

1. 수정 내용 설명

1.1. 전체적인 디자인 변경

- 아이콘: 프로그램 아이콘을 주황색 말풍선 이미지로 변경하였습니다.
- 텍스트 폰트: verdana 로 통일하고, 각자 오브젝트의 크기에 맞게 글자 크기도 수정하였습니다.
- 버튼 변경: 배경 색상을 변경하였습니다.

1.2. 서버 접속/해제 시 알림창 출력

1.3. 이모지 기능 추가

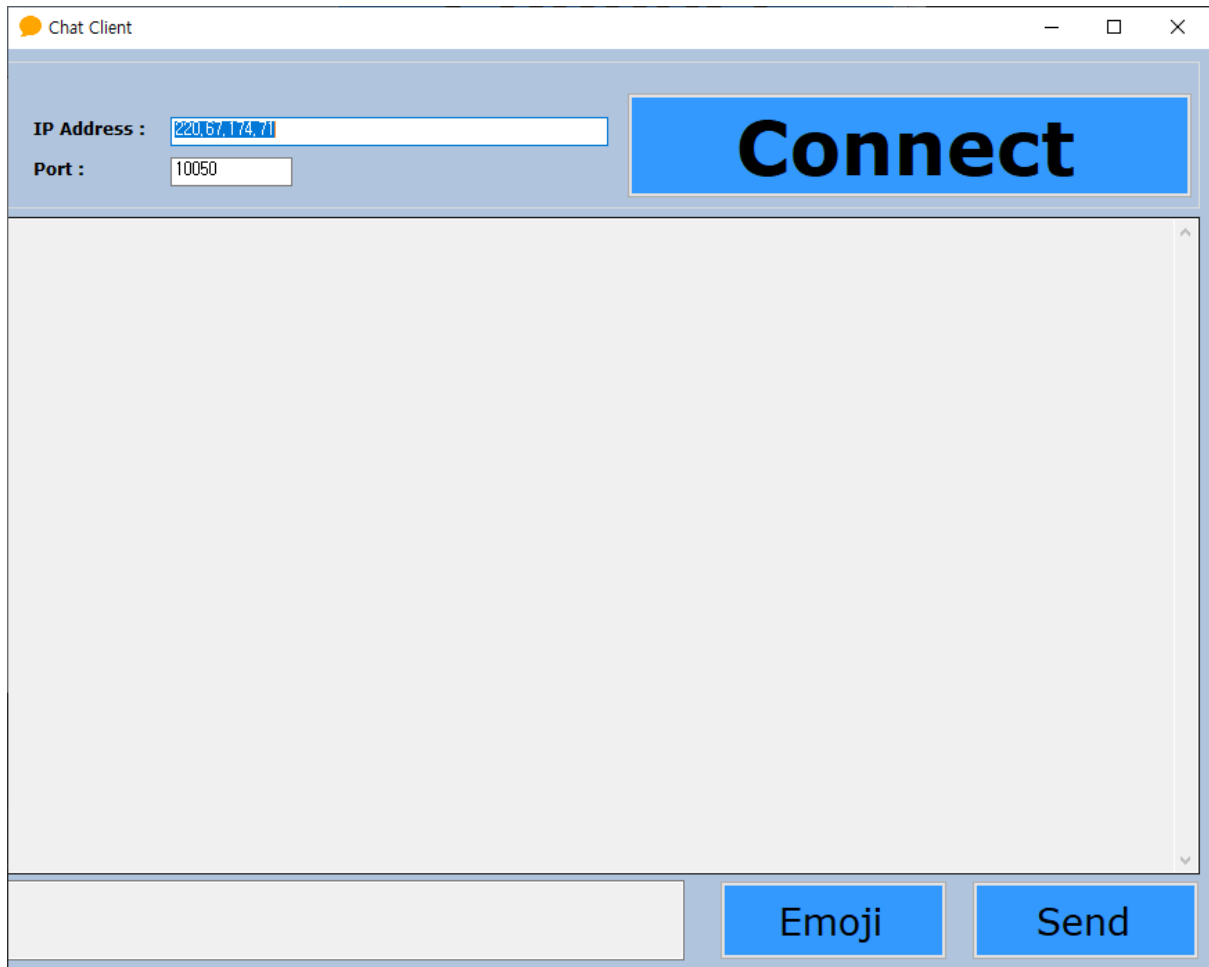
- 이제 이모티콘을 입력할 수 있습니다.(뒤에 사용설명에 자세히 설명)

