# ROAD SIMULATION

by

Gerard Aytona
Genovea, Christopher Jed

Submitted in Partial Fulfillment of the Requirements for the

Parallel and Distributed Computing CS0051

at

Instructor: Hadji Tejuco
FEU Institute of Technology

January / 27 / 2025

**TABLE OF CONTENTS**

Chapter 1

# INTRODUCTION

**Purpose:** The purpose of this program is to implement the use of threads in C++ through a traffic light simulation.

**Objectives:** The main aim is to use threads in a real-life application. The program should be able to process two different functions simultaneously using threads.

**Scope:** This program includes the thread feature in C++ and applies it to a traffic simulation involving 8 roads. It is limited to C++ and the command prompt window for output and input.

Chapter 2

# PROJECT OVERVIEW

**Problem Statement:** This project aims to demonstrate the functionality of C++ threads by implementing them in a traffic light simulation for an 8-road intersection. The simulation will showcase how threads can effectively manage concurrent processes. Simulating real-world traffic control highlights the efficiency and synchronisation benefits of multithreading. This project serves as a practical example of how C++ threads can handle complex, time-critical tasks.

**Key Features:** This project presents a detailed demonstration of how threads are implemented to manage traffic lights in a simulation with a visual representation. It showcases the ability of threads to handle concurrent processes, ensuring smooth synchronization across an 8-road intersection. By simulating real-world traffic scenarios, it highlights the practical advantages of multithreading in managing complex systems. The project serves as an insightful example of using C++ threads for efficient and dynamic traffic control.

Chapter 3

# REQUIREMENTS ANALYSIS

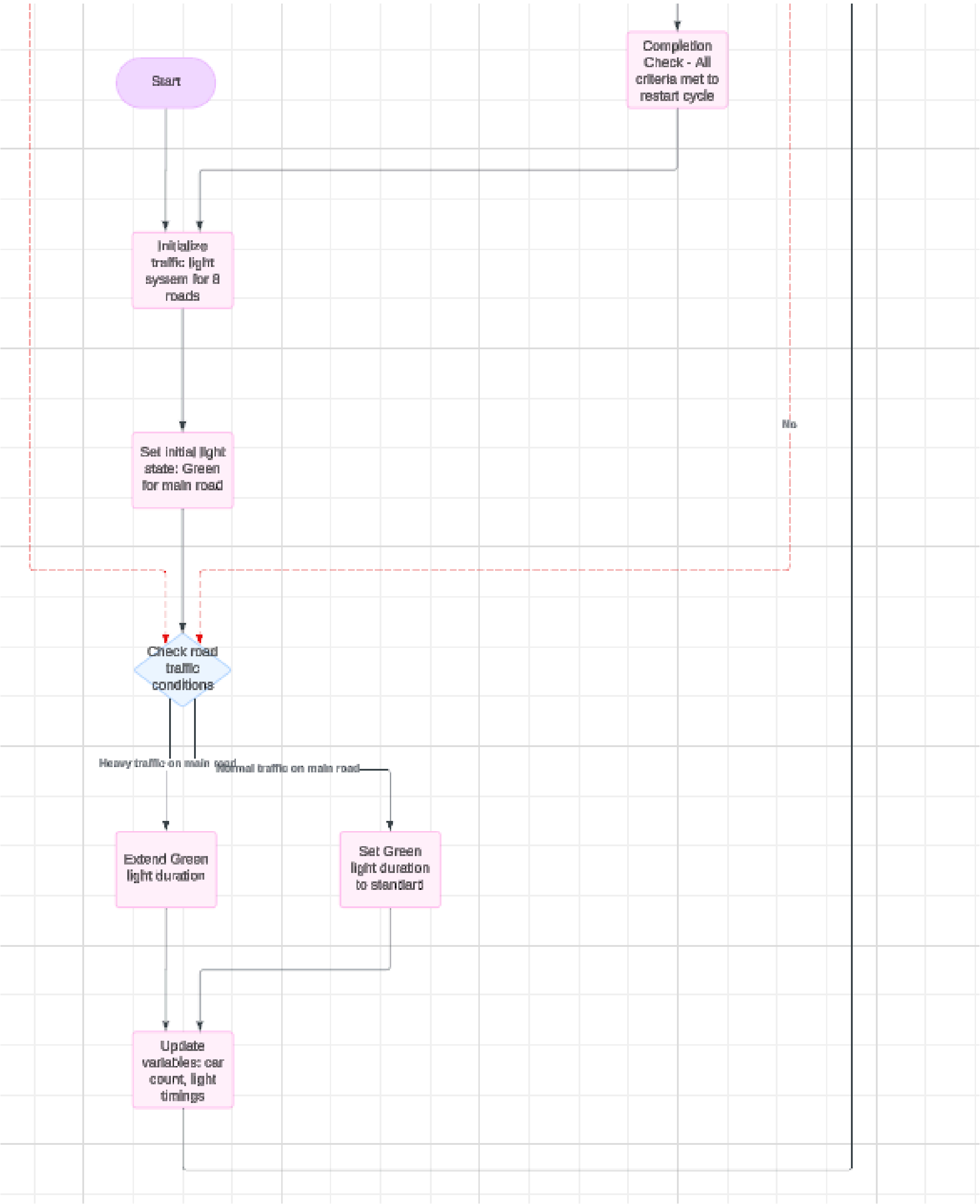
**Functional Requirements:** it needs user input for the program to work

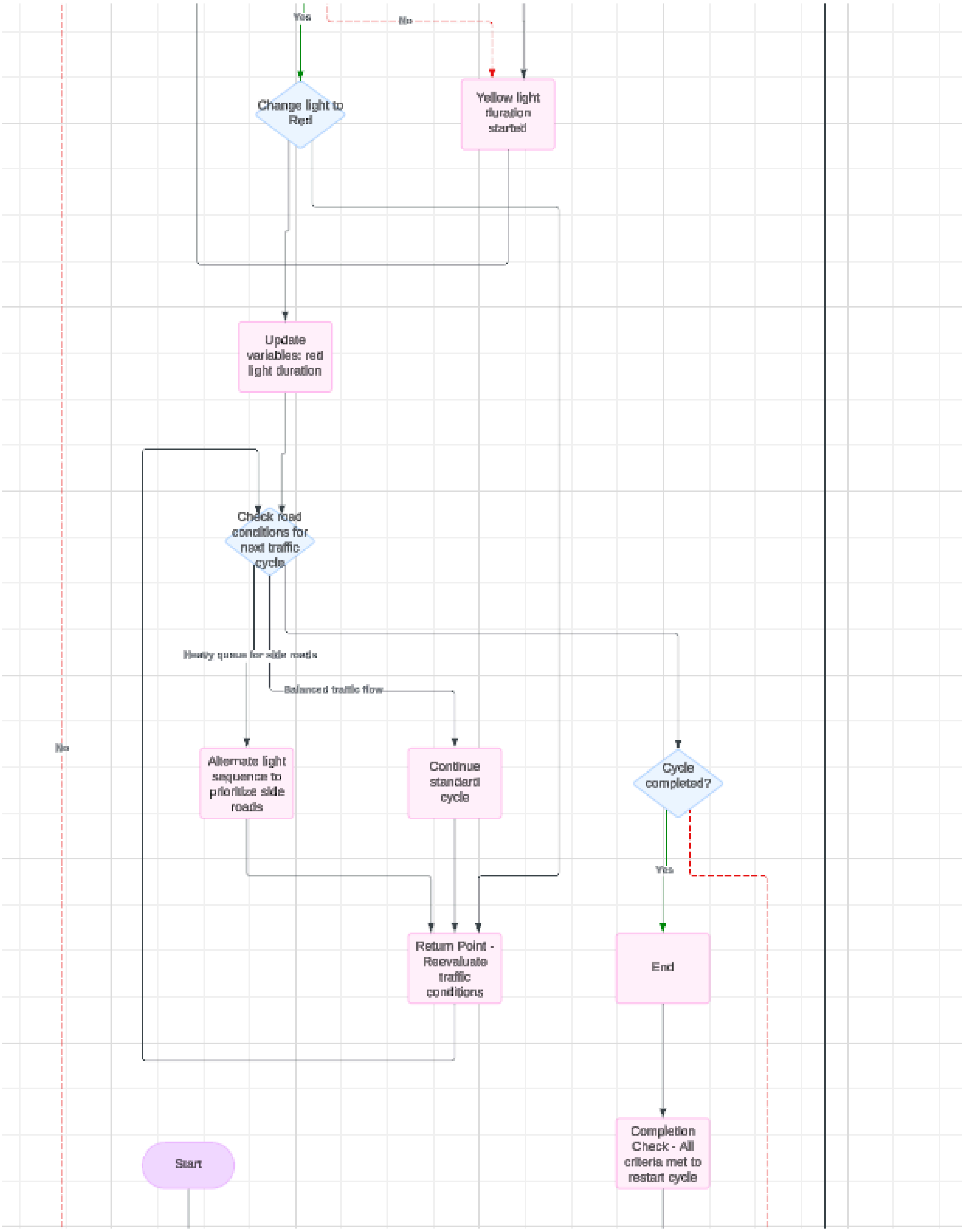
**Non-Functional Requirements:** Include performance, security, and usability requirements.

