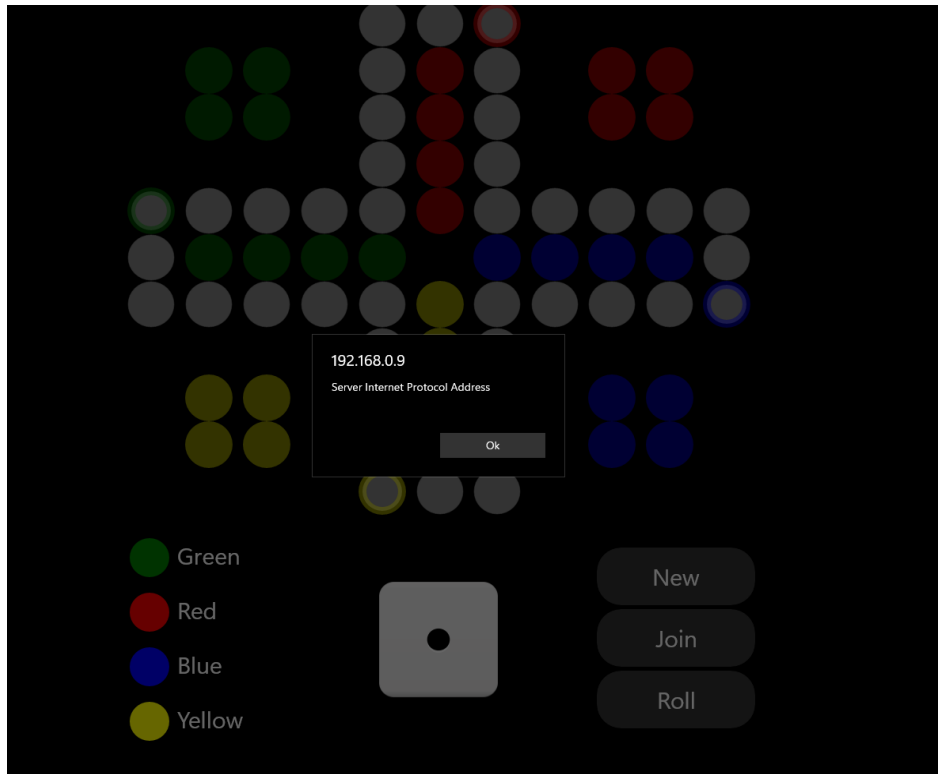


Don't get angry

# Button “New”

- ▶ First we start new game with button “New”. That create a server and other players can connect to it by IP address locally.



```
private async void New_Click(object sender, RoutedEventArgs e)
{
    // WhoAmI
    WhoAmI = "green";

    // New Game
    board = new Board();
    board.Players = 1;
    board.Roll = 0;
    board.Turn = WhoAmI;
    board.Green = new List<int>() { 0, 0, 0, 0 };
    board.Red = new List<int>() { 0, 0, 0, 0 };
    board.Blue = new List<int>() { 0, 0, 0, 0 };
    board.Yellow = new List<int>() { 0, 0, 0, 0 };

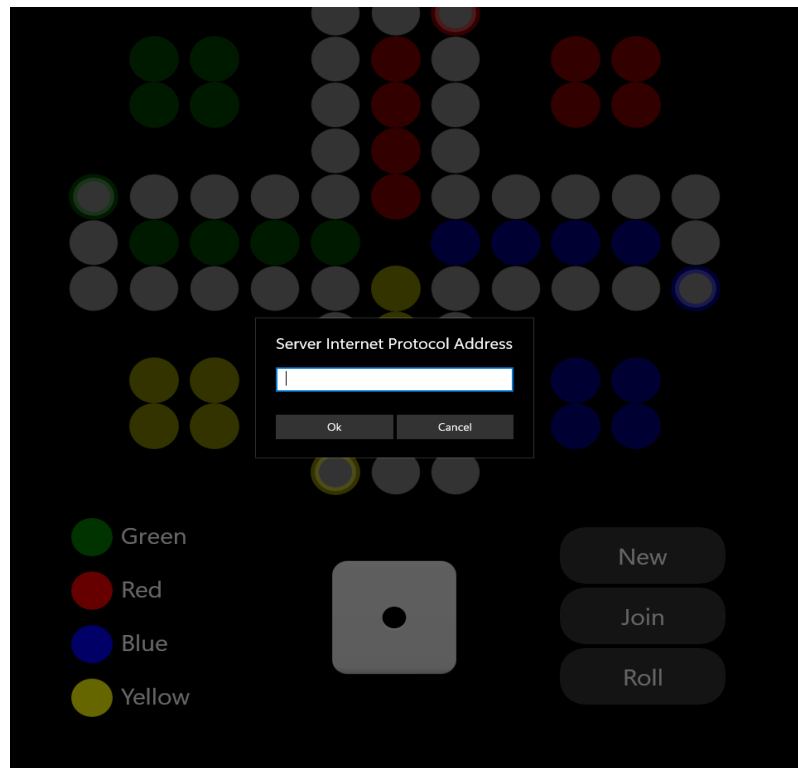
    // Wait All Clients
    server = new Server();
    server.OnMessageReceived += OnMessageReceived;
    server.OnClientConnected += OnClientConnected;
    ContentDialog dialog = new ContentDialog()
    {
        Title = GetIPAddress(),
        Content = "Server Internet Protocol Address",
        CloseButtonText = "Ok"
    };
    await Dialog.Show(dialog);

    // Server Send The Board as json
    await server.Send(JsonConvert.SerializeObject(board));
    UpdateUI();

    // Buttons
    this.Roll.Visibility = Visibility.Visible;
    this.New.Visibility = Visibility.Collapsed;
    this.Join.Visibility = Visibility.Collapsed;
}
}
```