1.4. 서버 접속 전 텍스트 전송 창을 비활성화하고, 대화 출력 창은 아예 입력할 수 없게 변경

1.5. 텍스트 인코딩: 이모티콘 전송 및 출력 시 정상 작동을 위해 UTF8 로 인코딩하였습니다.

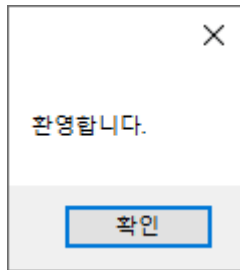
2. 동작 화면 및 사용설명서

2.1. 프로그램 시작

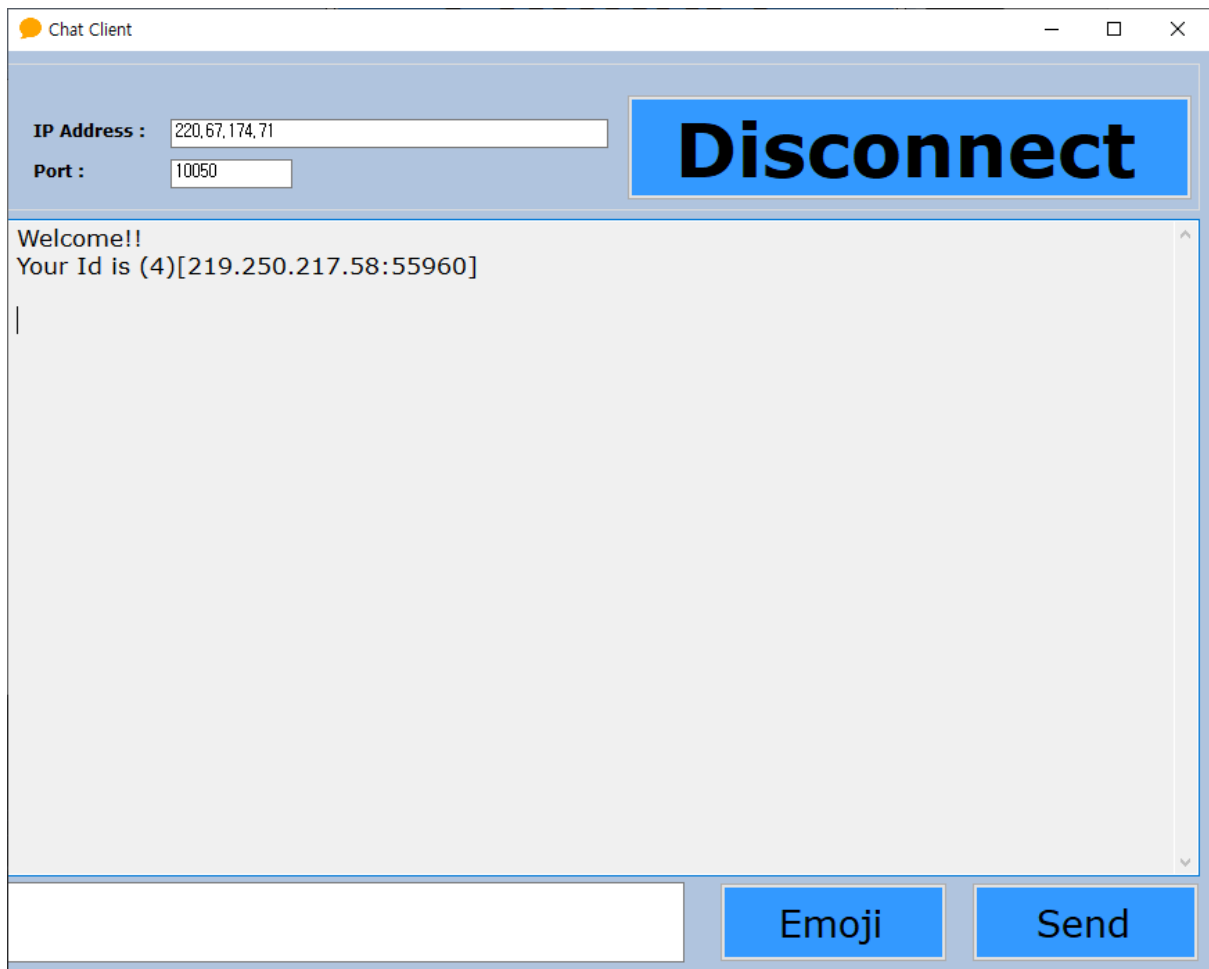


시작 시 보여지는 화면입니다. ip 주소와 포트 번호를 입력하여 접속할 수 있습니다. emoji 버튼을 눌러 emoji 를 넣을 수 있는 창을 출력할 수 있습니다. 