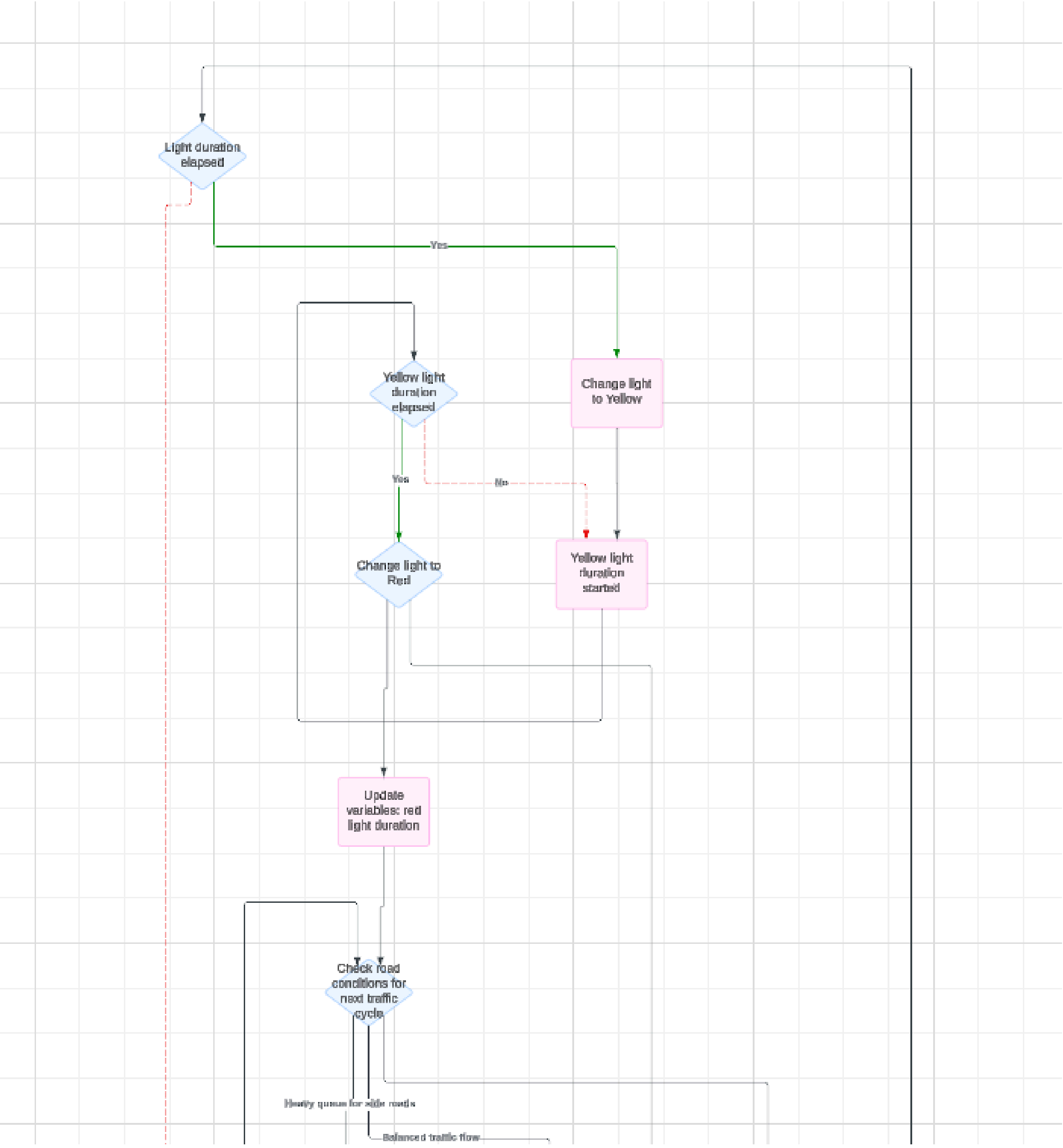
Chapter 4

# SYSTEM DESIGN

**Flowcharts:** Show the flow of processes within the system.


Completion Check - All criteria met to restart cycle
Start
Initialize traffic light system for 8 roads
No
Set initial light state: Green for main road
Check road traffic conditions
Heavy traffic on main road
Normal traffic on main road
Extend Green light duration
Set Green light duration to standard
Update variables: car count, light timings


Yes
No
Change light to Red
Yellow light duration started
Update variables: red light duration
Check road conditions for next traffic cycle
Heavy queue for side roads
Balanced traffic flow
No
Alternate light sequence to prioritize side roads
Continue standard cycle
Cycle completed?
Yes
Return Point - Reevaluate traffic conditions
End
Completion Check - All criteria met to restart cycle
Start


Light duration elapsed
Yes
Yellow light duration elapsed
Change light to Yellow
Yes
No
Change light to Red
Yellow light duration started
Update variables: red light duration
Check road conditions for next traffic cycle
Heavy queue for side roads
Balanced traffic flow

Chapter 5

## IMPLEMENTATION

**Technologies Used:**

Programming Languages

- **C++**

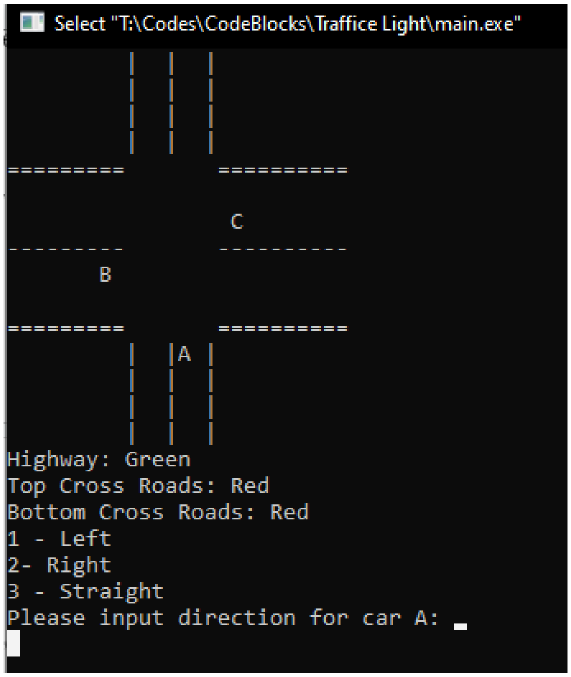
**Libraries:**

- thread
- windows.h
- Iostream

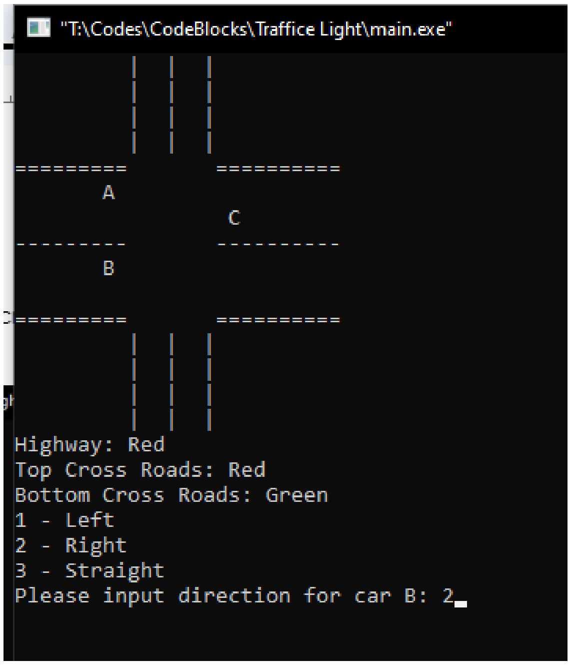
**Tools:**

- Code Blocks

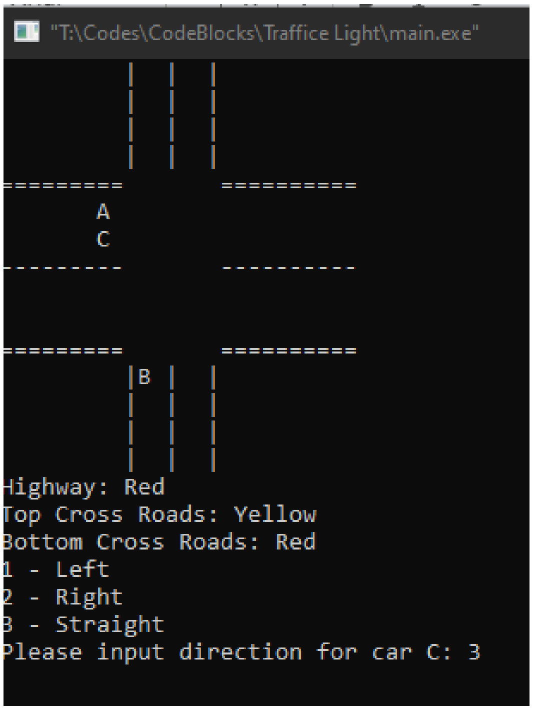
**Screenshots:** Include screenshots of the working software.



