

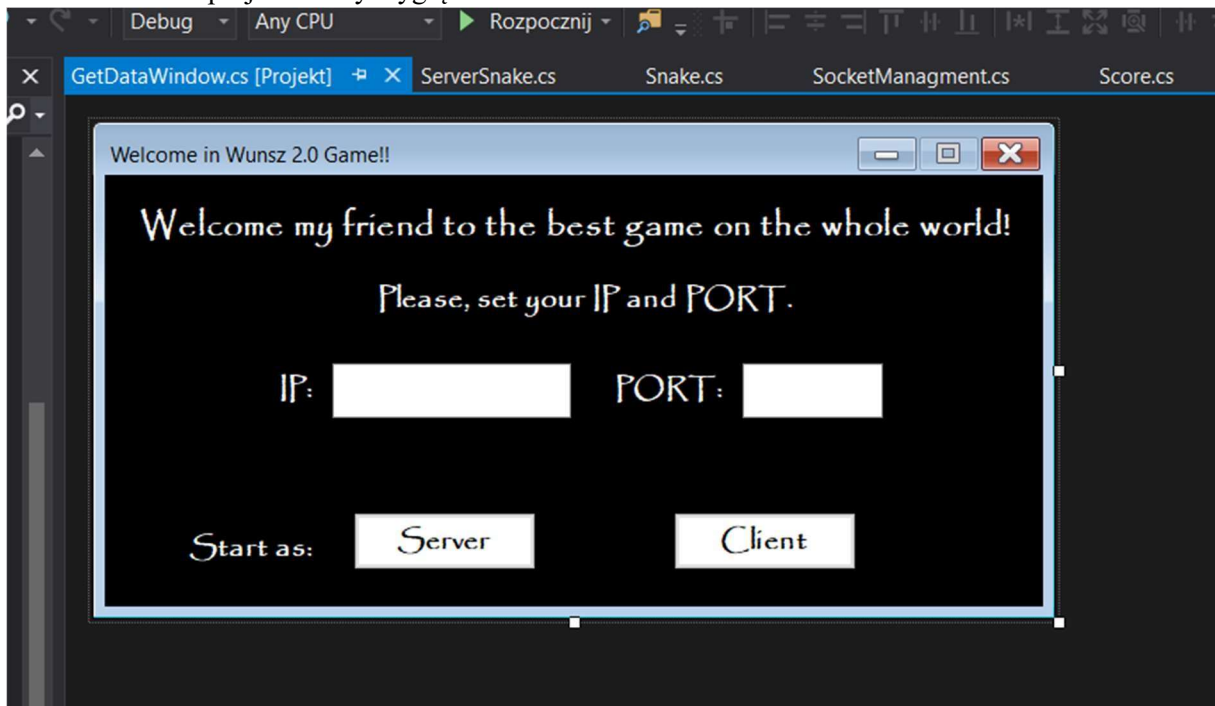
Wybór języka programowania do projektu był podyktowany dwoma założeniami:

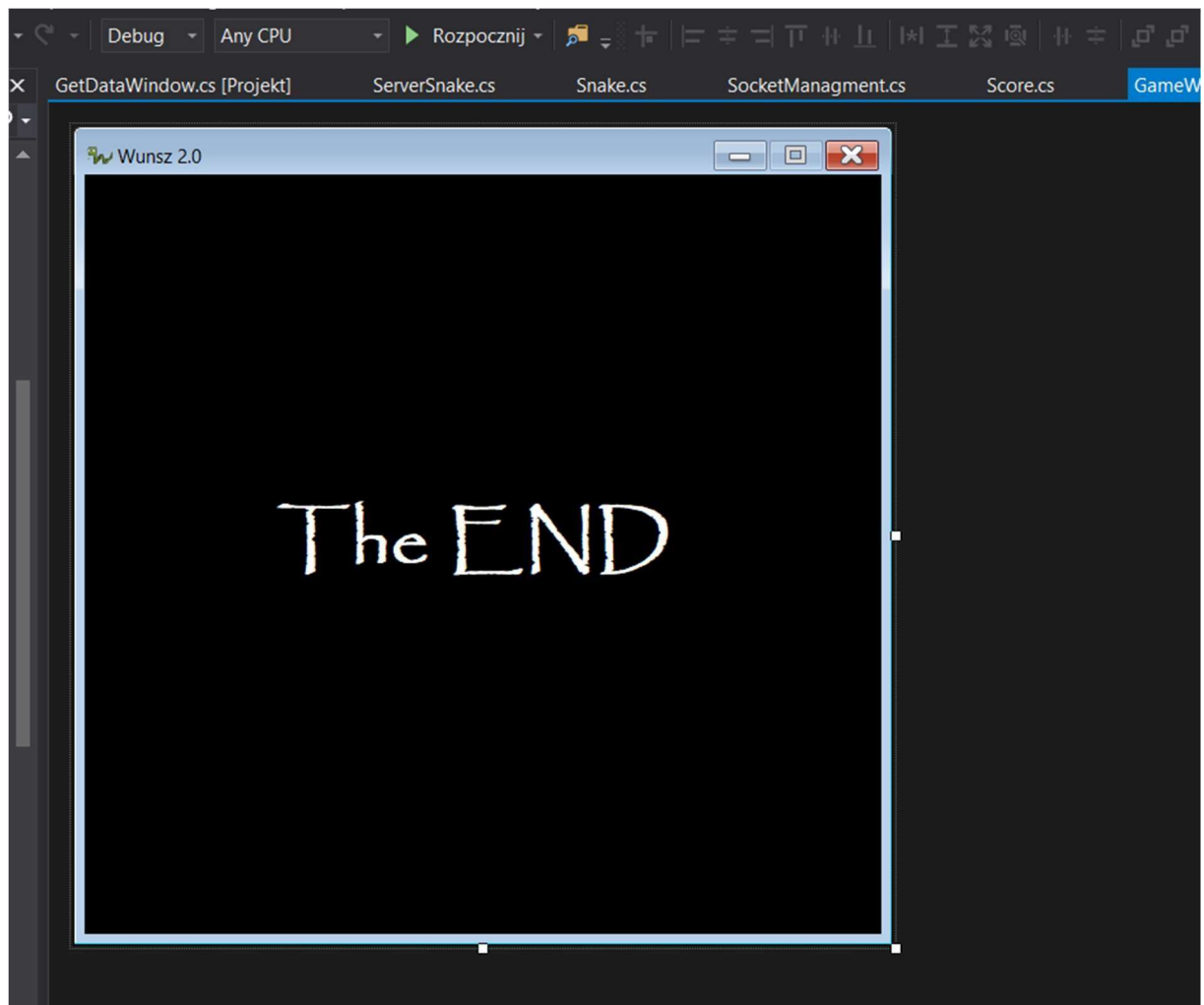
- Gra będzie aplikacją okienkową
- Gra będzie dedykowana na komputery z systemem Windows

