

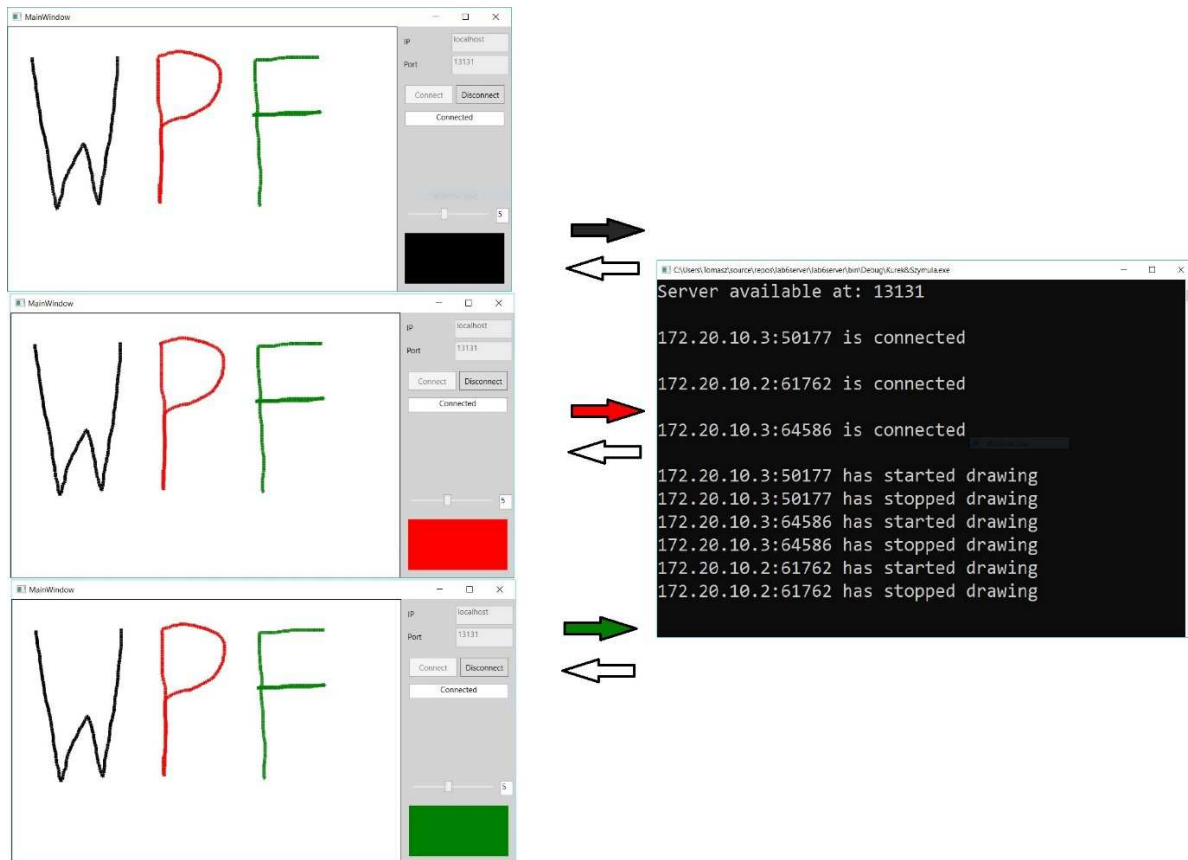
## C# – Lab 6

### Dokumentacja

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

### Zadanie

Napisać aplikację klient-serwer (wykorzystującą komunikację poprzez protokół UDP) umożliwiającą jednocześnie rysowanie przez wielu klientów na „współdzielonym ekranie” (Canvas). 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 Canvas myszy z naciśniętym przyciskiem powoduje „zbieranie punktów” (wykorzystać zdarzenia MouseDown/MouseUp, MouseMove, MouseEnter/MouseLeave). 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 WPF:

Przeciagnij na formularz kontrolkę Canvas oraz dodaj następujący kod:

```
Brush brush = Brushes.Red;
Line line = new Line() {
    X1 = 10,
    Y1 = 10,
    X2 = 100,
    Y2 = 100,
    Stroke = brush,
    StrokeThickness = 5
};
canvas.Children.Add(line);
```

WPF nie posiada wbudowanej metody obsługi okna do wyboru koloru, należy skorzystać z klasy `ColorDialog` udostępnionej w `System.Windows.Forms` (należy najpierw ją dodać do projektu: Add -> Reference... -> System.Windows.Forms)

```
ColorDialog colorDialog = new ColorDialog();
if(colorDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK) {
    Brush brush = new SolidColorBrush(Color.FromRgb(
        colorDialog.Color.R,
        colorDialog.Color.G,
        colorDialog.Color.B));
}
```

Do pobrania rysowanych punktów można użyć Eventu `MouseMove`

```
Private void canvas_MouseMove(object sender, MouseEventArgs e) {
    Point point = e.GetPosition(this);
}
```

## 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)