하지만 접속 전에는 모든 텍스트가 입력되지 않습니다.

2.2. 접속 성공 시



해당 알림창이 팝업되고 확인을 누르면 접속됩니다.



접속 후 나의 id 가 출력됩니다. 이제 텍스트 전송창(txtSend) 텍스트와 이모티콘 입력이 가능합니다.

2.3. 이모티콘

Emoji 버튼을 눌러 Smilies 탭이 선택되어 있는 EmojiMain form 을 띄울 수 있습니다.

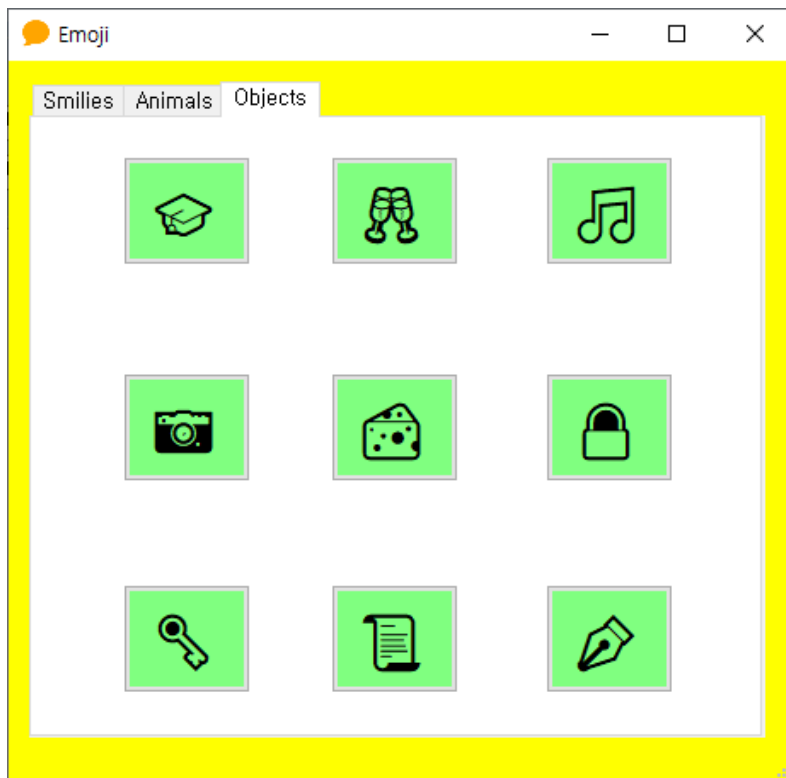


탭을 전환하면 해당 탭의 이모티콘 버튼으로 변경됩니다.

