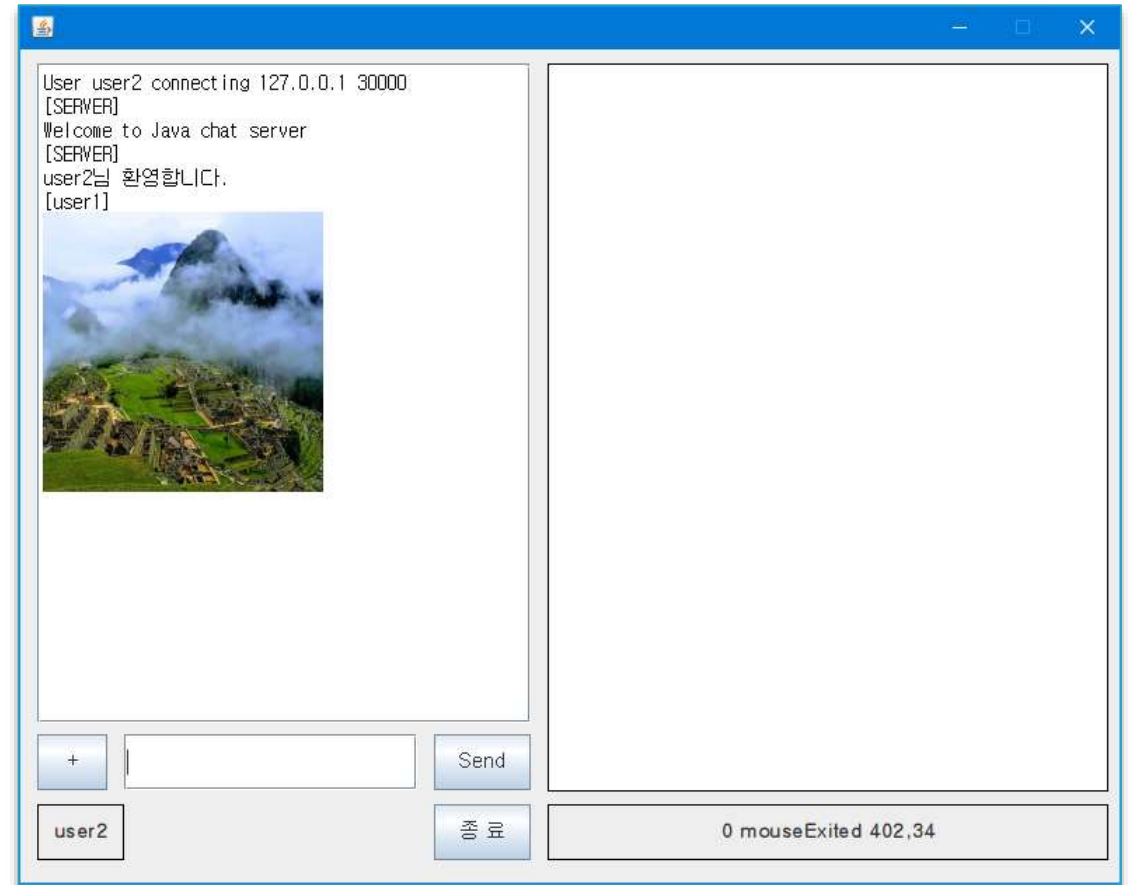
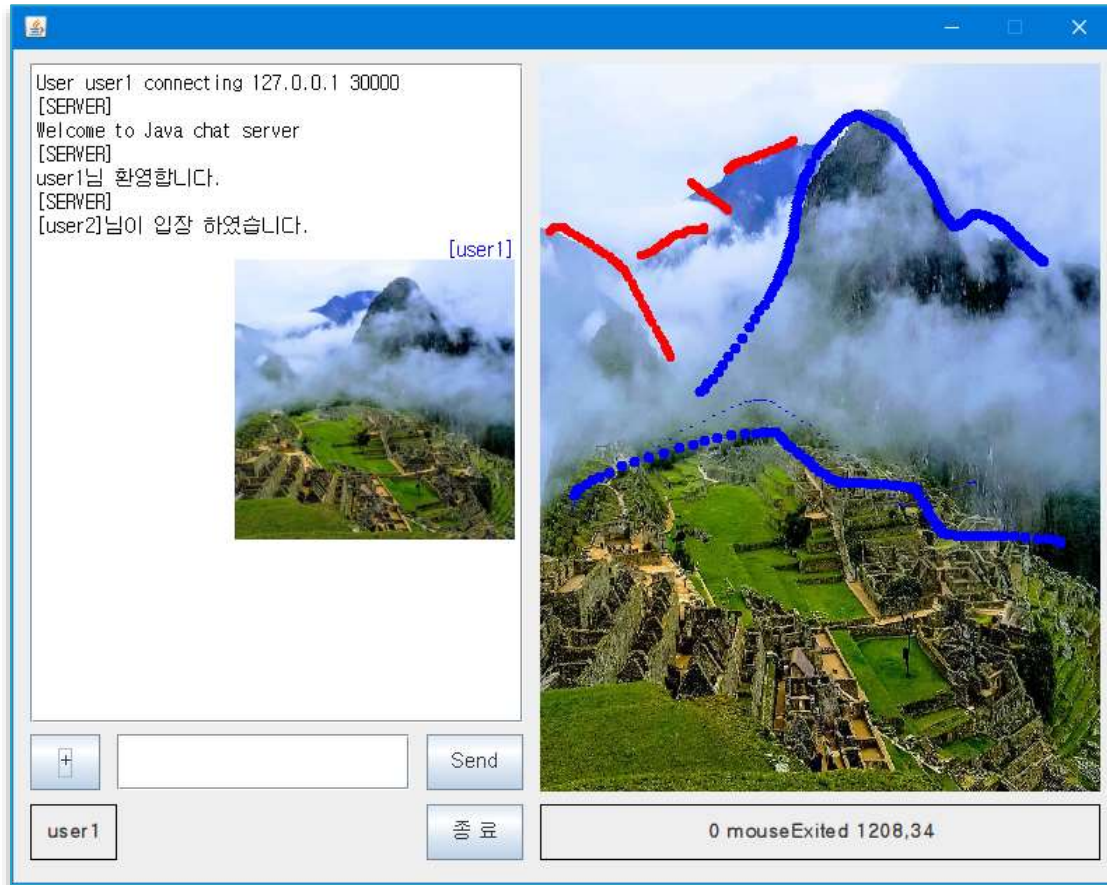
