
Práctica 5: Una interfaz gráfica para el simulador de tráfico

Fecha de entrega: el 30 de abril de 2018 a las 23:00

Objetivo: Patrón Modelo-Vista-Controlador, Interfaces Gráficas de Usuario con Swing.

Introducción

En esta práctica, crearemos una interfaz gráfica de usuario para el simulador de tráfico. Para ello, primero modificaremos el simulador para que se ajuste mejor al patrón MVC, y luego añadiremos una interfaz con Swing que permita interactuar con el simulador.

IMPORTANTE: algunas partes son opcionales, y están siempre marcadas con la palabra **OPCIONAL**.

Simulador MVC

En esta primera parte, modificaremos el simulador para que soporte observadores, que podrán registrarse para ser notificados cuando haya cambios en el simulador (también llamados *eventos*, aunque no tengan nada que ver con los de la anterior práctica).

El simulador debe poder notificar, como mínimo, cuando ocurran los siguientes eventos:

- REGISTERED. Cuando un observador se registre, ese observador, y sólo ese, deberá recibir esta notificación.
- RESET. Notificado cuando el simulador se reinicie.
- **NEW_EVENT. Notificado cuando se añadan eventos al simulador.**
- ADVANCED. Notificado cuando se avance el tiempo de la simulación.
- ERROR. Cuando se produzca un error durante la simulación. Contendrá, adicionalmente, una descripción del error.

Todos los eventos notificados por el modelo incluirán el estado completo de la simulación en el momento de lanzarse el evento, y serán, con la excepción de REGISTERED, notificados a todos los observadores registrados.

Interfaz gráfica de usuario

En esta segunda parte, construiremos una interfaz Swing para el simulador. La interfaz debe ser similar, pero no necesariamente idéntica, a la que se muestra en la Figura 1. A continuación se describen los componentes de la interfaz.

Editor de eventos

Este componente permite a los usuarios editar eventos. Contiene un editor de texto, basado por ejemplo en `JTextArea`, y debe suministrar la siguiente funcionalidad:

- *cargar* un fichero de texto en el editor.
- *guardar* su contenido en un fichero de texto.
- *borrar* el contenido, dejando un texto vacío.

Como parte del enunciado, recibirás un editor de texto Swing que podrás adaptar para su uso en la realización de este editor de eventos.

Cola de eventos

Este componente contendrá una lista de los eventos pendientes en la cola del simulador. Cada elemento representará un evento, mediante su tiempo de ejecución y una descripción textual. Puedes usar una `JTable` ó una `JList` para implementar el componente.

Zona de informes

Este componente consta, esencialmente, de un área de texto donde se escriben los informes. Estos informes se pueden generar de las siguientes formas:

- Todos los informes. La interfaz debe permitir generar todos los informes del estado actual del simulador, que se mostrarán a continuación en esta zona. Por ejemplo, puedes hacer que se muestren todos tras pulsar sobre un botón.
- Informes concretos (**OPCIONAL**). La interfaz permitirá seleccionar vehículos, carreteras e intersecciones y generar informes de los así seleccionados. La selección se podrá hacer mediante (1) un diálogo similar al de la Figura 3; (2) una selección en las correspondientes tablas, usando `getSelectionModel()` sobre la `JTable` correspondiente; ó (3) cualquier otro método que tenga sentido y funcione. Junto con este enunciado, recibirás un ejemplo de cómo crear un diálogo. **Si implementas esta opción, no es necesario implementar la anterior.**
- Generados por el simulador (**OPCIONAL**). Redirige salida del simulador (es decir, lo que escribe en su `OutputStream`) a la zona de texto. Para ello, deberás implementar una `OutputStream` que implemente un `write` que resulte en escribir en la `JTextArea` correspondiente; y se lo deberás pasar al simulador para

que lo use de forma normal. El usuario también deberá poder deshabilitar esta redirección (según se ve en el menú de la Figure. 2) – esto se puede hacer poniendo el `OutputStream` de salida del simulador a `null`. **Si no vas a implementar esta parte, recomendamos poner la salida del simulador a `null` desde un principio.**

La interfaz también debe permitir:

- Mostrar el contenido actual de los informes a un fichero de texto (**OPCIONAL**);
- Borrar el contenido de la zona de informes.

Tablas con objetos del simulador

La interfaz debe contener una tabla por categoría de objeto simulado, es decir, para vehículos, carreteras e intersecciones. Estas tablas se deben mostrar usando `JTable` con sus correspondientes modelos (`TableModel`). La información a incluir es similar a la que se muestra en los informes. Tienes libertad para mostrar información adicional, y para mostrarla de cualquier forma que sea fácilmente inteligible.