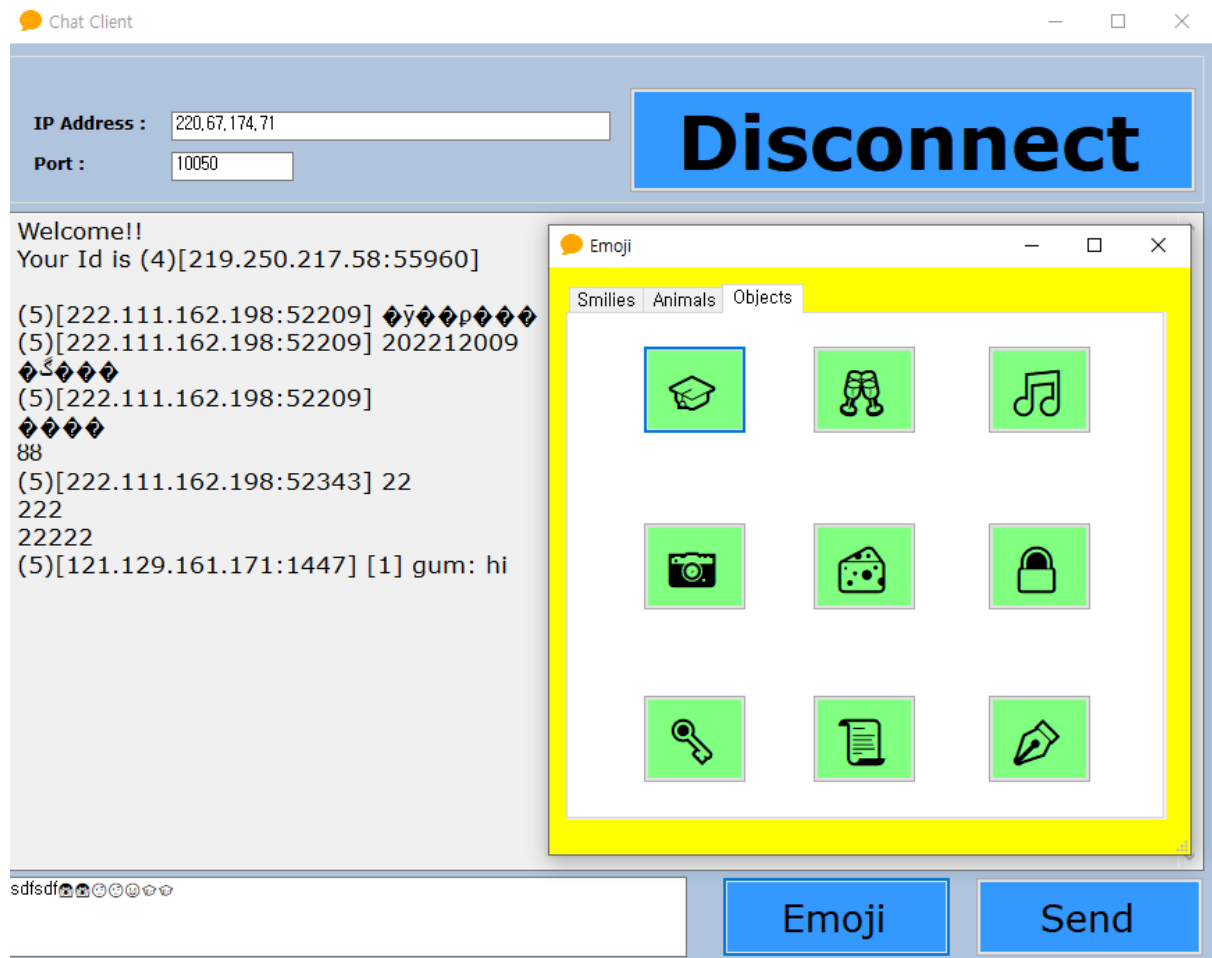
Animals 탭 클릭:



