

СОДЕРЖАНИЕ

| | |
|--|-----------|
| ВВЕДЕНИЕ | 3 |
| 1 Аналитическая часть | 4 |
| 1.1 Формализация проблемы | 4 |
| 1.2 Подход к решению | 4 |
| 1.3 Потенциальные сложности | 5 |
| 2 Конструкторская часть | 6 |
| 2.1 Разработка алгоритма | 6 |
| 2.2 Структура и состав классов | 7 |
| 3 Технологическая часть | 8 |
| 3.1 Выбор языка программирования | 8 |
| 3.2 Сведения о модулях программы | 8 |
| 3.3 Реализация программы | 8 |
| 3.4 Примеры работы программы | 11 |
| ЗАКЛЮЧЕНИЕ | 13 |

ВВЕДЕНИЕ

В условиях быстрого развития городов и значительного увеличения объема дорожного движения эффективное и безопасное управление транспортными системами становится серьезной проблемой. Традиционные светофоры больше не могут удовлетворить потребности многолюдного города, вызывая заторы, аварии и трату времени людей. Одно из решений – использование системы умного светофора.

Целью данной работы является разработка программы контроля движения автомобилей на Т-образном перекрестке с пешеходными переходами.

В рамках выполнения работы необходимо решить следующие задачи:

- формализовать задание;
- провести анализ логику системы;
- построить схемы алгоритма работы система;
- реализовать программу.

1 Аналитическая часть

1.1 Формализация проблемы

Большинство систем светофоров работают по стандартной схеме, установленной на основе исследований движения в определенное время дня или недели. Однако они не способны реагировать на изменения в потоке транспорта в реальном времени.

Многие современные системы управления транспортными не обладают необходимой степенью автоматизации. Даже если есть счетчики движения, их данные редко используются для динамического управления светофорами.

1.2 Подход к решению

Как описано выше, данная работа моделирует ”умную” систему светофоров на Т-образном перекрестке с пешеходными переходами. Последовательность преобразования сигнала типичного светофора можно упрощенно представить следующим образом:

- водители едут;
- оба (водители и пешеходы) ждут;
- пешеходы идут;
- оба (водители и пешеходы) ждут;
- повторить цикл.

«Умный» светофор работает ни в коей мере не так. Это владение информацией об автомобилях на дороге и пешеходах на пешеходном переходе. При наличии пешеходов и автомобилей светофор работает как обычный светофор. Если машин нет, а на пешеходном переходе ждут пешеходы, то им будет гореть зеленый свет до тех пор, пока не появится хотя бы одна машина, и наоборот.

Если нет ни машин, ни пешеходов, то для машин будет гореть зеленый свет, потому что им нужно больше времени, чтобы замедлиться и набрать скорость, чтобы продолжить движение.

Ниже, на таблице 1.1, приведена функциональная логика для «умного» светофора.

Таблица 1.1 – Функциональная логика для «умного» светофора

| | | |
|-------------------|------------------------|----------------------------|
| | Нет пешеходов | Пешеходы существуют |
| Нет машин | Зеленый свет для машин | Зеленый свет для пешеходов |
| Машины существуют | Зеленый свет для машин | Обычный режим светофора |

1.3 Потенциальные сложности

При внедрении системы интеллектуального светофора возникают следующие проблемы, которые необходимо учитывать и решать, чтобы обеспечить эффективность и безопасность системы.

1. Проектирование системы. Чтобы обеспечить гибкость и масштабируемость системы, проект должен быть специально спланирован. Необходимо определить расположение светофоров с учетом транспортного потока, скорости и размера дороги, а также других факторов, таких как перекрестки, кольцевые развязки и близость к школам или жилым районам.
2. Связь и коммуникация. "Умные" системы светофоров должны иметь возможность связываться и взаимодействовать с различными устройствами, такими как датчики, камеры наблюдения, системы управления сигналами и системы управления дорожным движением. Это требует использования общих протоколов и стандартов для обеспечения совместимости и интеграции систем.
3. Управление и обслуживание. Интеллектуальные системы светофоров требуют регулярного управления и обслуживания для обеспечения непрерывной и эффективной работы.

