Bryce Remick
CSCI 144
Fall 2019
Stop Light Project Report

This project was a successful attempt to simulate a stop light using multithreading in C++ (version 11). To accomplish this, C++ 11 threads were used to represent the cars driving through the intersection, four priority queues were utilized to store the cars corresponding to the direction of the car(north, south, east, and west), and one queue was used to represent the intersection itself. Given a file containing cars with their corresponding time and direction, the program first starts by parsing this file. While reading each car, a thread is created with a callback function that pushes that car into it's corresponding direction priority queue. After the car is in it's queue, since all cars are initialized into a "non-ready" state, the system forces the thread to wait.

While the car parsing is running, another thread called "central processing" has been started. This thread is the program's main thread, and is essentially representing the stop light at the intersection. It is in charge of signaling cars and ensuring that no collisions occur. It accomplishes this by continually iterating across the top cars of each direction priority queue, comparing each top car to all the cars *currently* in the intersection queue. If there are no collisions, this thread then signals to the cars that are ready to enter the intersection. This algorithm continues until there are no cars remaining in any of the directional priority queues.

The car collision checking is done using vectors containing integers which represent the section of the intersection a car drives through. The intersection is broken up into four quadrants, 0 through 3 (see figure 1). For example, if car_1 is heading north, it's intersection coverage would be $\{3, 1\}$, car_2 heading south turning east would have the coverage of $\{0, 3\}$, and car_3 heading east turning south would simply have the coverage of $\{2\}$. In this model,

any two cars who's coverage *overlaps* would collide. In this scenario, car_1 and car_2 would collide because they both have a 3 in their coverage vectors. The edge cases in this model are the diagonal turns, which are $\{0, 3\}$ and $\{2, 1\}$. If two cars have the same diagonal coverage vector, they are allowed to both go.

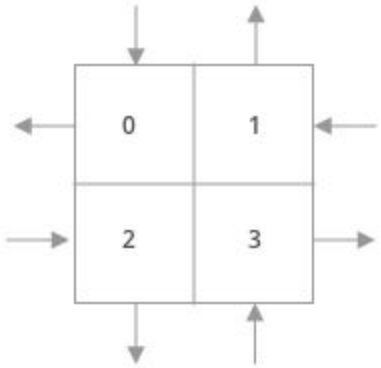

Figure 1