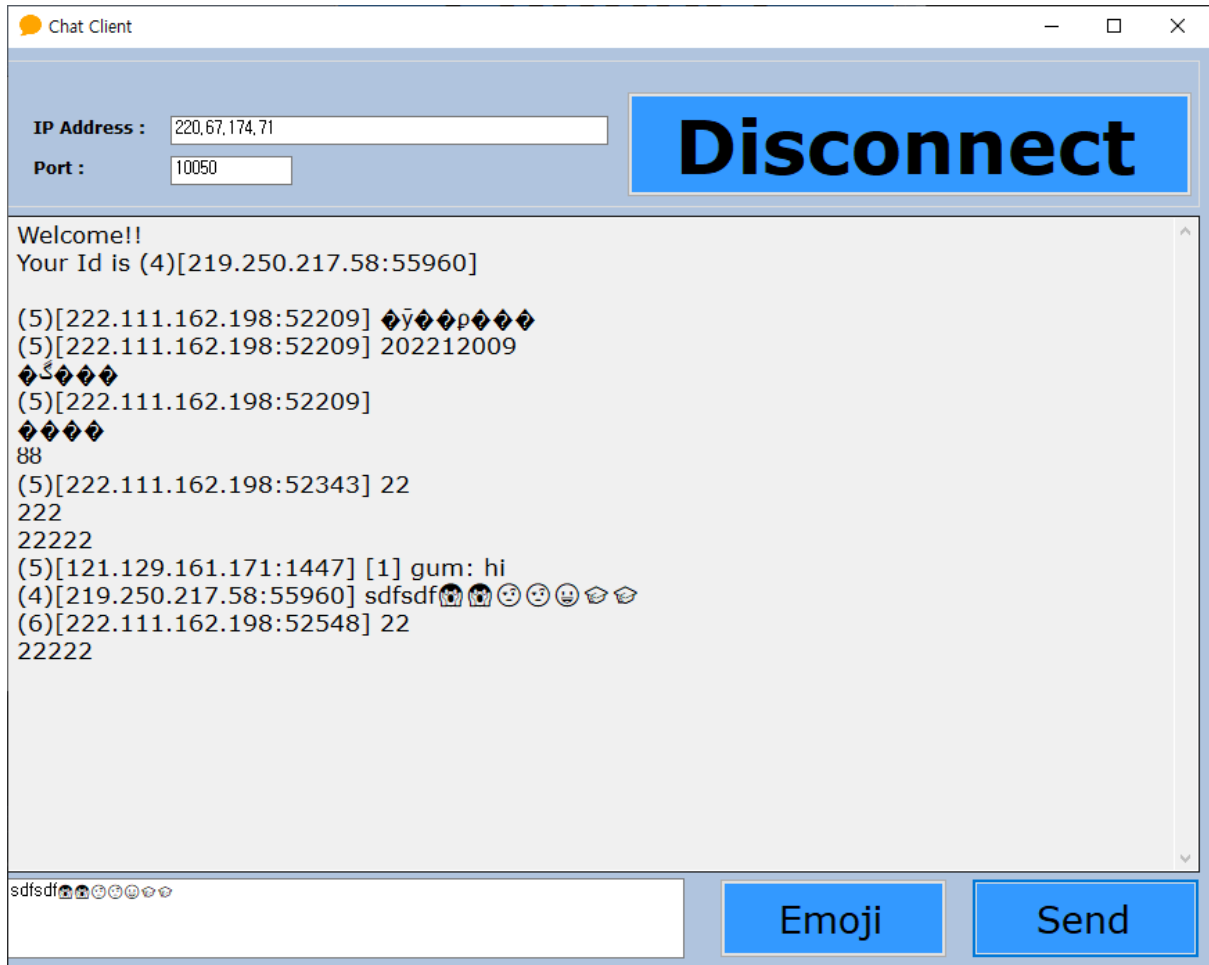
Objects 탭:



해당 이모티콘 버튼을 누르면 입력 텍스트 창에 바로 출력됩니다. 해당 form 을 닫기 전까지 탭에 상관없이 버튼을 누를 때마다 해당 이모티콘이 추가됩니다.

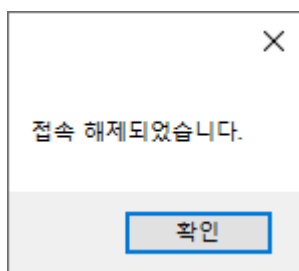


emoji 창을 닫고 send 를 누르면 입력창의 텍스트가 서버로 전송되어 중단의 텍스트 박스에 출력됩니다.



2.4. 접속 종료

Disconnect 버튼을 누르면 접속이 해제되고 알림 창이 팝업됩니다. 텍스트와 이모티콘을 입력할 수 없습니다.