Wybór padł więc na język C#. Jest on dedykowany do tworzenia aplikacji okienkowych w środowisku Windows.

Do dnia 21.04 zostały wykonane następujące zadania:

1. Został zaprojektowany wygląd okien





2. Został napisany kod, dla klas Snake oraz Score

```
GetDataWindow.cs [Projekt]  Snake.cs  Score.cs  GameWindow.cs [Projekt]*  GameWindow.cs*
Multiplayer Snake Game  Multiplayer_Snake_Game.Snake

2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Drawing;
7
8  namespace Multiplayer_Snake_Game
9  {
10     class Snake
11     {
12
13         public Rectangle[] SnakePart;
14         public SolidBrush solidBrush;
15         int x, y, width, height;
16         Random random = new Random();
17         //snake musi różnić się pozycją oraz kolorem
18
19         public Snake()
20         {
21             //z iloma częściami snake zacznie prace
22             SnakePart = new Rectangle[7];
23             solidBrush = new SolidBrush(Color.Orange);
24
25             x = random.Next(50, 300);
26             y = random.Next(25, 100);
27             width = height = 10;
28
29             for (int i = 0; i < SnakePart.Length; i++)
30             {
31                 SnakePart[i] = new Rectangle(x, y, width, height);
32                 x += 11;
33             }
34         }
35
36         //kolor węża server- .Chartreuse, klient- .Aqua
37         public void DrawSnake(Graphics paper)
38         {
39
40             foreach (Rectangle rec in SnakePart)
41             {
42                 paper.FillRectangle(solidBrush, rec);
43             }
44         }
45     }
46 }
```

```
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 //obsługa 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, width, height);
86 }
87
88
89
90 }
```

```
GetDataWindow.cs [Projekt]    Snake.cs    Score.cs* X    GameWindow.cs [Projekt]*    GameWindow.cs*
Multiplayer Snake Game    Multiplayer_Snake_Game.Score

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Drawing;
7
8  namespace Multiplayer_Snake_Game
9  {
10     class Score
11     {
12         //public int WindowWidth, WindowHeight;
13         public Rectangle ScoreRectangle;
14         public SolidBrush so;
15         public int x, y, width, height;
16
17         //można by było poprawić coś z pojawianiem się punktów
18         public Score(Random ScoreRandom)
19         {
20             width = height = 15;
21             so = new SolidBrush(Color.Yellow);
22
23             x = ScoreRandom.Next(20, 500);
24             y = ScoreRandom.Next(20, 500);
25
26             ScoreRectangle = new Rectangle(x, y, width, height);
27         }
28
29         public void ScoreLocation(Random ScoreRandom)
30         {
31             x = ScoreRandom.Next(15, 500); //405 - 15
32             y = ScoreRandom.Next(15, 500);
33         }
34
35         public void DrawScore(Graphics paper)
36         {
37             ScoreRectangle.X = x;
38             ScoreRectangle.Y = y;
39
40             paper.FillRectangle(so, ScoreRectangle);
41         }
42     }
43 }
```