2 Конструкторская часть

2.1 Разработка алгоритма

Ниже, на рисунке 1.1, приведен алгоритм для «умного» светофора, сделанны студентом Динь Вьет Ань.

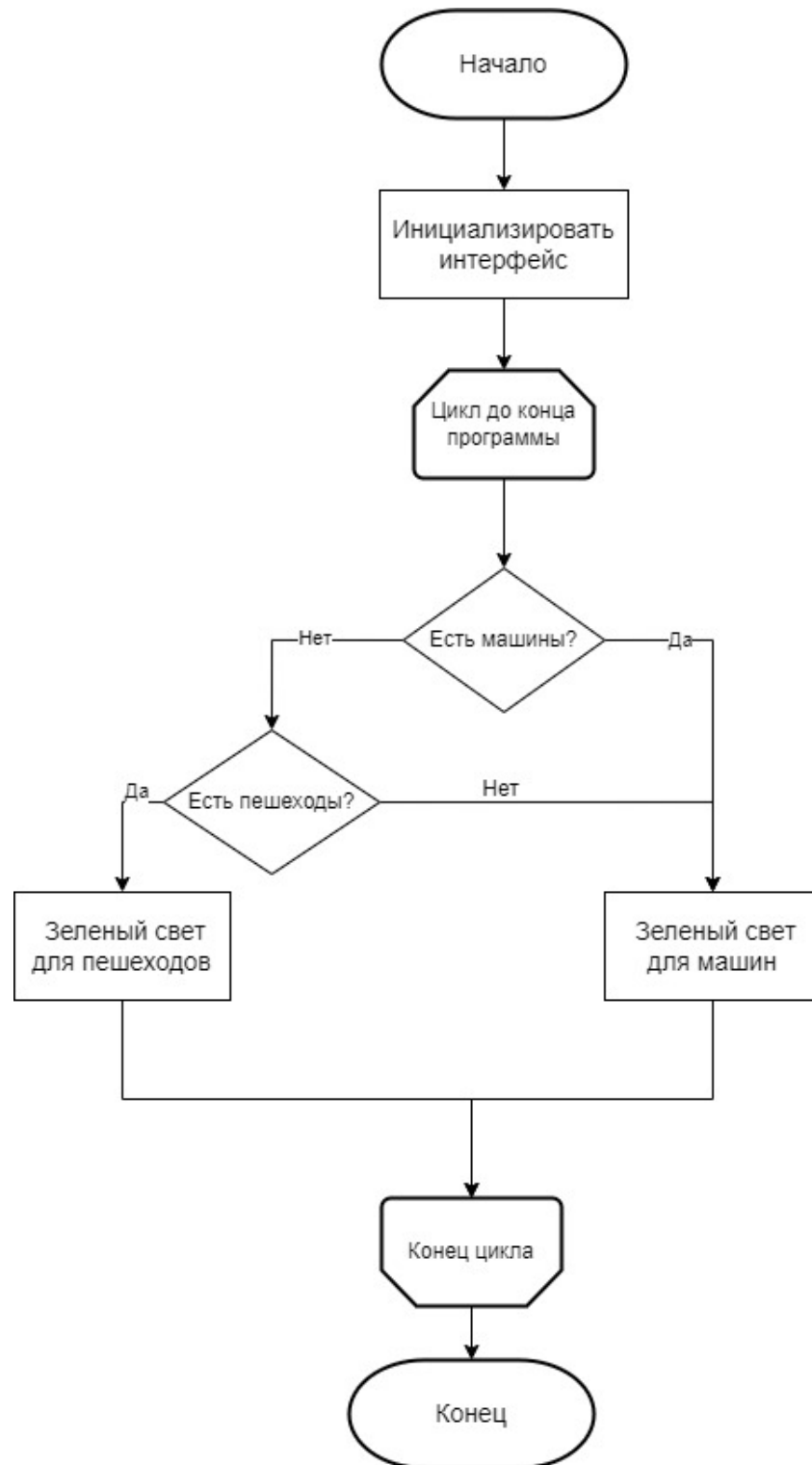


Рисунок 2.1 – Алгоритм для «умного» светофора.