3. 수정 코드

3.1. CFormMain.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChatClient
{
    public partial class CFormMain : Form
    {
        public TcpClient mTcpClient = null;
        public NetworkStream mStream = null;
        public static EmojiMain emojiForm = new EmojiMain(); //이모티콘을 선택하여 입력할 수
        있는 form

        public CFormMain()
        {
            InitializeComponent();
        }

        private void btConnect_Click(object sender, EventArgs e)
        {
            if (mTcpClient == null)
            {
                mTcpClient = new TcpClient(AddressFamily.InterNetwork);
                IPAddress ipAddress = IPAddress.Parse(txtIpAddress.Text);
                int port = Convert.ToInt32(txtPort.Text);

                MessageBox.Show("환영합니다."); //접속 성공 후 알림 창 팝업

                IPEndPoint iPEndPoint = new IPEndPoint(ipAddress, port);
                mTcpClient.Connect(iPEndPoint);

                btConnect.Text = "Disconnect";

                timerReceive.Enabled = true;
                mStream = mTcpClient.GetStream();
            } else
            {
                mStream.Close();
                mStream.Dispose();
                mStream = null;
            }
        }
    }
}
```



```

        mTcpClient.Close();
        mTcpClient.Dispose();
        mTcpClient = null;

        MessageBox.Show("접속 해제되었습니다."); //접속 해제 후 알림 창 팝업

        btConnect.Text = "Connect";
        timerReceive.Enabled = false;
    }
}

private void checkEmoji()
{
    if (emojiForm.buttonText != null) //emojiForm 내의 buttonText 값이 바뀌었을 때
    {
        txtSend.Text += emojiForm.buttonText; //txtSend에 내용 추가
        emojiForm.buttonText = null; //buttonText 초기화하여 다시 버튼을 누르면 그 값을
        받아올 준비를 한다.
    }
}

private void btSend_Click(object sender, EventArgs e)
{
    if (mTcpClient == null || mTcpClient.Connected == false)
    {
        return;
    }

    byte[] data = Encoding.UTF8.GetBytes(txtSend.Text); //입력한 문자열을 UTF8로
    인코딩하여 byte 배열에 저장
    mTcpClient.Client.Send(data);
}

private void timerReceive_Tick(object sender, EventArgs e)
{
    if (mTcpClient == null || mTcpClient.Connected == false || mStream == null)
    {
        return;
    }
    txtSend.ReadOnly = false; //접속 성공 후에 txtSend내에 내용을 입력할 수 있게 설정
    if (emojiForm.buttonText != null) //emojiForm 내의 buttonText 값이 바뀌었을 때
    {
        txtSend.Text += emojiForm.buttonText; //txtSend에 내용 추가
        emojiForm.buttonText = null; //buttonText 초기화하여 다시 버튼을 누르면 그 값을
        받아올 준비를 한다.
    }

    if (mStream.DataAvailable == true)
    {
        byte[] bzReceive = new byte[1024];

        mStream.ReadAsync(bzReceive, 0, 1024);
    }
}

```

```
문자열로 변환      string strReceive = Encoding.UTF8.GetString(bzReceive); //해당 byte 배열을 UTF8

StringBuilder sb = new StringBuilder(strReceive);

txtReceive.Text += sb;
txtReceive.AppendText("WrWn");

    }
}

private void btnEmoji_Click(object sender, EventArgs e)
{
    emojiForm.ShowDialog(); //emojiForm 보여주기
}
}
```

3.2. EmojiMain.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChatClient
{
    public partial class EmojiMain : Form
    {
        public string buttonText = null; //button의 text를 담을 변수, public으로 선언하여 다른
        클래스에서 접근할 수 있도록 한다.
        public EmojiMain()
        {
            InitializeComponent();

            void button_Click(object sender, EventArgs e) //모든 emoji 버튼의 이벤트를 처리하는
            메서드
            {
                Button btn = sender as Button; //이벤트를 발생시킨 객체를 버튼으로 정의
                byte[] bt = Encoding.UTF8.GetBytes(btn.Text); //해당 버튼의 문자열을 UTF8
                encoding(하지 않을 경우 깨짐)
                buttonText = Encoding.UTF8.GetString(bt); //encoding된 byte 배열을 다시 String
                type으로 변환
            }
        }
    }
}
```