# Button “Join”

- ▶ Players can connect to the server with “Join” button. They must enter IP of the server to be connected.

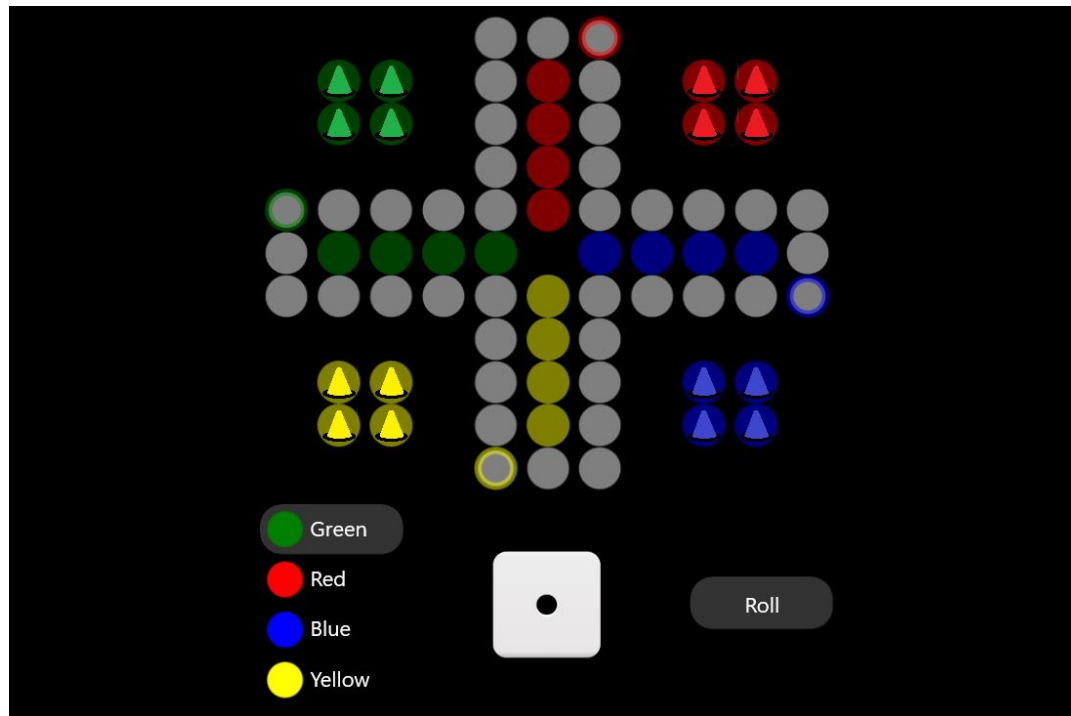


```
1 reference
private async void Join_Click(object sender, RoutedEventArgs e)
{
    string ip = await SetIpAddress("Server Internet Protocol Address");
    client = new Client(ip);
    client.OnMessageReceived += OnMessageReceived;

    // Buttons
    this.Roll.Visibility = Visibility.Collapsed;
    this.New.Visibility = Visibility.Collapsed;
    this.Join.Visibility = Visibility.Collapsed;
}
```

