

# Intro to C++ Final Project

## Traffic Light Simulation

Adam Haugen  
180378  
April 11th, 2024

## Intro

For my final project, I chose to do the Traffic Light Simulation. This simulation requires east-to-west and south-to-north traffic to be simulated as well as functioning traffic lights which adjust based on the traffic density. All of this needs to take into account the safety of our vehicles, the fairness of wait times and priorities, and the efficiency of the light timings. I will attach a separate README file detailing how to use my program.

## Difficulties

I had a lot of difficulties with this project and spent a long time replanning and rewriting my code to fit the restrictions I kept finding myself with. Most of these issues were with the `simplecpp` library.

The main issue I found myself faced with was not being able to declare methods outside of the main method. In C++, you need to declare help methods before your main method so that the compiler doesn't get confused. Unfortunately, most of the methods I was making had to declare and manipulate rectangle objects, the issue is that I could not do anything with rectangles without first calling `initCanvas()`, which must be called in the main method. This means any outside method I wanted to create could not deal with any graphical objects that `initCanvas()` is required for, which was the majority of the methods I wanted to make.

I dealt with this issue by essentially hard-coding everything inside the main method, after `initCanvas()` had been called. This mostly affected how my Cars would work and the overall "tidiness" of my code, but it still functions as I first planned it to.

Another smaller issue was the timing of my lights. I was able to make help methods outside of my main method that altered my traffic lights which was very helpful, but with my Cars not being declared outside of the main method, it was hard to move the cars and have the traffic lights adjust accordingly. Since I moved the Cars within my main method using a for loop, I couldn't adjust the lights while my cars moved. How I resolved this issue was by adding a boolean value that would turn to true once a car reached a certain point on the screen. With this boolean value, it would call the lights method **ONLY ONCE** in the for loop (instead of every iteration of the for loop). This worked however the light only turns red once the cars are fully off the screen. I think it looks great though.

## How it works

The program is 274 lines long, the majority of which is my main method, outside of the main method I have 3 methods that control the colour of my traffic lights.

The first thing my program does is create an intersection using 2 grey rectangles, secondly, both of my traffic lights are created, which use multiple circle shapes of various colours. I also defined a struct called 'Car' which is just a rectangle object, it isn't necessary but it was nice declaring arrays of Cars rather than random rectangles with no name. Then I instantiate all of the cars within an array, place them accordingly on the screen, and hide them until the traffic density is selected.

At this point the functionality of my code comes from a switch statement, that is in a while loop for repeated use. The switch statement has cases for heavy, medium and light traffic, and a case for exiting the program. After user input is acquired, the corresponding traffic is simulated by showing either 1, 2 or 3 Car objects then moving them across the screen. Calls to the light methods are made after the last car moves across a certain point in the road to ensure the lights are fair and efficient. Once all cars have safely moved the Cars restart and then user is asked for input again.

## Conclusion

Overall I am very happy with my code performance and how my graphics look. Simplecpp limited me to making some coding decisions I would not normally make, but I made do and made a program that works and that I am proud of. Through making this project I became very comfortable with various C++ and simplecpp concepts, as well as becoming a stronger programmer by making a fully functioning project start to finish 100% independently.