Mapa de carreteras

Un componente Swing que muestra el mapa de carreteras. Proporcionaremos un componente similar que muestra un grafo (con puntos en las aristas, que podrías usar para mostrar coches). Todo lo que necesitas es reemplazar el `Graph` por un `RoadMap`.

Barra de estado

La interfaz deberá incluir una barra de estado o equivalente (en el sentido de “zona de información contextual”), donde se mostrarán mensajes informativos mientras se interacciona con la interfaz. En la Figura 1, esta barra está en la zona inferior.

Interactuando con el simulador

La interfaz deberá proporcionar las siguientes opciones:

- Insertar eventos. Añadirá los eventos que contenga el editor de eventos al simulador. Ten en cuenta que puedes crear una `InputStream` a partir de una cadena `s` mediante `new ByteArrayInputStream(s.getBytes())`.
- Lanzar/lanza el simulador durante tantos ciclos de tiempo como haya especificado el usuario. Por ejemplo, puedes usar un `JSpinner` similar al de la Figura 1)
- Reiniciar el simulador.

Barras de herramientas y de menú

Tanto la barra de herramientas como la de menú permiten hacer esencialmente lo mismo, y deberás implementar, al menos, una de las dos, resultando (**OPCIONAL**) implementar la otra.

A continuación explicaremos los elementos de la barra de herramientas de izquierda a derecha: cargar un fichero de eventos, salvar un fichero de eventos, limpiar la zona

de eventos, insertar eventos en el simulador, lanzar el simulador, reiniciarlo, establecer cuántos ciclos se debe simular cuando se lance, una etiqueta que indica el ciclo actual de la simulación, generar informes, limpiar la zona de informes, limpiar los informes, y finalmente salir.

Plantillas para eventos (OPCIONAL)

Para facilitar la creación de eventos en el editor, añade un menú contextual (activado vía click-derecho en el editor) que permita elegir entre varias plantillas de evento. Si se selecciona una plantilla, deberás insertar el texto correspondiente en el punto de inserción actual del editor (ver Figure 4). Te proporcionaremos un ejemplo de menú contextual similar al que te pedimos implementar.

La clase Main

Deberás modificar la clase principal de tu aplicación para que soporte tanto el uso de interfaz gráfica (esta práctica) o el modo de texto sin interacción de la práctica anterior. Para ello, deberás usar un argumento adicional `-m` que podrá tomar los valores `'batch'` o `'gui'`, y que lanzará la aplicación en el modo correspondiente. Si este argumento no se especifica, deberá tomar por defecto el valor `'batch'`.

El resultado de ejecutar el programa con el argumento `-h` (que muestra la ayuda) deberá ser similar al siguiente:

```
> java Main -h
usage: Main [-h] [-i <arg>] [-m <arg>] [-o <arg>] [-t <arg>]
  -h,--help                Print this message
  -i,--input <arg>         Events input file
  -m,--mode <arg>          'batch' for batch mode and 'gui' for GUI mode
                           (default value is 'batch')
  -o,--output <arg>        Output file, where reports are written.
  -t,--time <arg>          Time units to execute the simulator's main loop
                           (default value is 10).
```

En modo no-interactivo (`batch`), todo el comportamiento debe ser idéntico al de la práctica anterior. En el modo gráfico (`gui`) mode, deberás tener en cuenta lo siguiente:

- `-i`: si se proporciona un fichero de eventos iniciales usando esta opción, debe cargarse inmediatamente en la zona de eventos nada más mostrarse la interfaz.
- `-o` y `-t`: ambas deben ser ignoradas, aunque podrás, a tu elección, usar el valor de `-t` como valor inicial para el control en el que se configura el número de ciclos de simulación a ejecutar cada vez que se lanza la misma.

Comentarios y resquisitos adicionales

- **Está estrictamente prohibido incluir código generado mediante generadores de código.**
- Proporcionamos código de ejemplo para: (1) el editor de texto; (2) la ventana de diálogo; (3) el mapa de carreteras; y (4) el menú contextual.

