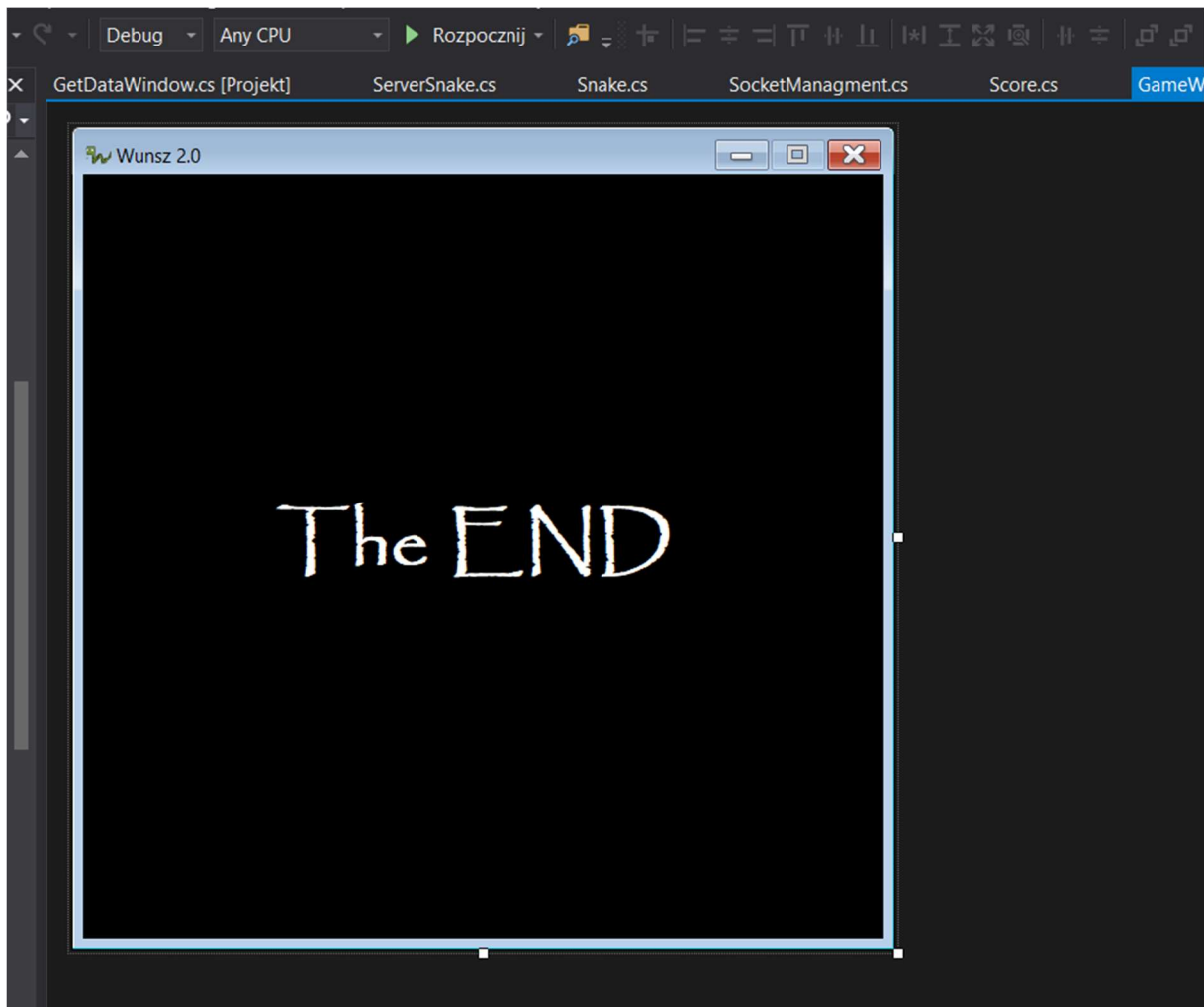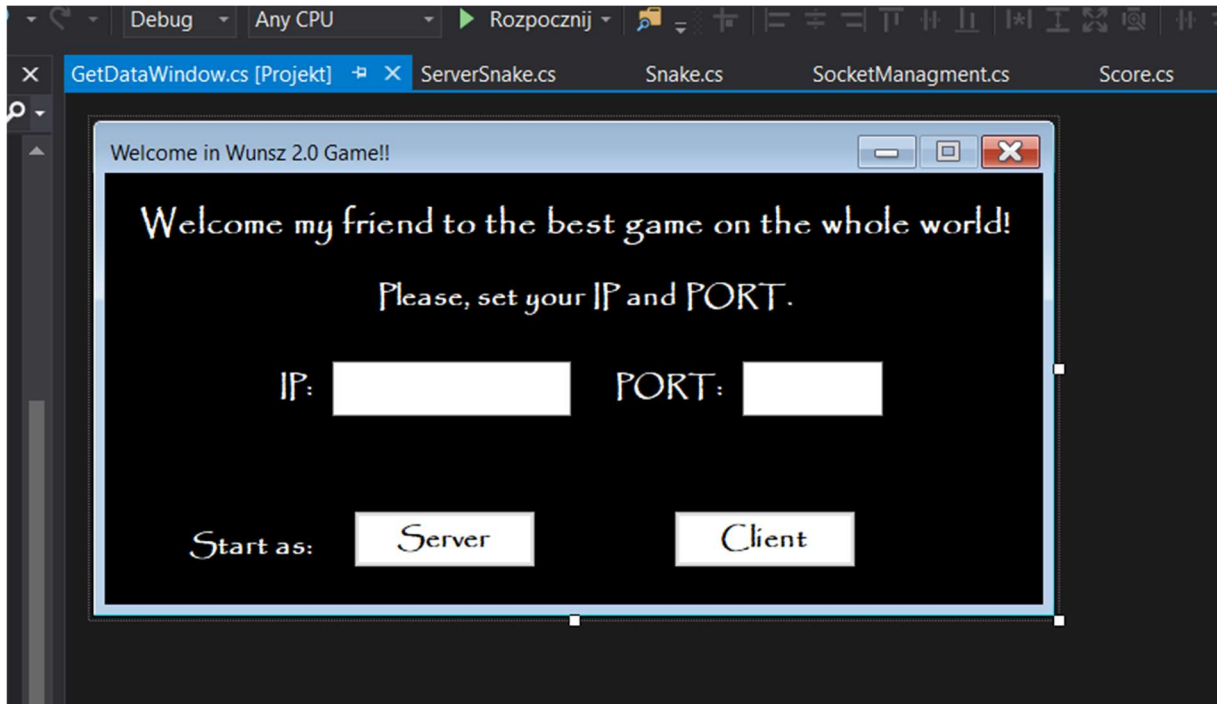