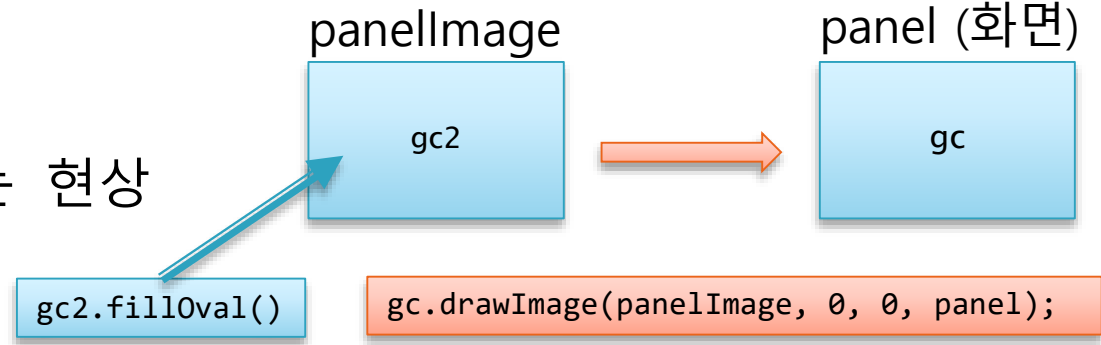