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
                 C
---------       ----------
       B

=========       ==========
         |  |A |
         |  |  |
         |  |  |
         |  |  |
Highway: Green
Top Cross Roads: Red
Bottom Cross Roads: Red
1 - Left
2- Right
3 - Straight
Please input direction for car A: 
```



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
                 C
---------       ----------
       B

=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Red
Bottom Cross Roads: Green
1 - Left
2 - Right
3 - Straight
Please input direction for car B: 2
```



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
       C
---------       ----------


=========       ==========
         |B |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Yellow
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car C: 3
```

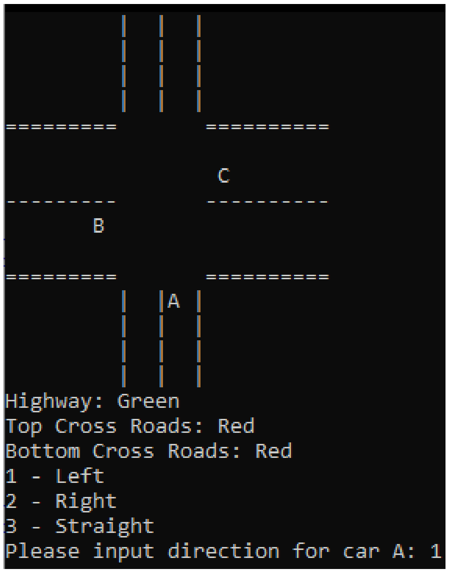
Chapter 6

## TESTING

**Test Cases: List test cases with inputs, expected outputs, and actual results.**

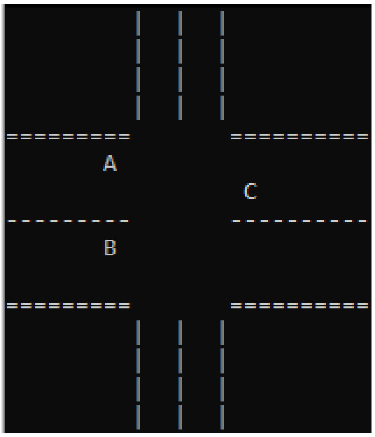
**Case 1:**

**Input:** 1



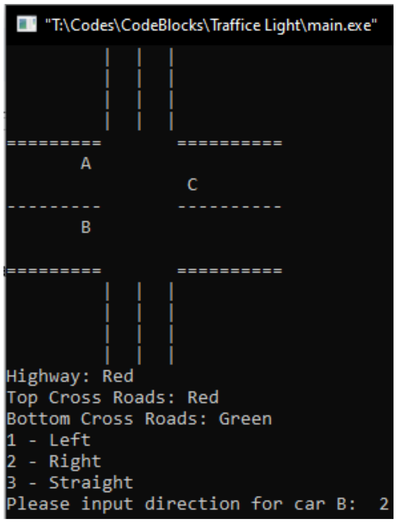
```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========

                 C
---------       ----------
       B

=========       ==========
         |  |A |
         |  |  |
         |  |  |
         |  |  |
Highway: Green
Top Cross Roads: Red
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car A: 1
```

**Expected Outputs:** A turns to the left. Traffic lights change to Red, Red, and Green
**Actual Results:** A turns to the left. Traffic lights change to Red, Red, and Green



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
                 C
---------       ----------
       B

=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
```

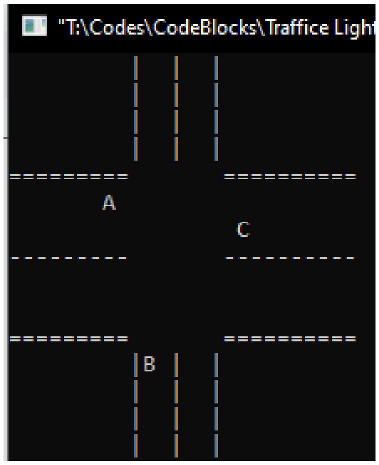
**Input:** 2



"T:\Codes\CodeBlocks\Traffice Light\main.exe"

```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
                 C
---------       ----------
       B

=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Red
Bottom Cross Roads: Green
1 - Left
2 - Right
3 - Straight
Please input direction for car B:  2
```

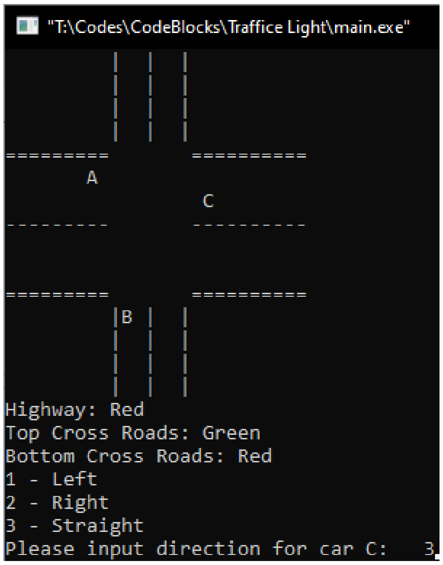
**Expected Outputs:** B turns right. Traffic lights change to Red, Green, and Red
**Actual Results:** B turns right. Traffic lights change to Red, Green, and Red



"T:\Codes\CodeBlocks\Traffice Light

```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
                 C
---------       ----------


=========       ==========
         |B |  |
         |  |  |
         |  |  |
         |  |  |
```

**Input:** 3

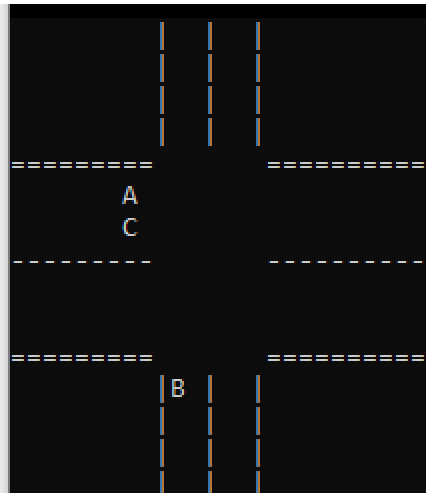


"T:\Codes\CodeBlocks\Traffice Light\main.exe"

```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
                 C
---------       ----------


=========       ==========
         |B |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Green
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car C:   3
```

**Expected Outputs:** C goes straight. Traffic lights change to Green, Red, and Red

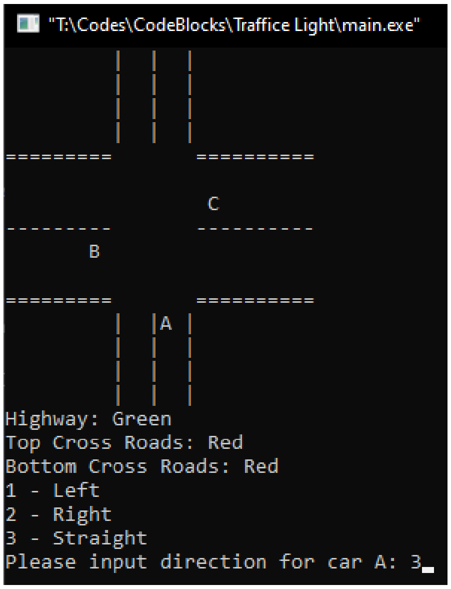
**Actual Results:** C goes straight. Traffic lights change to Green, Red, and Red



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
       C
---------       ----------


=========       ==========
         |B |  |
         |  |  |
         |  |  |
         |  |  |
```

**Case 2:**

**Input:** 3

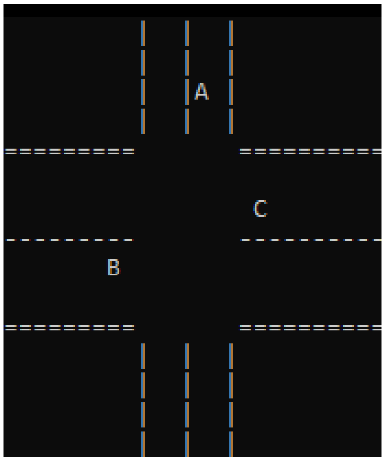


"T:\Codes\CodeBlocks\Traffice Light\main.exe"

