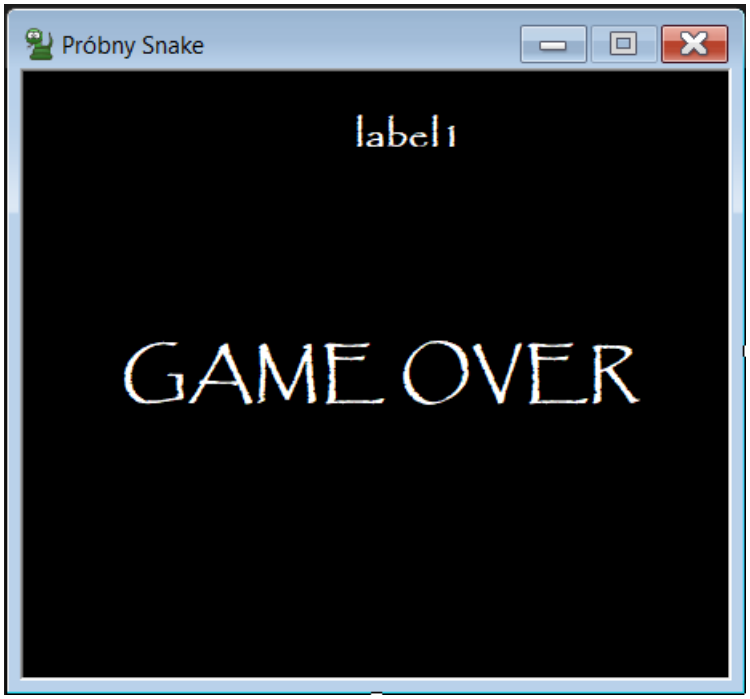
Tworzenie aplikacji postanowiono podzielić na 3 etapy

Najpierw postanowiono wykonać prostszą wersję zamierzonej aplikacji tak, aby mieć pewność, że mechanizm poruszania się Snake będzie działał w sposób poprawny. Następnie na tej aplikacji będą nabudowywane kolejne „warstwy”.

W kolejnym etapie aplikacja zostanie rozszerzona o drugiego Snake, a w trzecim etapie zostanie dodana komunikacja sieciowa.

Etap I



Stworzono okno aplikacji o małych wymiarach (większe nie było na razie konieczne).

Następnie klasy Snake oraz Score (poprzedni raport) zostały dodane do klasy Form1 sterującej oknem. W klasie Form1, znalazła się m.in. obsługa klawiatury, metoda odpowiedzialna za kolizję oraz za „jedzenie” przez węża punktów.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GameS
{
    public partial class Form1 : Form
    {
        Graphics g;
        Snake snake = new Snake();
        Score score;
        Random randScore = new Random();
        public int ScoreQuantity = 0;
        bool game = false;

        bool Up = false;
        bool Down = false;
        bool Left = false;
        bool Right = false;


        public Form1()
        {
            InitializeComponent();
            score = new Score(randScore);
            gameover.Visible = false;
            result.Text = Convert.ToString(ScoreQuantity);
```

```
            game = true;
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            g = e.Graphics;

            snake.DrawSnake(g);
            score.DrawScore(g);
        }


        //Move
        private void Form1_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyData == Keys.Down && Up == false)
            {
                Up = false;
                Down = true;
                Left = false;
                Right = false;
            }

            if (e.KeyData == Keys.Up && Down == false)
            {
                Up = true;
                Down = false;
                Left = false;
                Right = false;
            }

            if (e.KeyData == Keys.Right && Left == false)
            {
                Up = false;
                Down = false;
                Left = false;
                Right = true;
            }

            if (e.KeyData == Keys.Left && Right == false)
            {
                Up = false;
                Down = false;
                Left = true;
                Right = false;
            }
        }


        private void timer1_Tick(object sender, EventArgs e)
        {
            if (Up)
            {
                snake.MoveUp();
            }
            if (Down)
            {
                snake.MoveDown();
            }
            if (Right)
            {
                snake.MoveRight();
            }
            if (Left)
            {
                snake.MoveLeft();
            }

            //speed
            if (ScoreQuantity == 15)
            {
                timer1.Interval = 200;
            }
            if (ScoreQuantity == 30)
            {
                timer1.Interval = 150;
            }
            if (ScoreQuantity == 45)
            {
                timer1.Interval = 100;
            }
            if (ScoreQuantity == 60)
            {
                timer1.Interval = 50;
            }


            this.Invalidate();

            Collision();
            Eat();
        }

        public void Collision()
        {
            //5 przypadków kolizji
```

```
127         public void Collision()
128         {
129             //5 przypadków kolizji
130
131             if (snake.SnakePart[0].X <= 1 || snake.SnakePart[0].X >= 315)
132             {
133                 gameover.Visible = true;
134                 timer1.Stop();
135
136
137
138             }
139             if (snake.SnakePart[0].Y <= 1 || snake.SnakePart[0].Y >= 300)
140             {
141                 gameover.Visible = true;
142                 timer1.Stop();
143
144             }
145
146             for (int i = 1; i < snake.SnakePart.Length; i++)
147             {
148                 if (snake.SnakePart[0].IntersectsWith(snake.SnakePart[i]))
149                 {
150                     gameover.Visible = true;
151                     timer1.Enabled = false;
152
153                 }
154             }
155
156         }
157
158         public void Eat()
159         {
160             if (snake.SnakePart[0].IntersectsWith(score.ScoreRectangle))
161             {
162                 ScoreQuantity += 1;
163                 result.Text = Convert.ToString(ScoreQuantity);
164                 snake.GrowUp(ScoreQuantity);
165
166                 score.ScoreLocation(randScore);
167
168             }
169         }
170
171
172     }
173 }
174
```

Następnie przetestowano działanie całego programu. Aplikacja działała w sposób poprawny, jedyny mankament, który będzie poprawiony w kolejnym etapie tworzenia aplikacji to zderzenie węża ze ścianami, ponieważ aplikacja reaguje na zderzenie z pewnym opóźnieniem.