Image 보관 및 paint() 처리

- ▶ Windows가 가려졌다 나타나면 Image가 지워지는 현상
- ▶ paint() 이벤트 함수 처리 필요
- ▶ 일종의 Double Buffering 방법을 사용



```
JPanel panel;
private JLabel lblMouseEvent;
private Graphics gc;
private int pen_size = 2; // minimum 2
// 그려진 Image를 보관하는 용도, paint() 함수에서 이용한다.
private Image panelImage = null;
private Graphics gc2 = null;
```

```
// Image 영역 보관용. paint() 에서 이용한다.
panelImage = createImage(panel.getWidth(), panel.getHeight());
gc2 = panelImage.getGraphics();
gc2.setColor(panel.getBackground());
gc2.fillRect(0,0, panel.getWidth(), panel.getHeight());
gc2.setColor(Color.BLACK);
gc2.drawRect(0,0, panel.getWidth()-1, panel.getHeight()-1);
```

```
public void paint(Graphics g) {
    super.paint(g);
    // Image 영역이 가려졌다 다시 나타날 때 그려준다.
    gc.drawImage(panelImage, 0, 0, this);
}
```

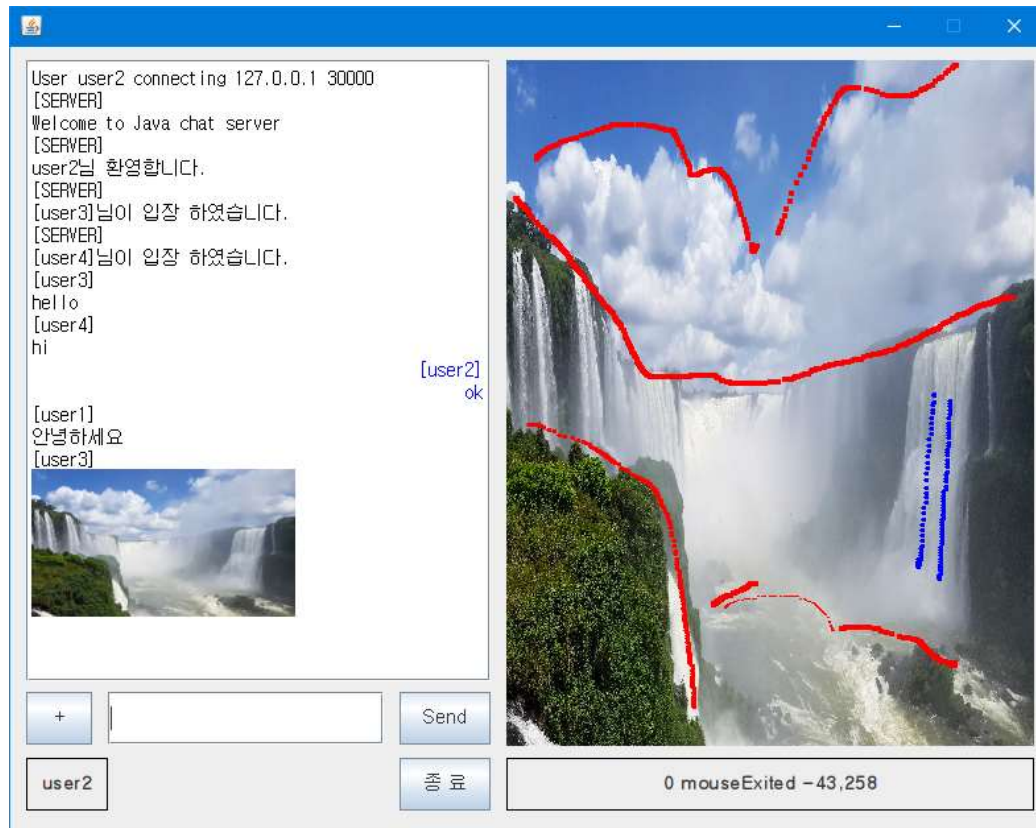
```
// Mouse Event 수신 처리
public void DoMouseEvent(ChatMsg cm) {
    Color c;
    if (cm.UserName.matches(Username)) // 본인 것은
        return;
    c = new Color(255, 0, 0); // 다른 사람 것은 Red
    gc2.setColor(c);
    gc2.fillOval(cm.mouse_e.getX() - pen_size/2,
    gc.drawImage(panelImage, 0, 0, panel);
}
```

```
public void mouseDragged(MouseEvent e) {
    lblMouseEvent.setText(e.getButton() + " mouseDragged " + e.getX() + "," + e.getY()); // 좌표출력가능
    Color c = new Color(0,0,255);
    gc2.setColor(c);
    gc2.fillOval(e.getX()-pen_size/2, e.getY()-pen_size/2, pen_size, pen_size);
    // panelImage는 paint()에서 이용한다.
    gc.drawImage(panelImage, 0, 0, panel);
    SendMouseEvent(e);
}
```

```
@Override
public void mouseClicked(MouseEvent e) {
    lblMouseEvent.setText(e.getButton() + " mouseClicked " + e.getX() + "," + e.getY());
    Color c = new Color(0,0,255);
    gc2.setColor(c);
    gc2.fillOval(e.getX()-pen_size/2, e.getY()-pen_size/2, pen_size, pen_size);
    gc.drawImage(panelImage, 0, 0, panel);
    SendMouseEvent(e);
}
```