```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========

                 C
---------       ----------
       B

=========       ==========
         |  |A |
         |  |  |
         |  |  |
         |  |  |
Highway: Green
Top Cross Roads: Red
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car A: 3
```

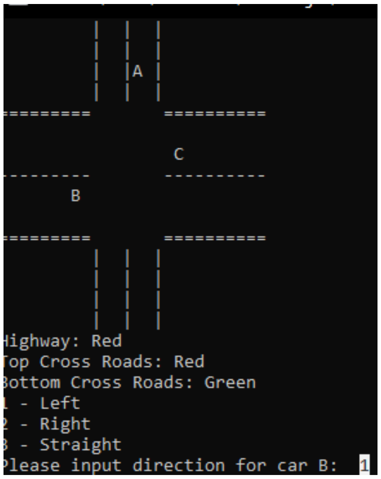
**Expected Outputs:** A goes straight. Traffic lights change to Red, Red, and Green

**Actual Results:** A goes straight. Traffic lights change to Red, Red, and Green



```
         |  |  |
         |  |  |
         |  |A |
         |  |  |
=========       ==========

                 C
---------       ----------
       B

=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
```

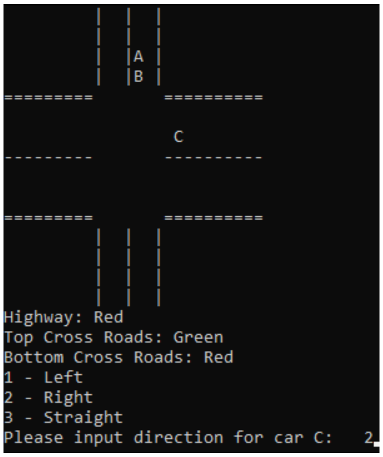
**Input:** 1



```
         |  |  |
         |  |  |
         |  |A |
         |  |  |
=========      ==========

                C
---------      ----------
      B

=========      ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Red
Bottom Cross Roads: Green
1 - Left
2 - Right
3 - Straight
Please input direction for car B:  1
```

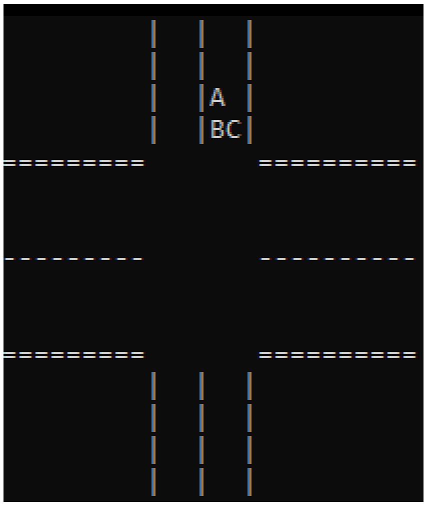
**Expected Outputs:** B turns left. Traffic lights change to Red, Green, and Red
**Actual Results:** B turns left. Traffic lights change to Red, Green, and Red

**Input:** 2



```
         |  |  |
         |  |  |
         |  |A |
         |  |B |
=========      ==========

                C
---------      ----------


=========      ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Green
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car C:   2
```

**Expected Outputs:** C turns right. Traffic lights change to Green, Red, and Red
**Actual Results:** C turns right. Traffic lights change to Green, Red, and Red



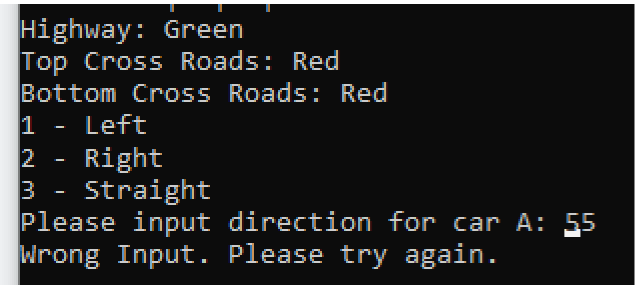
```
         |  |  |
         |  |  |
         |  |A |
         |  |BC|
=========       ==========

---------       ----------

=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
```

**Case 3:**

**Input:** 5

**Expected Outputs:** Outputs “Wrong Input. Please try again.” then retry another input

**Actual Results:** Outputs “Wrong Input. Please try again.” then retry another input

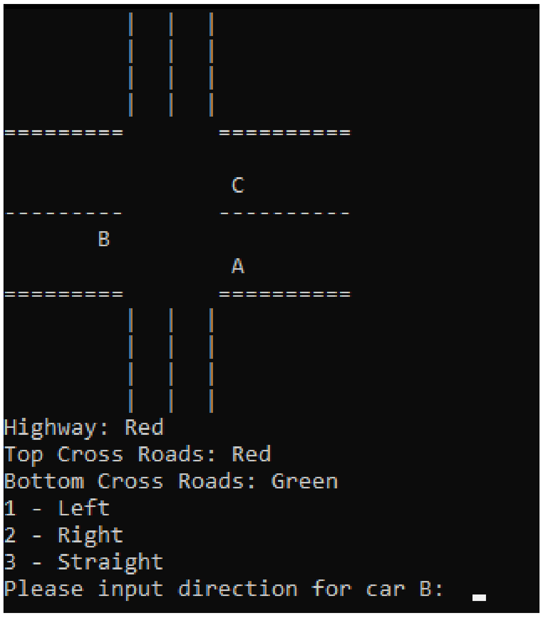


```
Highway: Green
Top Cross Roads: Red
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car A: 55
Wrong Input. Please try again.
```

**Input:** 2

**Expected Outputs:** A turns right. Traffic lights change to Red, Red, and Green

**Actual Results:** A turns right. Traffic lights change to Red, Red, and Green

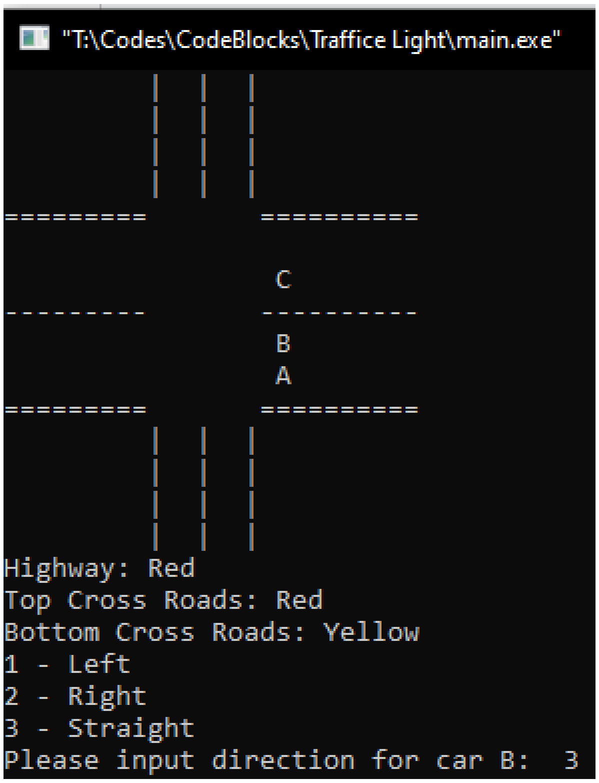


```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========

                 C
---------       ----------
       B
                 A
=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Red
Bottom Cross Roads: Green
1 - Left
2 - Right
3 - Straight
Please input direction for car B:  
```

**Input:** 3

**Expected Outputs:** B goes straight. Traffic lights change to Red, Red, and Green

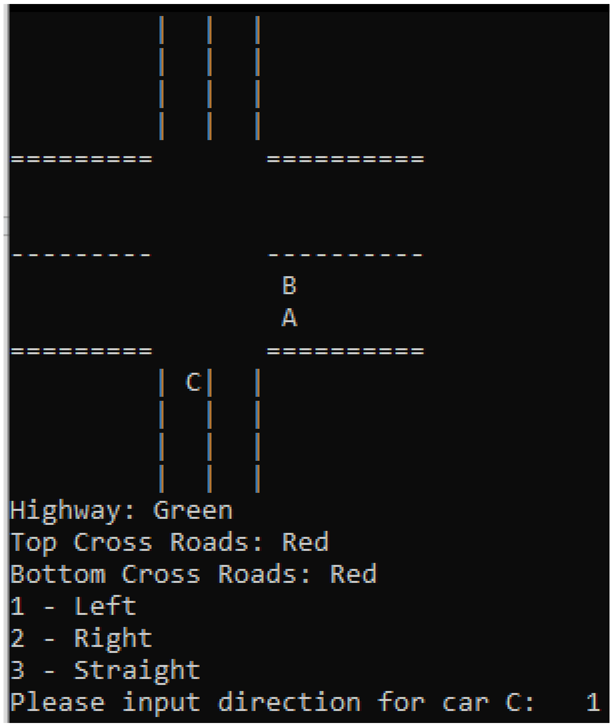
**Actual Results:** B goes straight. Traffic lights change to Red, Red, and Green



"T:\Codes\CodeBlocks\Traffice Light\main.exe"