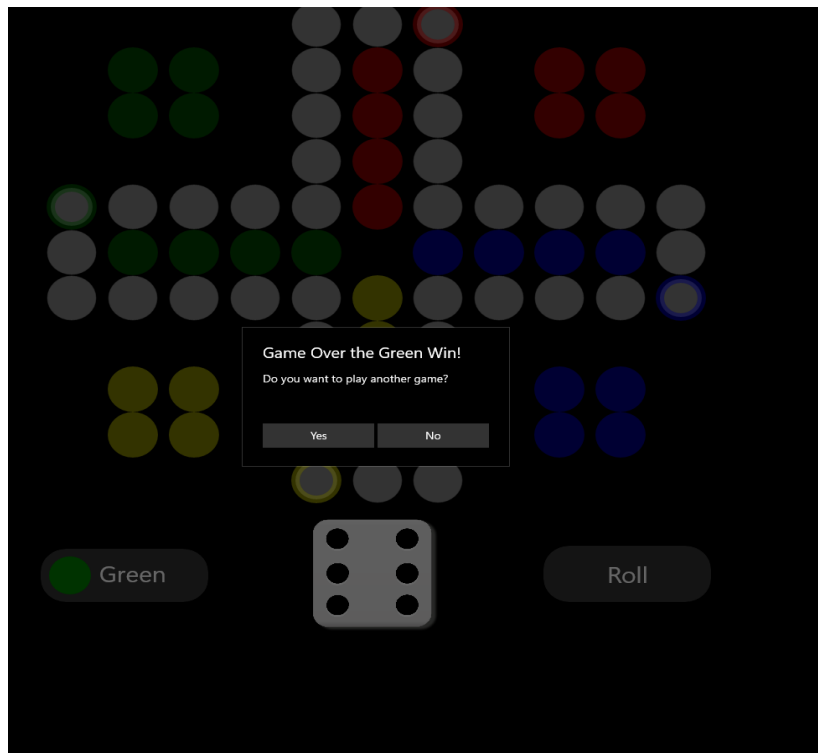
# During the game

- ▶ During the game you have button “Roll” for moving the figures. When you click “Roll” button that send message to the server and after this server send the board (in json format) to all players for updating the UI.



# End of the game

- ▶ In the end is showed a modal with description who wins the game and you can choose between starting a new game or close the game.



```
private async Task GameOver(string title)
{
    ContentDialog dialog = new ContentDialog();
    dialog.Title = title;
    dialog.Content = "Do you want to play another game?";
    dialog.PrimaryButtonText = "Yes";
    dialog.SecondaryButtonText = "No";
    dialog.IsSecondaryButtonEnabled = true;
    var result = await Dialog.Result(dialog);

    // If "Yes" then start a new game
    if (result == ContentDialogResult.Primary)
    {
        // New board
        board = new Board();
        board.Players = 1;
        board.Roll = 0;
        board.Turn = WhoAmI;
        board.Green = new List<int>() { 0, 0, 0, 0 };
        board.Red = new List<int>() { 0, 0, 0, 0 };
        board.Blue = new List<int>() { 0, 0, 0, 0 };
        board.Yellow = new List<int>() { 0, 0, 0, 0 };

        // Server initialize new board and send it to clients
        if (server != null)
        {
            await server.Send(JsonConvert.SerializeObject(board));
            UpdateUI();
            this.Roll.Visibility = Visibility.Visible;
            this.New.Visibility = Visibility.Collapsed;
            this.Join.Visibility = Visibility.Collapsed;
        }
        return;
    }

    if (result == ContentDialogResult.Secondary)
    {
        Application.Current.Exit();
    }
}
```

# Summary

- ▶ First we start new game with button “New” and create server.
- ▶ Players can connect to the server with button “Join”.
- ▶ During the game players can move their figures with button “Roll”.
- ▶ In the end of the game you can choose between start new game or close the game.

Thanks for your attention!