2.2 Структура и состав классов

В программе имеет следующие классы:

- Program — входная точка в программу;
- Form — класс интерфейса;
- trafficlight — сохранит информации о системе светофоров для машин, методы преобразования цвета светофора для машин;;
- pedestrianlight — сохранит информации о системе светофоров для пешеходов, методы преобразования цвета светофора для пешеходов;
- car — сохранит информации о машинах.

3 Технологическая часть

3.1 Выбор языка программирования

В качестве языка программирования был выбран C#, так как:

1. данный язык программирования объектно-ориентирован;
2. имеется навыки использования данного языка программирования.

В качестве среды разработки была выбрана «Visual Studio 2022», так как:

1. студенты могут использовать бесплатно;
2. она имеет множество удобств, которые облегчают процесс написания и отладки кода;
3. она обеспечивает работу с Windows Forms — интерфейсом, который упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обертки для существующего Win32 API в управляемом коде;
4. я знаком с данной средой разработки, что сократит время изучения возможностей среды.

3.2 Сведения о модулях программы

Программа состоит из следующих модулей:

- Program.cs — главная входная точка в программу;
- Form.cs — интерфейс программы;
- trafficlight.cs — описывает систему светофоров для машин, метод преобразования цвета светофора для машин;
- pedestrianlight.cs — описывает систему светофоров для пешеходов, метод преобразования цвета светофора для пешеходов;
- car.cs — описывает информации о машинах.

3.3 Реализация программы

Ниже, на листинге 1 - 3, приведены реализации классов, сделанные студентом Динь Вьет Ань.

Листинг 1 – Класс trafficlight

```
1 class trafficlight
2 {
3     public PictureBox y;
4     public int yellowtime = 2;
5     public int total;
6     public PictureBox r;
7     public PictureBox g;
8     public int redtime = 8;
9     public int greentime = 5;
10    public int counter = 0;
11    public trafficlight(PictureBox red1, PictureBox green1, PictureBox yellow1, Queue queue)
12    {
13        r = red1;
14        g = green1;
```

```

15         y = yellow1;
16         total = yellowtime + greentime;
17         y.Visible = false;
18     }
19     public virtual void turnongreen()
20     {
21         r.Visible = false;
22         g.Visible = true;
23         y.Visible = false;
24     }
25     public virtual void turnonred()
26     {
27         r.Visible = true;
28         g.Visible = false;
29         y.Visible = false;
30     }
31     public void turnonyellow()
32     {
33         r.Visible = false;
34         g.Visible = false;
35         y.Visible = true;
36     }
37     public void switchcolor1(pedestrialight ps1, pedestrialight ps2)
38     {
39         if (r.Visible == true && redtime <= counter && ps1.r.Visible == true && ps2.r.Visible ==
40             true)
41         {
42             turnongreen();
43             counter = 0;
44         }
45         else
46         {
47             if (g.Visible == true && counter >= greentime)
48             {
49                 turnonyellow();
50                 counter = 0;
51             }
52             else
53             {
54                 if (y.Visible == true && counter >= yellowtime)
55                 {
56                     turnonred();
57                     counter = 0;
58                 }
59             }
60         }
61     }
62 }