```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========

                 C
---------       ----------
                 B
                 A
=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Red
Bottom Cross Roads: Yellow
1 - Left
2 - Right
3 - Straight
Please input direction for car B:  3
```

**Input:** 1
**Expected Outputs:** C turns right. Traffic lights change to Red, Red, and Green
**Actual Results:** C turns right. Traffic lights change to Red, Red, and Green



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========


---------       ----------
                 B
                 A
=========       ==========
         | C|  |
         |  |  |
         |  |  |
         |  |  |
Highway: Green
Top Cross Roads: Red
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car C:   1
```

Results: The first case shows cars A and C at the same crossroad while B went through the highway. The second case should have all three cars on the same highway without interrupting their visual cue. The third case shows the error handling of the project by showing an error message and a retry of the input. The result should show cars A and B together at the same crossroads and car C going through the highway.

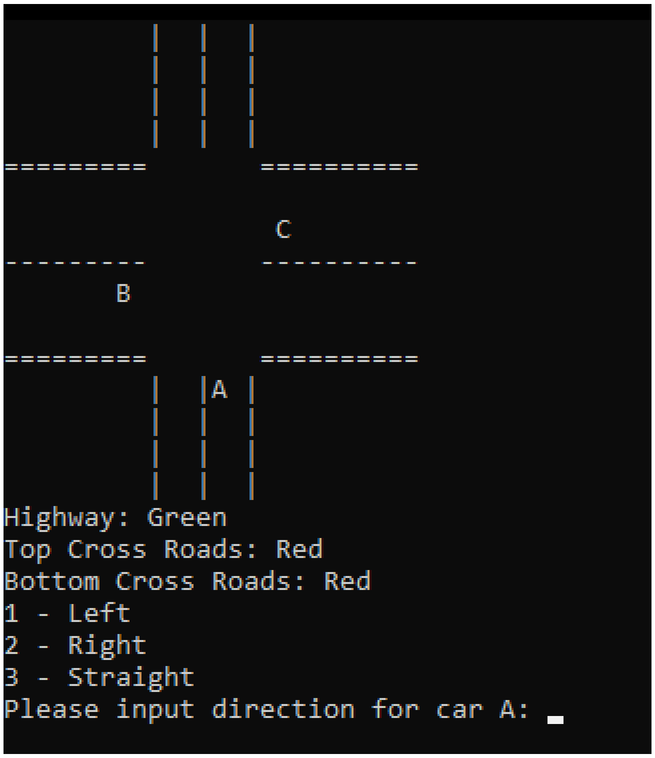
## Chapter 7

## USER MANUAL

**Installation Guide:** The first step is to download the C++ project. The second step is to open code blocks or Dev C++ and open the project. The third step is to press the run button on the main.cpp file that showed when opening the project.

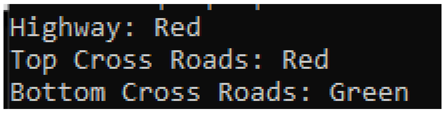
**Usage Instructions:**

1. Once running, a prompt will ask you to input one of the three options 1, 2, or 3 each with corresponding directions for car A to go.



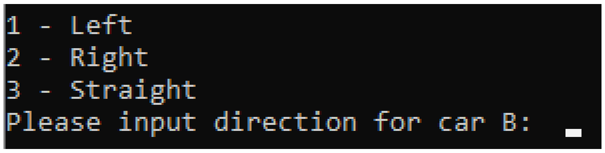
```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========

                 C
---------       ----------
       B

=========       ==========
         |  |A |
         |  |  |
         |  |  |
         |  |  |
Highway: Green
Top Cross Roads: Red
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car A:
```

2. After inputting the direction, the traffic lights will change indicating which car will move next.



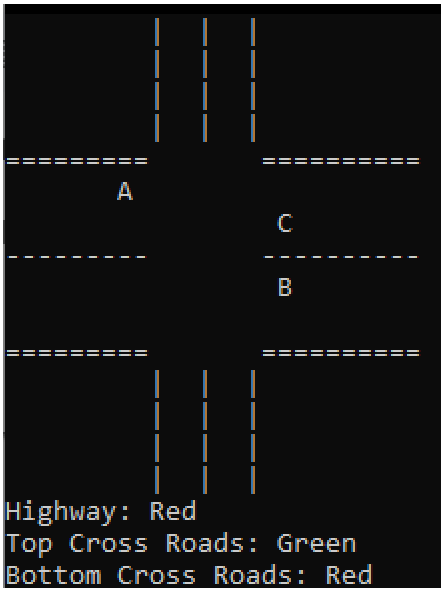
```
Highway: Red
Top Cross Roads: Red
Bottom Cross Roads: Green
```

3. After the lights have changed. Another prompt will ask you for the direction of Car B.



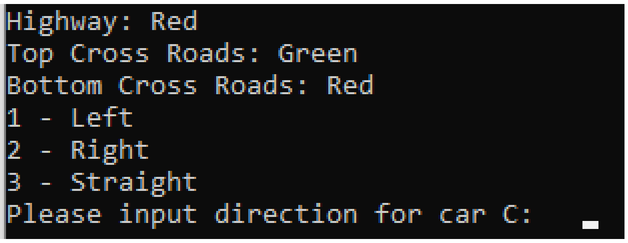
```
1 - Left
2 - Right
3 - Straight
Please input direction for car B:
```

4. Once you input the direction, the visuals will change again for Car B this time. Wait for the traffic lights to change again.



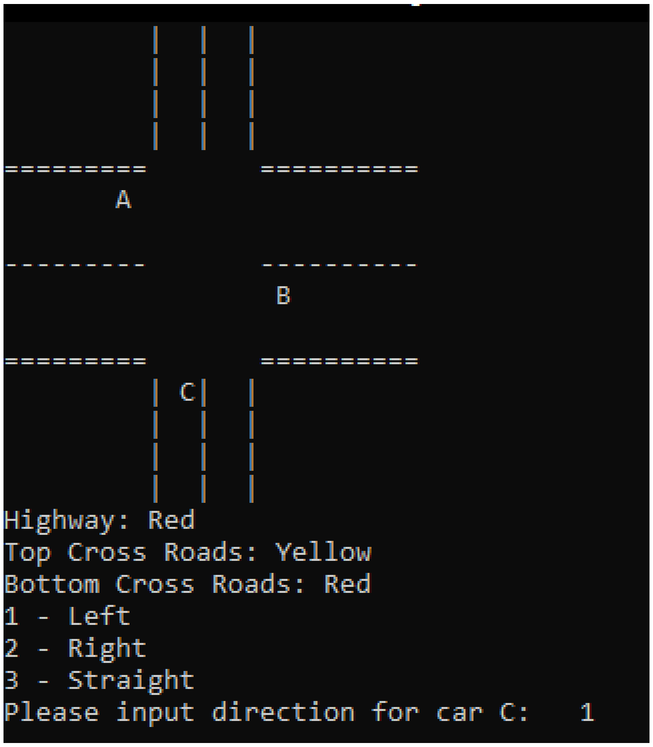
```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A
                 C
---------       ----------
                 B

=========       ==========
         |  |  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Green
Bottom Cross Roads: Red
```

5. After the lights have changed. Another prompt will ask you for the direction of Car C.



```
Highway: Red
Top Cross Roads: Green
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car C:
```

6. After inputting, the program will end and show you the final visual of the map.



```
         |  |  |
         |  |  |
         |  |  |
         |  |  |
=========       ==========
       A

---------       ----------
                 B

=========       ==========
         | C|  |
         |  |  |
         |  |  |
         |  |  |
Highway: Red
Top Cross Roads: Yellow
Bottom Cross Roads: Red
1 - Left
2 - Right
3 - Straight
Please input direction for car C:   1
```

## Chapter 8

## CHALLENGES AND SOLUTIONS

**Challenges Faced:**

- The Traffic Light Algorithm
- Visuals for traffic lights, cars, and roads
- Application of threads

**Solutions Implemented:**

- We used a switch case statement for the traffic light where three traffic lights will switch from red to green and vice versa between each other. A shared integer between the cases will change to ensure the next case will be called when the traffic light function is called again.
- We used a function called gotoxy which uses the windows.h library to specify where the text cursor should be. We then made the text-based visual of traffic lights, cars, and roads.
- We applied the threads when making the cars move and the traffic lights countdown.