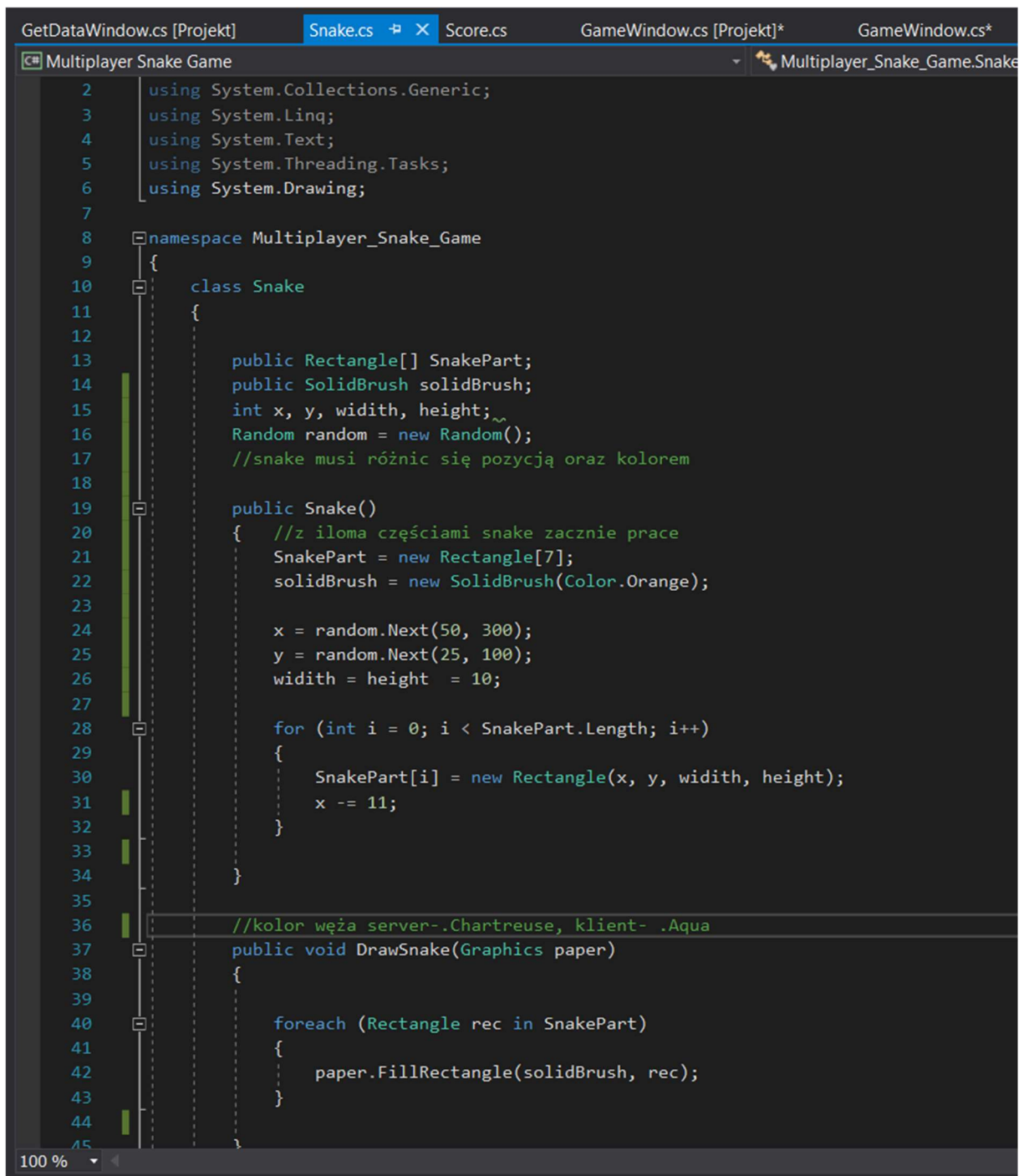Do dnia 21.04 zostały wykonane następujące zadania:

