

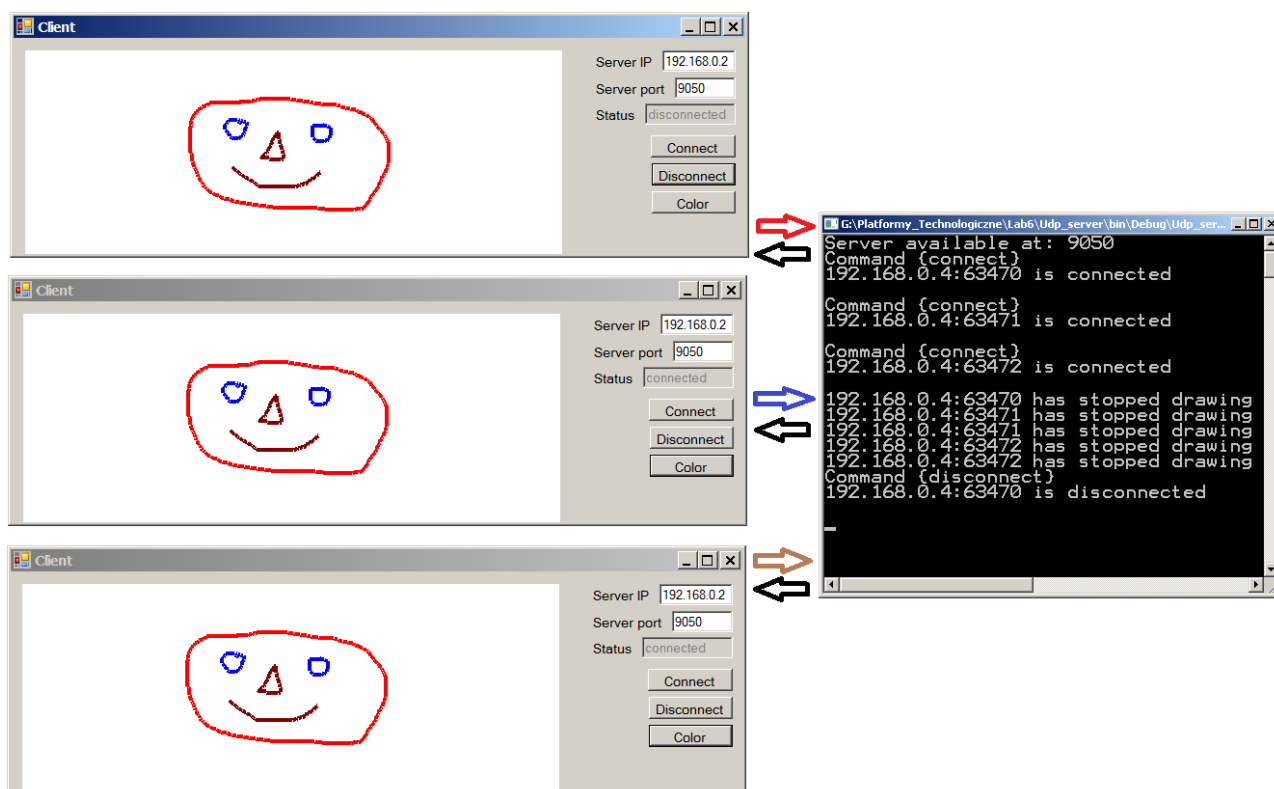
## C# – Lab 6

### Dokumentacja

- [UdpClient](#)
- [ConcurrentQueue](#)
- [BlockingCollection](#)
- [BitConverter](#)
- [Buffer.BlockCopy](#)

### Zadanie

Napisać aplikację klient-serwer (wykorzystującą komunikację poprzez protokół UDP) umożliwiającą jednoczesne rysowanie przez wielu klientów na „współdzielonym ekranie” (PictureBox). Serwer służy do przesyłania tego co rysuje jeden klient do wszystkich pozostałych. Aby klient mógł rysować na „współdzielonym ekranie” najpierw musi się do serwera podłączyć. Jeśli klient jest podłączony, przeciągnięcie po własnym PictureBox myszy z naciśniętym przyciskiem powoduje „zbieranie punktów” (wykorzystać zdarzenia MouseDown, MouseUp, MouseMove oraz własność MouseEventArgs.Location). Każdy zebrany punkt przesyłany jest do serwera. Serwer przekazuje odebrany punkt (wraz z informacją od którego klienta pochodzi) do wszystkich podłączonych klientów. Każdy klient odbiera od serwera punkty oraz rysuje linie pomiędzy kolejnymi punktami pochodzącymi od tego samego klienta. Klient może się od serwera odłączyć (wówczas przestaje wysyłać i odbierać punkty) oraz podłączyć ponownie.



### Wskazówka jak rysować w C#

Przeciągnij na formularz kontrolkę PictureBox oraz dodaj następujący kod:

```
Pen pen = new Pen(System.Drawing.Color.Red, 5);  
pictureBox1.Image = new Bitmap(pictureBox1.Width, pictureBox1.Height);  
Graphics graphics = Graphics.FromImage(pictureBox1.Image);  
graphics.DrawLine(pen, 10, 10, 100, 100);
```

## Punktacja

1. W dowolny sposób zaimplementować wyżej opisaną funkcjonalność (2.5 pkt)
2. Serwer posiada 3 wątki:
  - wątek główny nasłuchuje żądania podłączenia i odłączenia klienta (klient żąda podłączenia wysyłając komunikat "connect" a rozłączenia wysyłając "disconnect"); na komendę "connect" serwer odpowiada numerem portu, pod którym będzie odbierał „dane rysownicze” (0.5 pkt)
  - wątek drugi służy do odbierania „danych rysowniczych”; doklejania do nich id klienta, który je wysłał (id klienta ma mieć 1 bajt i ma być przydzielane przez serwer; zakładamy że nie będzie więcej niż 256 klientów); a następnie wrzucania do kolejki `BlockingConcurrentQueue` (1 pkt)
  - wątek trzeci służy do wyciągania „danych rysowniczych” z kolejki i przesyłania ich do klientów (wskazówka: użyć 3-argumentowy `send`)
3. Umożliwić wybór koloru rysowania; kolor przysyłać tylko raz na początku rysowania nowej linii; w trakcie „zbierania punktów” klient przesyła do serwera tylko współrzędne (1 pkt)