```

Листинг 2 – Класс pedestrialight

```

1 class pedestrialight
2 {
3     public trafficlight tl;
4     public PictureBox r;
5     public PictureBox g;
6     public int redtime = 8;
7     public int greentime = 5;
8     public int counter = 0;
9     public virtual void turnongreen()
10    {
11        r.Visible = false;
12        g.Visible = true;
13    }
14    public virtual void turnonred()
15    {
16        r.Visible = true;
17        g.Visible = false;
18    }
19    public pedestrialight(PictureBox red, PictureBox green, trafficlight traffic)
20    {
21        r = red;
22        g = green;
23        tl = traffic;
24        redtime = traffic.greentime + traffic.yellowtime;
25        greentime = traffic.redtime;
26    }
27    public virtual void switchcolor(trafficlight tl1, trafficlight tl2, trafficlight tl3)
28    {
29        if (r.Visible == true && redtime <= counter && (tl1.r.Visible == true || tl1.y.Visible ==
30            true) && (tl2.r.Visible == true || tl2.y.Visible == true) && (tl3.r.Visible == true || tl3.y
31                .Visible == true))
32        {
33            turnongreen();
34            counter = 0;
35        }
36        else
37        {
38            if (g.Visible == true && greentime == counter)
39            {
40                turnonred();
41            }
42        }
43    }
44 }

```



```

38         counter = 0;
39     }
40 }
41 }

```

Листинг 3 – Класс car

```

1  class car
2  {
3      public PictureBox pb = new PictureBox();
4      public int direction, road, roadYlim = 200, mid = 250, XFINAL=0, YFINAL=0;
5      bool cross = false;
6      int[] lst;
7      int index;
8      public car(int X0, int Y0, int r, int dir)
9      {
10         pb.SizeMode = PictureBoxSizeMode.StretchImage;
11         pb.BackColor = Color.Transparent;
12         pb.BackgroundImage = Properties.Resources.b;
13         pb.Anchor = AnchorStyles.Left;
14         pb.Location = new Point(X0, Y0);
15         pb.Visible = true;
16         direction = dir;
17         road = r;
18     }
19     public bool passed ()
20     {
21         if (road == 1)
22             if (pb.Location.Y >= roadYlim) return true;
23             else return false;
24         else
25             if (road == 2)
26                 if (pb.Location.X <= roadYlim) return true;
27                 else return false;
28             if (road == 3)
29                 if (pb.Location.X >= roadYlim) return true;
30                 else return false;
31             if (road == 4)
32                 if (pb.Location.Y <= roadYlim) return true;
33                 else return false;
34         return false;
35     }
36     public bool middle()
37     {
38         if (road == 1)
39             if (pb.Location.Y >= mid) return true;
40             else return false;
41         if (road == 2)
42             if (pb.Location.X <= mid) return true;
43             else return false;
44         if (road == 3)
45             if (pb.Location.X >= mid) return true;
46             else return false;
47         if (road == 4)
48             if (pb.Location.Y <= mid) return true;
49             else return false;
50         return false;
51     }
52     public bool crash()
53     {
54         if (road == 1 && !passed() && index > 0)
55             if ((pb.Location.Y + pb.Size.Height + 20) > lst[index - 1])
56                 return true;
57         if (road == 2 && !passed() && index > 0)
58             if ((pb.Location.X - 20 - pb.Size.Width) < lst[index - 1])
59                 return true;
60         if (road == 3 && !passed() && index > 0)
61             if ((pb.Location.X + pb.Size.Width + 20) > lst[index - 1])
62                 return true;
63         if (road == 4 && !passed() && index > 0)
64             if ((pb.Location.Y - pb.Size.Height - 20) < lst[index - 1])
65                 return true;
66         return false;
67     }
68     public int move(trafficlight traffic, int[] l, int i)
69     {
70         lst = l;
71         index = i;
72         if (road == 1)
73         {
74             if (!crash())
75                 if (!passed() || (cross && !middle()))
76                     pb.Top += 10;
77             else
78                 if (traffic.g.Visible) cross = true;
79         }
80         if (middle())
81         {
82             if (direction == 1) pb.Left -= 20;

