Diseño y Programación Orientados a Objetos
1er. Semestre 2025

Tarea 3: Simulador Gráfico en C++ del Patrón de Diseño Publicador Suscriptor

Lea detenidamente la tarea. **Si algo no lo entiende, consulte.** Si es preciso, se incorporarán aclaraciones al final.

1. Objetivos de la tarea

- Modelar objetos reales como objetos de software.
- Ejercitar la creación y extensión de clases dadas para satisfacer nuevos requerimientos.
- Reconocer clases y relaciones entre ellas en códigos fuentes C++.
- Ejercitar la compilación y ejecución de programas C++ desde una consola de comandos.
- Ejercitar la configuración de un ambiente de trabajo para desarrollar aplicaciones en C++ y Qt.
- Manejar proyectos de software vía Git.
- Ejercitar la preparación y entrega de resultados de software (readme, documentación).
- Familiarización con desarrollos "iterativos" e "incrementales".
- Desarrollar software con interfaz gráfica utilizando la biblioteca Qt.

2. Descripción General

Esta tarea busca programar, usando C++ y Qt, las funcionalidades de la tarea 2 ya desarrollada en Java y JavaFX para simular gráficamente el patrón de diseño publicador y suscriptor.
Este simulador permite crear dos tipos de publicadores: un publicador de URLs y otro de posiciones de un automóvil. El URL ingresado representa la ubicación en Internet de un video. Los dos tipos de publicadores solicitan ingresar un nombre del publicador y el nombre del tópico. Para esto se sugiere usar el método estático QInputDialog::getText(…).

2.1. Publicador y suscriptor de videos.

Este tipo de publicador muestra un campo de texto donde el usuario ingresa el URL del video, usted puede considerar instancias de QLabel y QLineEdit. Los suscriptores al tópico de este publicador muestran un botón (QPushButton) cuyo texto corresponde al último URL publicado. Al presionar el botón, el programa permite al suscriptor ver el video publicado. Un ejemplo para reproducir video está aquí, el cual muestra una interfaz gráfica como la Figura 1.

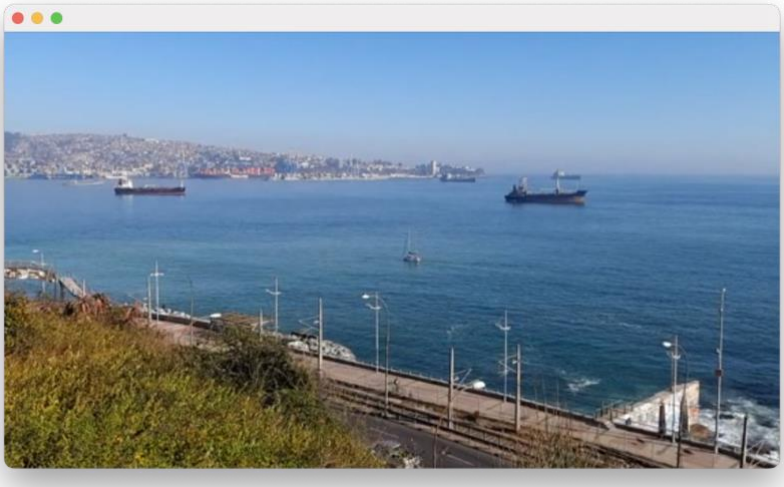

Figura 1: Ejemplo de reproductor del video al presionar el botón

2.2. Publicador y suscriptor de la posición de un móvil.

Al crear este tipo de publicador, luego de especificar el nombre del publicador y su tópico, usando la clase QFileDialog (usado en ejemplo ya mencionado), el programa pide seleccionar el archivo

que contiene las posiciones relevantes a simular. El archivo de posiciones tiene el siguiente formato por cada línea de texto:
<tiempo> <posición x> <posición y>
El tiempo es un **número entero** de segundos, aun cuando pueda ingresarse y leerse como número real.
El GPS del móvil lee este archivo e interpola las posiciones intermedias para enviar una posición cada segundo (Revise este ejemplo de uso de QTimer). El mensaje enviado por el publicador contiene la información de posición en cada segundo a partir de las líneas del archivo.
Los suscriptores de posición muestran un círculo de color, cuya posición se actualiza cada vez que una nueva posición es publicada. Cada suscriptor crea una ventana propia para mostrar el avance del móvil de su tópico. En la parte inferior de la ventana se muestra el tiempo y las coordenadas x e y del móvil asociado a ese tópico, ver Figura 2. Este ejemplo debería ser útil para que el suscriptor muestre la posición del auto conforme llegan actualizaciones de su posición.