## Chapter 9

## FUTURE ENHANCEMENTS

**Potential Improvements:**

- A countdown feature for the traffic lights
- Recurring program where the user can choose to end it or keep the cars changing lanes
- Better Traffic light visuals

Chapter 10

## CONCLUSION

**Summary:** It simulates a traffic scenario with cars (A,B, C) navigating a highway and crossroads intersection, along with changing traffic lights. The project demonstrates a simplified yet interactive traffic management simulation, combining user input, and threading.

**Lessons Learned:** We have learned the implementation of threads in C++ and the importance of using threads to execute two different sets of instructions or actions.

Chapter 11

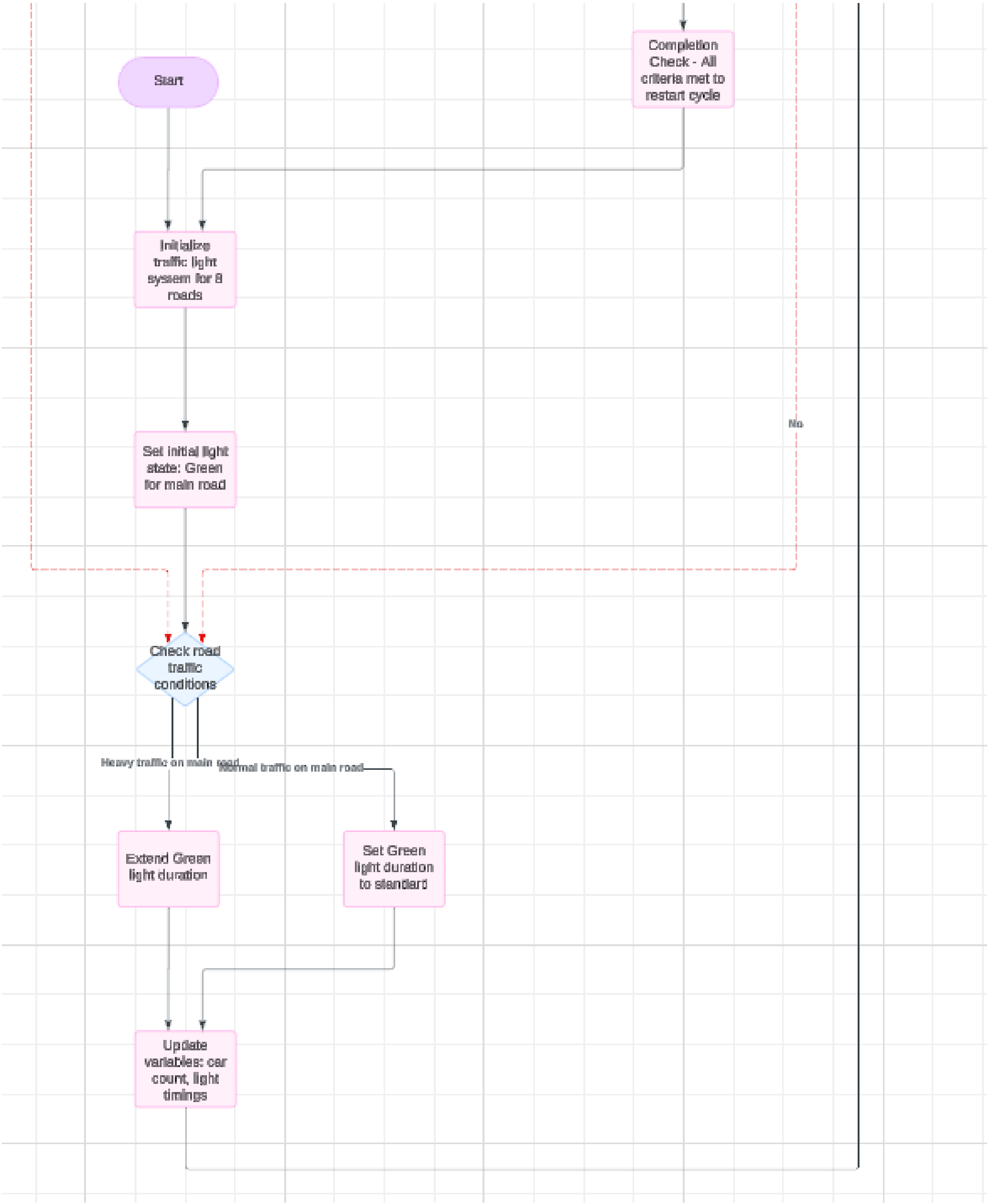
## REFERENCES

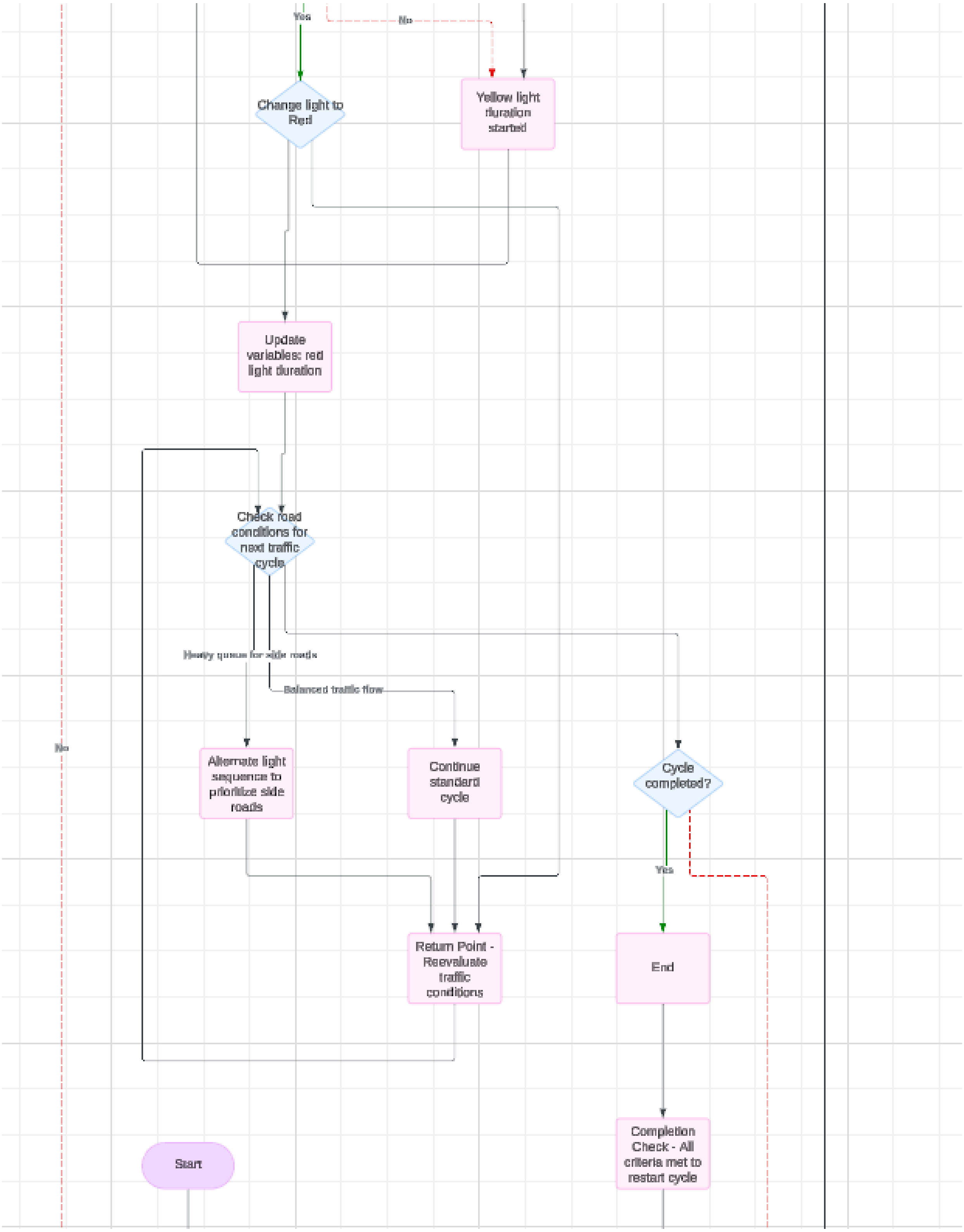
**List all the resources, tools, and references used in the project (e.g., books, articles, websites).**