```

```

82         if (direction == 2) pb.Left += 20;
83         if (direction == 3) pb.Top += 20;
84         return 10000;
85     }
86     return pb.Location.Y;
87 }
88 if (road == 2)
89 {
90     if (!crash())
91         if (!passed() || (cross && !middle()))
92             pb.Left -= 10;
93     else if (traffic.g.Visible) cross = true;
94     if (middle())
95     {
96         if (direction == 1) // go left
97             pb.Top += 20;
98         if (direction == 2) // go right
99             pb.Top -= 20;
100        if (direction == 3) // go straight
101            pb.Left -= 20;
102        return 0;
103    }
104    return pb.Location.X;
105 }
106 if (road == 3)
107 {
108     if (!crash())
109         if (!passed() || (cross && !middle()))
110             pb.Left += 10;
111     else
112         if (traffic.g.Visible) cross = true;
113     if (middle())
114     {
115         if (direction == 1) // go left
116             pb.Top -= 20;
117         if (direction == 2) // go right
118             pb.Top += 20;
119         if (direction == 3) // go straight
120             pb.Left += 20;
121        return 10000;
122    }
123    return pb.Location.X;
124 }
125 if (road == 4)
126 {
127     if (!crash())
128         if (!passed() || (cross && !middle()))
129             pb.Top -= 10;
130     else
131         if (traffic.g.Visible) cross = true;
132     if (middle())
133     {
134         if (direction == 1) // go left
135             pb.Left -= 20;
136         if (direction == 2) // go right
137             pb.Left += 20;
138         if (direction == 3) // go straight
139             pb.Top -= 20;
140        return 0;
141    }
142    return pb.Location.Y;
143 }
144 return pb.Location.X;
145 }
146 }

```

3.4 Примеры работы программы

Ниже, на таблице 3.1 - 3.2, приведены интерфейсы работы программы, сделанны студентом Ву Минь Куанг.

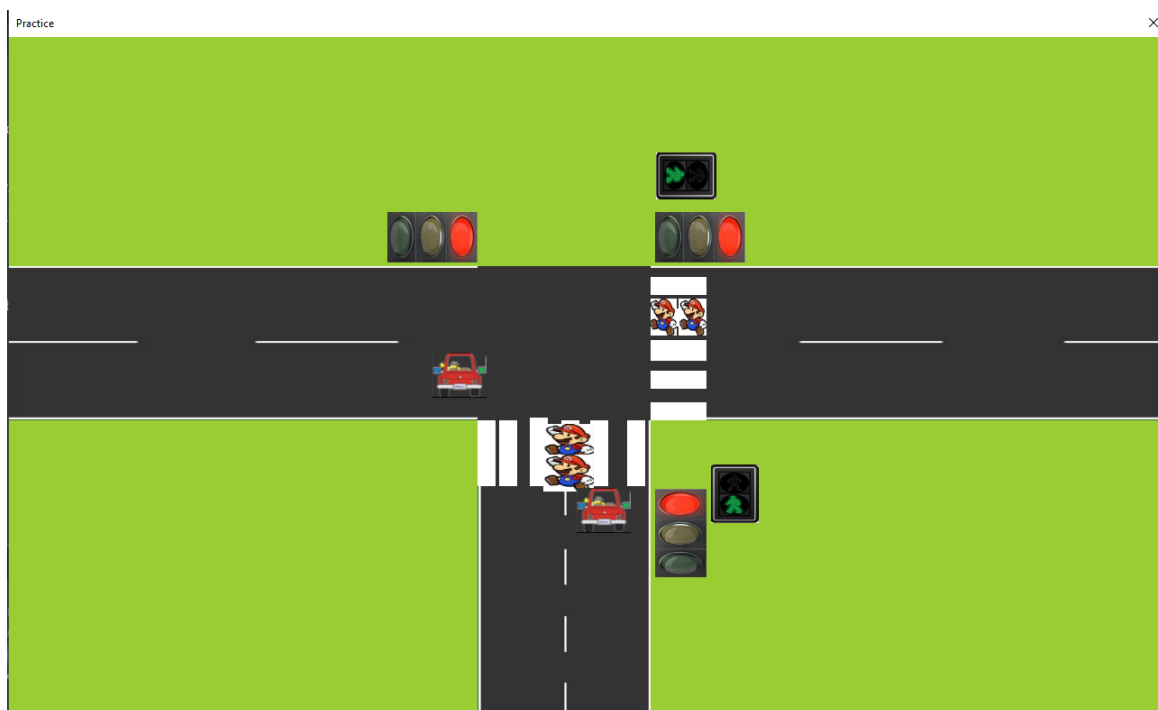


Рисунок 3.1 – Интерфейс работы программы.