1. Został zaprojektowany wygląd okien

2. Został napisany kod, dla klas Snake oraz Score

```csharp
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;

namespace Multiplayer_Snake_Game
{
    class Snake
    {

        public Rectangle[] SnakePart;
        public SolidBrush solidBrush;
        int x, y, widith, height;
        Random random = new Random();
        //snake musi różnic się pozycją oraz kolorem


        public Snake()
        {   //z iloma częściami snake zacznie prace
            SnakePart = new Rectangle[7];
            solidBrush = new SolidBrush(Color.Orange);

            x = random.Next(50, 300);
            y = random.Next(25, 100);
            widith = height  = 10;

            for (int i = 0; i < SnakePart.Length; i++)
            {
                SnakePart[i] = new Rectangle(x, y, widith, height);
                x -= 11;
            }

        }

        //kolor węża server-.Chartreuse, klient- .Aqua
        public void DrawSnake(Graphics paper)
        {

            foreach (Rectangle rec in SnakePart)
            {
                paper.FillRectangle(solidBrush, rec);
            }
```

100 % ▾ ◂

```csharp
47          //części snake'a "podążają" za sobą, tj. poprzedni zajmuje miejsce następnego
48          public void DrawSnake()
49          {
50
51              for (int i = SnakePart.Length - 1; i > 0; i--)
52              {
53                  SnakePart[i] = SnakePart[i - 1];
54              }
55
56          }
57          //obsluga poruszania się
58          public void MoveUp()
59          {
60
61              DrawSnake();
62              SnakePart[0].Y -= 11;
63          }
64          public void MoveDown()
65          {
66
67              DrawSnake();
68              SnakePart[0].Y += 11;
69          }
70          public void MoveRight()
71          {
72              DrawSnake();
73              SnakePart[0].X += 11;
74          }
75          public void MoveLeft()
76          {
77              DrawSnake();
78              SnakePart[0].X -= 11;
79
80          }
81
82          public void GrowUp(int a)
83          {
84              Array.Resize(ref SnakePart, 4 + a);
85              SnakePart[4 + a - 1] = new Rectangle(((x - 16) * 4 + a - 1), y, widith, height);
86
87          }
88
89      }
90  }
```

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;

namespace Multiplayer_Snake_Game
{
    class Score
    {
        //public int WindowWidith, WindowHeight;
        public Rectangle ScoreRectangle;
        public SolidBrush so;
        public int x, y, width, height;

        //można by było poprawić coś z pojawianiem się punktów
        public Score(Random ScoreRandom)
        {
            widith = height = 15;
            so = new SolidBrush(Color.Yellow);

            x = ScoreRandom.Next(20, 500);
            y = ScoreRandom.Next(20, 500);

            ScoreRectangle = new Rectangle(x, y, width, height);
        }

        public void ScoreLocation(Random ScoreRandom)
        {
            x = ScoreRandom.Next(15, 500); //405 - 15
            y = ScoreRandom.Next(15, 500);
        }

        public void DrawScore(Graphics paper)
        {
            ScoreRectangle.X = x;
            ScoreRectangle.Y = y;

            paper.FillRectangle(so, ScoreRectangle);
        }
    }
}
```