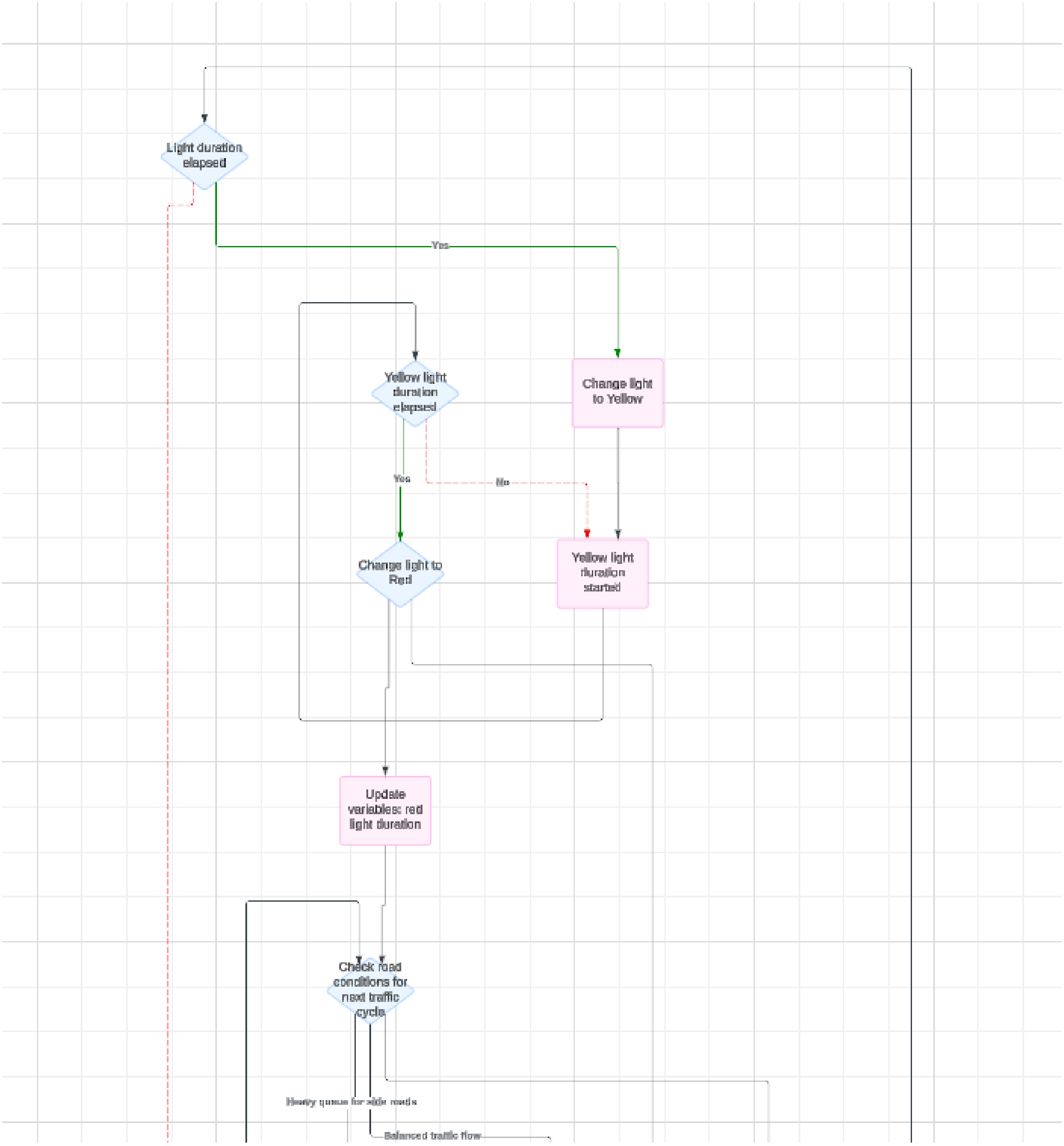
https://en.cppreference.com/w/cpp/language/goto
https://en.cppreference.com/w/cpp/thread
https://cppqa.blogspot.com/2013/11/gotoxy-function-in-c-by-umair-sajid.html#:~:text=The%20gotoxy%20function%20is%20used,specified%20location%20on%20the%20screen.

## Appendices

## Appendix A: Additional diagrams or charts.


Completion Check - All criteria met to restart cycle
Start
Initialize traffic light sysem for 8 roads
Set initial light state: Green for main road
No
Check road traffic conditions
Heavy traffic on main road
Normal traffic on main road
Extend Green light duration
Set Green light duration to standard
Update variables: car count, light timings


Yes
No
Change light to Red
Yellow light duration started
Update variables: red light duration
Check road conditions for next traffic cycle
Heavy queue for side roads
Balanced traffic flow
No
Alternate light sequence to prioritize side roads
Continue standard cycle
Cycle completed?
Yes
Return Point - Reevaluate traffic conditions
End
Completion Check - All criteria met to restart cycle
Start


Light duration elapsed
Yes
Yellow light duration elapsed
Change light to Yellow
Yes
No
Change light to Red
Yellow light duration started
Update variables: red light duration
Check road conditions for next traffic cycle
Heavy queue for side roads
Balanced traffic flow