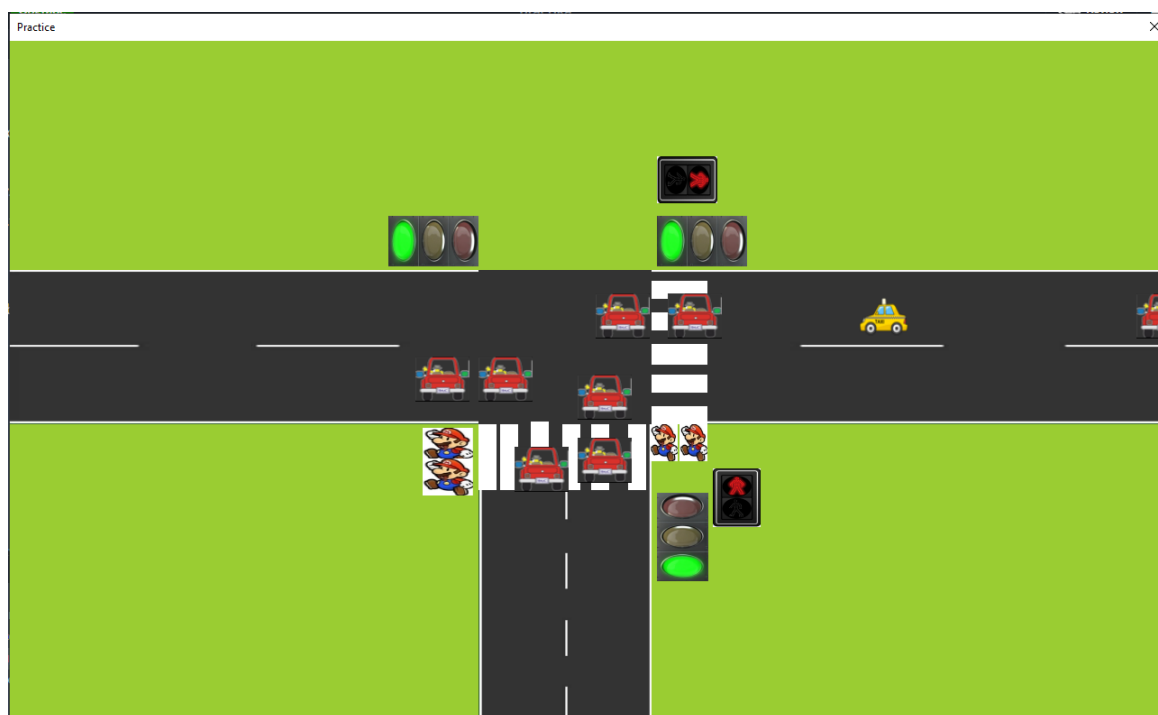


Рисунок 3.2 – Интерфейс работы программы.

ЗАКЛЮЧЕНИЕ

В процессе выполнения работы были формализованы проблему, рассмотрен подход к решению и описаны потенциальные сложности. Также была реализована программа контроля движения автомобилей на Т-образном перекрестке с пешеходными переходами.

В ходе выполнения поставленной задачи были изучены возможности языка C# и библиотеки WindowsForms. Также получен опыт работы по группе.

Цель работы достигнута, выполнены все задачи:

- формализовать задание;
- провести анализ логику системы;
- построить схемы алгоритма работы система;
- реализовать программу.