JavaGameClient2 멀티윈도우 통신

▶ 단일윈도우



▶ 멀티윈도우



JavaGameClient2 멀티윈도우 통신 구조

- ▶ 모든 통신은 socket을 최초 생성한 View class를 통해서 한다

JavaGameClientView

```
switch (cm.code) {  
    case "200": // chat message  
        if (cm.UserName.equals(Username))  
            AppendTextR(msg); // 내 메시지는 우측에  
        else  
            AppendText(msg);  
        break;  
    case "300": // Image 첨부  
        if (cm.UserName.equals(Username))  
            AppendTextR("[ " + cm.UserName + " ]");  
        else  
            AppendText("[ " + cm.UserName + " ]");  
        AppendImage(cm.img);  
        break;  
    case "500": // Mouse Event 수신  
        drawing.DoMouseEvent(cm);  
        break;  
}
```

```
public void AppendImage(ImageIcon ori_icon) {  
    drawing.AppendImage(ori_icon);  
}
```



JavaGameClientViewDrawing

```
public class JavaGameClientViewDrawing extends JFrame {  
    private static final long serialVersionUID = 1L;  
    private JPanel contentPane;  
    private String Username;  
    JPanel panel;  
    private JLabel lblMouseEvent;  
    private Graphics gc;  
    private int pen_size = 2; // minimum 2  
    // 그려진 Image를 보관하는 용도, paint() 함수에서 이용한다.  
    private Image panelImage = null;  
    private Graphics gc2 = null;  
    public JavaGameClientView mainview;  
  
    public JavaGameClientViewDrawing(String username, JavaGameClientView view) {  
        mainview = view;  
    }  
  
    public void SendMouseEvent(MouseEvent e) {  
        ChatMsg cm = new ChatMsg(Username, "500", "MOUSE");  
        cm.mouse_e = e;  
        cm.pen_size = pen_size;  
        mainview.SendObject(cm);  
    }  
}
```

```
JavaGameClientView view;  
view = this;  
JavaGameClientViewDrawing drawing  
drawing = new JavaGameClientViewDrawing(username, view)
```

10주 과제

- ▶ 과제1 : game_client1, 2, 3 실습 (화면 제출)
- ▶ 과제2 : JavaGameClient/Server (Drawing 공유) 실습
 - 배경 Image 변경해서 실습하고 화면 제출
- ▶ 도전과제 : JavaGameClient/Server Drawing 기능 추가
 - Dot 대신 Line 그리기
 - 또는 Drawing 메뉴 선택(Dot, Line, Rectangle, Circle 등등)
 - Color 선택하기
 - 지우개 기능
 - 기타 Idea
 - Drawing 창을 별도로 띄우기
 - 스크롤바 사용, 화면 줌인, 줌아웃
 - 이미지 저장하기
- ▶ 과제3 : 텀프로젝트 설계서 제출