UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Documentación Tarea 4: Aplicación Gráfica para Semáforos en Intersección en C++ y QT

ELO329 Diseño y Programación Orientada a Objetos

Profesor: Cristobal Nettle

Nombres alumnos: Paula Amigo (201504013-3)
Luis Bahamondes (201421077-9)
Jairo González (201304502-2)

Fecha de entrega: 6 de septiembre de 2019

Índice

1. Descripción del Desafío

En esta tarea se pide modelar semáforos en una intersección de calles. En particular analizaremos el funcionamiento de los semáforos ubicados en la intersección de Avenida Sporting Club con 1 Norte en Viña del Mar. Se utilizará lo diseñado en la Tarea 3, de forma que se busca adaptar este funcionamiento, a una aplicación gráfica en el entorno QT.

2. Objetivos

* Manejar proyectos vía GIT.
* Crear Interfaces gráficas en C++.
* Manejar hilos y Timer.
* Familiarizarse con el entorno de desarrollo QT
* Ejercitar la creación y extensión de clases para satisfacer nuevos requerimientos.

3. Diagrama de Alto Nivel - Stage 4

A continuación se presenta un diagrama general de alto nivel que representa las interacciones entre las diferentes clases pertenecientes a la etapa 4 (Stage4).

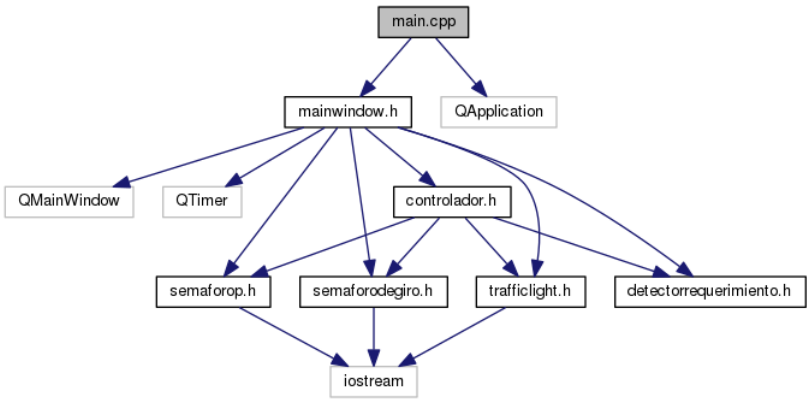

Figura 1: Diagrama de Alto Nivel Stage 4 Main

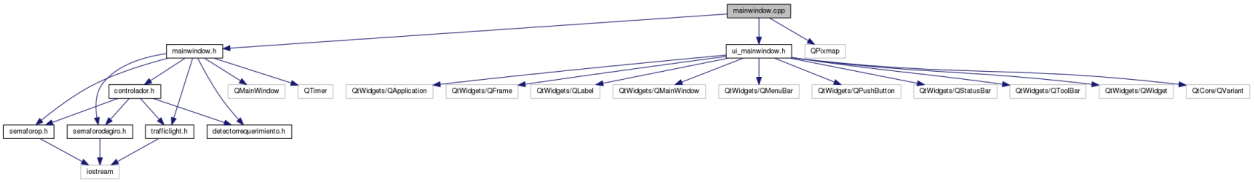

Figura 2: Diagrama de Alto Nivel Stage 4 MainWindow

4. Explicación de las clases - Stage 4

A continuación se explican las clases que componen el stage final del proyecto.

4.1. Main

Código main con el que inicia nuestro programa y llama a la clase mainwindow que nos entrega el entorno gráfico.

4.2. MainWindow

Clase dedicada a la interfaz de usuario, ventanas y las correspondientes visualizaciones de cada una de las imágenes deseadas, sean estas los semáforos peatonales, de giro u tráfico. además de presentar los métodos necesarios que son llamados por la interfaz gráfica al ser presionado un botón. Aquí se utiliza la clase Qtimer, nativa de Qt, la cual esta enfocada a realizar dos acciones, actualizaciones periódicas de los gráficos y un llamado periódico del controlador.

4.3. Controlador

La clase Controlador representa el control de todos los modelos de los distintos semáforos existentes. En esta clase se define el estado inicial de la intersección vial de las calles Sporting Club con 1 Norte y se define el método mannageTraffic(). Este funciona de manera similar que en las tareas previas, pero con diferencias en nombres de variables. Cabe destacar que se planificó el semáforo pensando en dar oportunidad a todos para cruzar. En el caso en que se presione el botón de cruce antes de darle el paso a los autos de 1 Norte, se les dará un segundo de verde y luego amarilla para que alcancen a pasar los primeros, para luego darle el cruce al peatón.

4.4. DetectorDeRequerimiento

Basado en las tareas previas, la clase DetectorRequerimiento conserva su estructura. Esta clase permite configurar como encendido o apagado algún requerimiento: solicitud de cruce peatonal o solicitud de giro vehicular. Representa el modelo para los sensores y botones.

4.5. SemaforoDeGiro

Clase dedicada al modelo del semáforo de giro, se almacena su estado y los métodos para su funcionamiento.

4.6. SemaforoP

Clase dedicada al modelo del semáforo peatonal, se almacena su estado y los métodos para su funcionamiento.

4.7. TrafficLight

Clase para los semáforos de tráfico. Ésta incluye los métodos para cambiar de estado entre FOLLOW, TRANSITION y STOP. Aquí se definen los parámetros que contienen los tiempos en luz verde y amarillo. Es posible leer los tiempos de funcionamiento y el estado actual mediante los métodos definidos.

5. Dificultades y Soluciones

Dentro de las dificultades existentes, estuvo plantear la lógica detrás del controlador, corrigiendo errores que se arrastraron desde la tarea 3 y lograr migrar correctamente al entorno gráfico considerando una nueva clase main. Crear un funcionamiento lo mas cercano a la realidad y que no presentara "bugs' o fallas en su procedimiento.

Otra dificultad fue la implementación gráfica, en específico, poder realizar cambios de imágenes dentro del mismo código a un cuadro de texto, o a un botón.

Finalmente lograr un uso adecuado en la implementaciones de signal y slots en concordancia con la clase Qtimer ya nombrada.