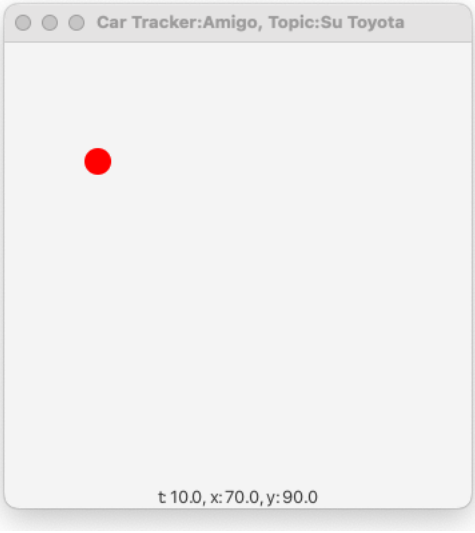

Figura 2: Ejemplo de vista para el seguidor de posición del móvil

3. Interfaz Gráfica

La interfaz gráfica de este simulador considera sólo dos publicadores los cuales ubica a el sector izquierdo de la interfaz. El primer publicador lo hace al tópico “Video” y el segundo lo hace al tópico “GPS”. Al momento de crear el publicador de posición GPS, la aplicación solicita al usuario seleccionar el archivo de texto con las posiciones.
Luego este simulador crea dos suscriptores en el lado derecho de la interfaz. El primero muestra un botón cuyo texto cambia con la publicación de cada URL nuevo. El segundo suscriptor muestra la posición del móvil en el tiempo (Figura 2) según nuevas posiciones son publicadas.
Al presionar el botón del suscriptor de video, el usuario puede ver el video correspondiente.

En esta oportunidad se espera que su programa se ejecute desde QtCreator.

4. Desarrollo en Etapas

Para llegar al resultado final de esta tarea usted debe aplicar una metodología de desarrollo de software "Iterativo y creciente (o incremental)". Usted y su equipo irán desarrollando etapas donde los requerimientos del sistema final son abordados gradualmente. En cada etapa usted y su equipo obtendrá una solución que funciona para un subconjunto de los requerimientos finales o bien es un avance hacia ellos. El desarrollo en etapas en esta tarea tiene por objetivo familiarizarse con la metodología de desarrollo iterativo e incremental.

4.1. Primera Etapa: Publicador y suscriptor de videos

En esta primera etapa se espera que usted cree versiones de las clases Topic, Broker, Component, Publisher y Subscriber de la Tarea2. Adicionalmente, cree las clases VideoPublisher y VideoFollower, las cuales extienden las clases Publisher y Subscriber respectivamente. Al presionar enter en el campo de texto, el URL es publicado y el campo de texto es borrado. VideoFollower actualiza su vista. El texto del botón se actualiza ante una nueva publicación de URL. En esta etapa el botón no responde al ser presionado. El programa Stage1.cpp crea la interfaz gráfica de la aplicación la cual contendrá sólo el publicador de URLs de video y su suscriptor único.
Entregue las clases de esta etapa.

4.2. Segunda Etapa: Se agrega reproductor de video a etapa previa

Complete el VideoFollower para reaccionar a su botón. Al presionarlo se inicia la reproducción del video como se muestra en la Figura 1.
Para probar, usted puede usar los siguientes videos:
http://profesores.elo.utfsm.cl/~agv/elo329/1s22/Assignments/20220430_100849.mp4
http://profesores.elo.utfsm.cl/~agv/elo329/1s22/Assignments/20220430_101027.mp4

4.3. Tercera Etapa: Publicador de GPS de posiciones de un móvil

Cree las clases GPSCarPublisher la cual extiende Publisher e incorpore un único publicador de posición GPS de un único móvil. Para este publicador, use una instancia de QFileDialog para seleccionar el archivo con las posiciones a simular en el GPS. La clase GPSCarPublisher reportar la posición cada 1 segundo aun cuando en el archivo de entrada pueda venir a intervalos mayores a 1 segundo. Revise el ejemplo de uso de QTimer. Se espera que usted interpole linealmente (con movimiento de velocidad constante) las posiciones intermedias entre dos instantes de tiempo dados en el archivo.
En esta etapa el suscriptor sólo mostrará y actualizará, en una ventana separada, el tiempo y la posición (x, y) recibida. Como en Figura 2, pero sin el círculo en movimiento.
Entregue todas las clases de esta etapa y el archivo de entrada usado.

4.4. Cuarta Etapa: Cumplimiento de las funcionalidades del simulador según secciones 2 y 3

En esta se da cumplimiento a la totalidad de la especificación de la tarea.

4.5. Extra-crédito: Use un QSlider para ajustar volumen de reproducción.

(8 puntos, la nota se satura en 100) Incorpore una instancia de QSlider (Slider horizontal) para ajustar el volumen de reproducción del video.
Si desarrolla esta parte, lo debe indicar en su readme.

5. Elementos a considerar en su entrega

- En esta tarea su equipo deberá entregar sólo la solución de la última que logre desarrollar (sólo una etapa, la más avanzada).
- Para la entrega, deberá subir a Aula un enlace a su repositorio Github/Gitlab, dando acceso a los ayudantes para su revisión. La sección de entrega de la tarea en AULA será exclusivamente para agregar el link al repositorio Git donde esté alojada la solución de la tarea.
- Entregue todo lo indicado en "Normas de Entrega de Tareas", excepto makefile. Toda la entrega es obligatoria por github/gitlab.
- En su archivo de documentación (pdf o html) incorpore el diagrama de clases de la aplicación (última etapa desarrollada)