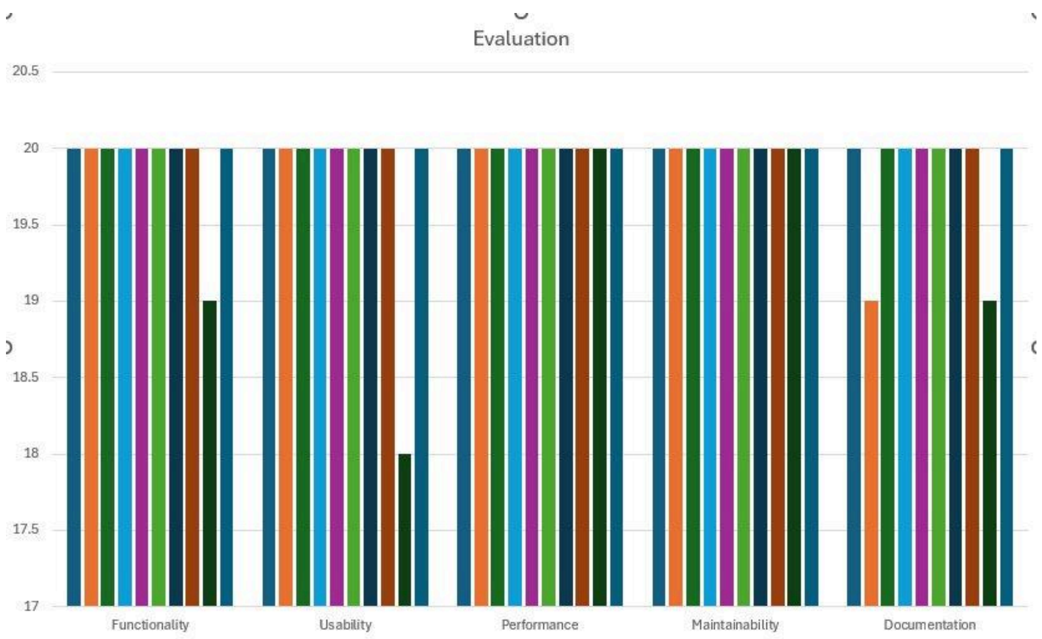
**Appendix B:**

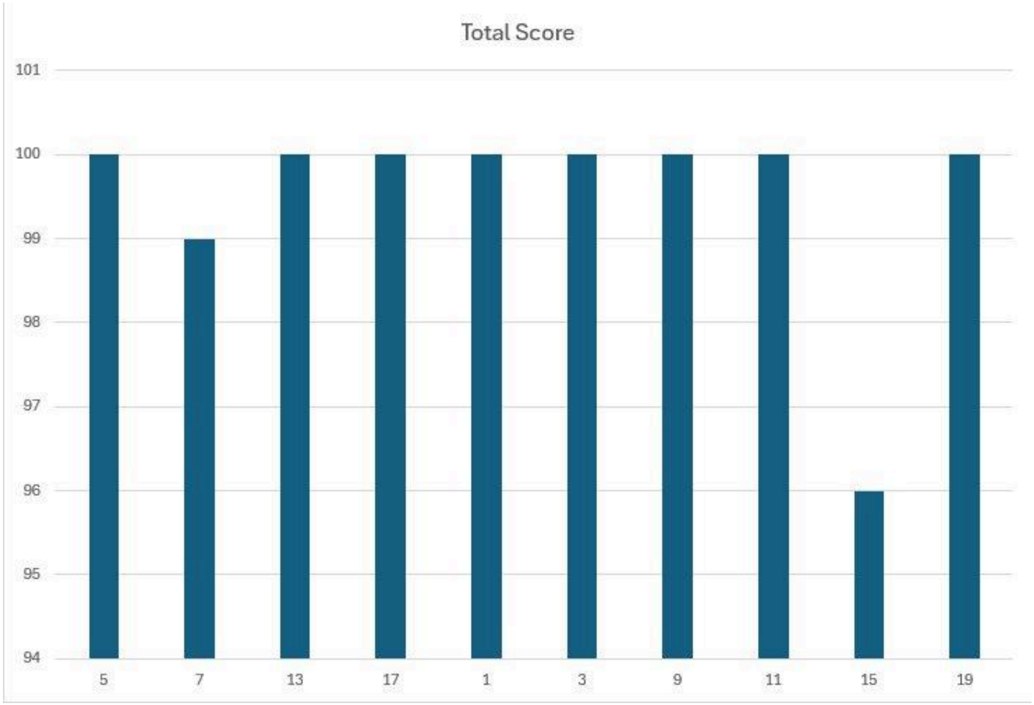
```
#include <iostream>
#include <windows.h>
#include <thread>
using namespace std;

int Lights = 1;
```

```
void gotoxy(int x, int y)//used to change text cursor position (Don't touch)
{
   COORD coordinate;
   coordinate.X = x;
   coordinate.Y = y;
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coordinate);
}

void street()//makes the road format
{
   int x = 9;

   for(int a = 0; a < 2; a++)// top part of the highway
   {
      for(int i = 0; i < 4; i++)
      {
         gotoxy(x,i);
         cout << "|";
      }
      x = 15;
   }
   x = 9;
   for(int a = 0; a < 2; a++)// bottom part of the highway
   {
      for(int i = 11; i < 15; i++)
      {
         gotoxy(x,i);
         cout << "|";
      }
      x = 15;
   }
   x = 4;
   for(int a = 0; a < 2; a++)  // left part of the side road
   {
      for(int i = 0; i < 9; i++)
      {
         gotoxy(i,x);
         cout << "=";
      }
      x = 10;
   }
   x = 4;
   for(int a = 0; a < 2; a++) // right part of the side road
   {
      for(int i = 16; i < 26; i++)
      {
         gotoxy(i,x);
         cout << "=";
      }
      x = 10;
   }
   x = 12;
   for(int i = 0; i < 15; i++) //middle part of the highway
   {
      gotoxy(x,i);
      if(i < 4 || i > 10 )
            cout << "|";
   }
   x = 7;
   for(int a = 0; a < 2; a++) //middle part of the side road
   {
      for(int i = 0; i < 26; i++)
      {
         gotoxy(i,x);
         if(i < 9 || i > 15 )
            cout << "-";
      }
   }

   gotoxy(13,11);
   cout << "A";
   gotoxy(7,8);
   cout << "B";
   gotoxy(17,6);
   cout << "C";

   gotoxy(0,15);
   cout << "Highway: Green";
   gotoxy(0,16);
```

```
    cout << "Top Cross Roads: Red";
    gotoxy(0,17);
    cout << "Bottom Cross Roads: Red";

}

void UpdateA(int a) //A checks direction
{
    switch(a)
    {
        case 1:
            gotoxy(13,11);
            cout << " ";
            gotoxy(13,8);
            cout << "A";
            Sleep(1000);

            gotoxy(13,8);
            cout << " ";
            gotoxy(11,6);
            cout << "A";
            Sleep(1000);

            gotoxy(11,6);
            cout << " ";
            gotoxy(7,5);
            cout << "A";
            break;
        case 2:
            gotoxy(13,11);
            cout << " ";
            gotoxy(14,10);
            cout << "A";
            Sleep(1000);

            gotoxy(14,10);
            cout << " ";
            gotoxy(17,9);
            cout << "A";
            break;
        case 3:
            gotoxy(13,11);
            cout << " ";
            gotoxy(13,7);
            cout << "A";
            Sleep(1000);

            gotoxy(13,7);
            cout << " ";
            gotoxy(13,2);
            cout << "A";
            break;

        default:
            cout << "Lol" << endl;
    }
}

void UpdateB(int a) // A checks direction
{
    switch(a)
    {
        case 1:
            gotoxy(7,8);
            cout << " ";
            gotoxy(12,7);
            cout << "B";
            Sleep(1000);

            gotoxy(12,7);
            cout << " ";
            gotoxy(13,3);
            cout << "B";
            break;
        case 2:
            gotoxy(7,8);
            cout << " ";
            gotoxy(10,9);
            cout << "B";
            Sleep(1000);
            gotoxy(10,9);
            cout << " ";
            gotoxy(10,11);
            cout << "B";
            break;
        case 3:
            gotoxy(7,8);
            cout << " ";
            gotoxy(13,8);
            cout << "B";
            Sleep(1000);

            gotoxy(13,8);
            cout << " ";
            gotoxy(17,8);
            cout << "B";
            break;
        default:
            cout << "Lol" << endl;
    }
}

void UpdateC(int a) // A checks direction
{
    switch(a)
    {
        case 1:
            gotoxy(17,6);
            cout << " ";
            gotoxy(12,7);
            cout << "C";
            Sleep(1000);

            gotoxy(12,7);
            cout << " ";
            gotoxy(11,11);
            cout << "C";
            break;
        case 2:
            gotoxy(17,6);
            cout << " ";
            gotoxy(15,5);
            cout << "C";
            Sleep(1000);

            gotoxy(15,5);
            cout << " ";
            gotoxy(14,3);
            cout << "C";
            break;
        case 3:
            gotoxy(17,6);
            cout << " ";
            gotoxy(13,6);
            cout << "C";
            Sleep(1000);

            gotoxy(13,6);
            cout << " ";
            gotoxy(7,6);
            cout << "C";
            break;
        default:
            cout << "Lol" << endl;
    }
}

void Allofthelights() //Traffic light for the roads
{      //If Lights is 1 highway is red then the bottom cross road is green and the top one is red
    switch(Lights)//if 2 highway is red, bottom cross road is red, and top cross road is green
    {//Programs starts with Highway as green
        case 1:
            Sleep(3000);            //Switch case to filter through which traffic light is green or red
            gotoxy(9,15);
            cout << "Yellow" << endl;
            Sleep(5000);

            gotoxy(9,15);
            cout << "Red   " << endl;
            gotoxy(20,17);
            cout << "Green " << endl;
            Lights = 2;
            break;

        case 2:
            Sleep(3000);
            gotoxy(20,17);
            cout << "Yellow" << endl;
            Sleep(5000);

            gotoxy(20,17);
            cout << "Red   " << endl;
            gotoxy(17,16);
            cout << "Green " << endl;
            Lights = 3;
            break;
        case 3:
            Sleep(3000);
            gotoxy(17,16);
            cout << "Yellow" << endl;
            Sleep(5000);

            gotoxy(17,16);
            cout << "Red   " << endl;
            gotoxy(9,15);
            cout << "Green " << endl;
            Lights = 1;
            break;
        default:
            cout << "Lol";
            break;
    }
}

bool CheckInput(int check) //checks if the input is within the options
{
    if(check < 1 || check > 3)
    {
        gotoxy(0,22);
        cout << "Wrong Input. Please try again.";
        return true;
    }
    else
    {
        gotoxy(0,22);
        cout << "                              ";
        return false;
    }
}

int main()
{
    int wayA = 0, wayB = 0, wayC = 0;

    street(); //initializes the visuals

    WrongA: //Label to go back to when Input for A is wrong
    gotoxy(0,18);//coordinates to be printed at
            cout << "1 - Left\n2 - Right\n3 - Straight\n"
         << "Please input direction for car A: ";
    cin >> wayA;

    if(CheckInput(wayA)) //calls function to check input
        goto WrongA;

    thread CA(UpdateA,wayA);
    thread t1(Allofthelights);
    CA.join();
    t1.join();

    WrongB: //Label to go back to when Input for B is wrong
    gotoxy(0,18);
    cout << "1 - Left\n2 - Right\n3 - Straight\n"
         << "Please input direction for car B:  ";
    cin >> wayB;

    if(CheckInput(wayB)) //calls function to check input
        goto WrongB;
    thread CB(UpdateB,wayB);
    thread t2(Allofthelights);
    CB.join();
    t2.join();

    WrongC: //Label to go back to when Input for C is wrong
    gotoxy(0,18);
    cout << "1 - Left\n2 - Right\n3 - Straight\n"
         << "Please input direction for car C:   ";
    cin >> wayC;

    if(CheckInput(wayC)) //calls function to check input
        goto WrongC;

    thread CC(UpdateC,wayC);
    thread t3(Allofthelights);
    CC.join();
    t3.join();

    gotoxy(0,22);
    system("pause");
    return 0;
}
```

## Appendix C: Any other relevant material.


Evaluation
20.5
20
19.5
19
18.5
18
17.5
17
Functionality
Usability
Performance
Maintainability
Documentation


Total Score
101
100
99
98
97
96
95
94
5
7
13
17
1
3
9
11
15
19