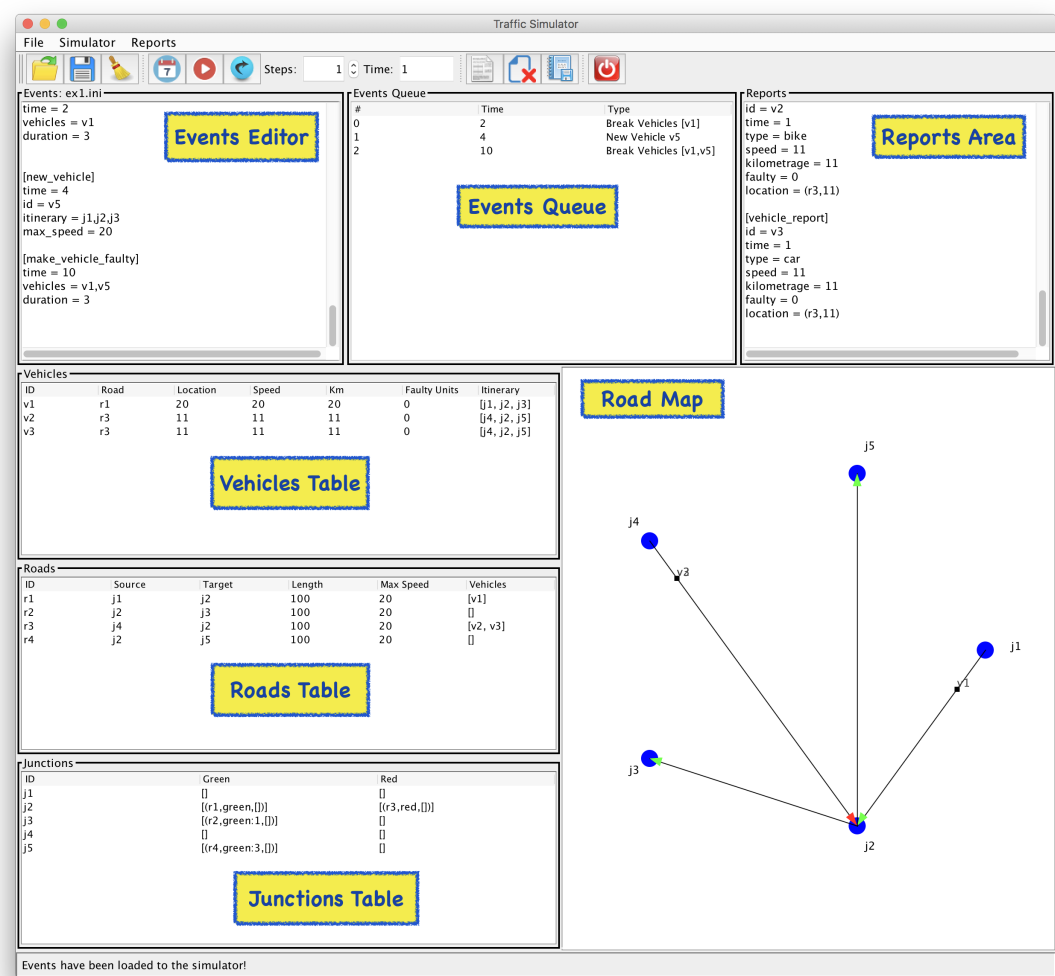


Figura 1: La interfaz. Los cuadrados amarillos son anotaciones, y no forman parte de la interfaz.

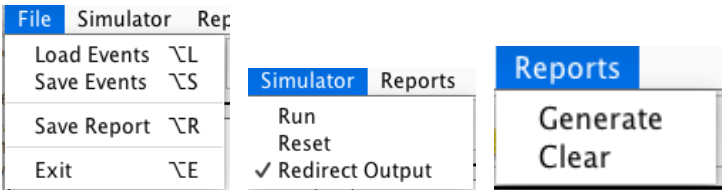


Figura 2: Los distintos menús en la barra de menús.

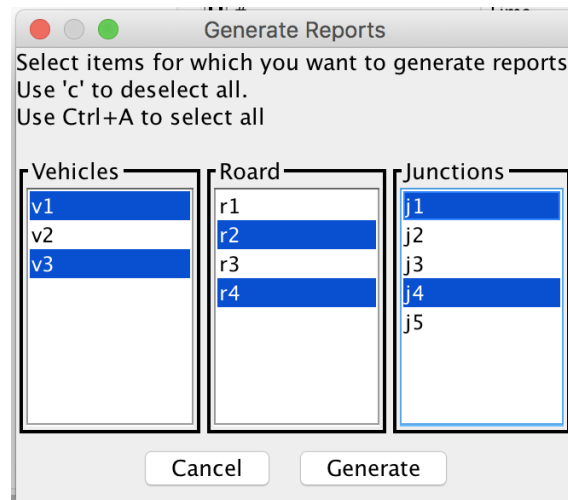


Figura 3: Un diálogo para elegir objetos simulados.

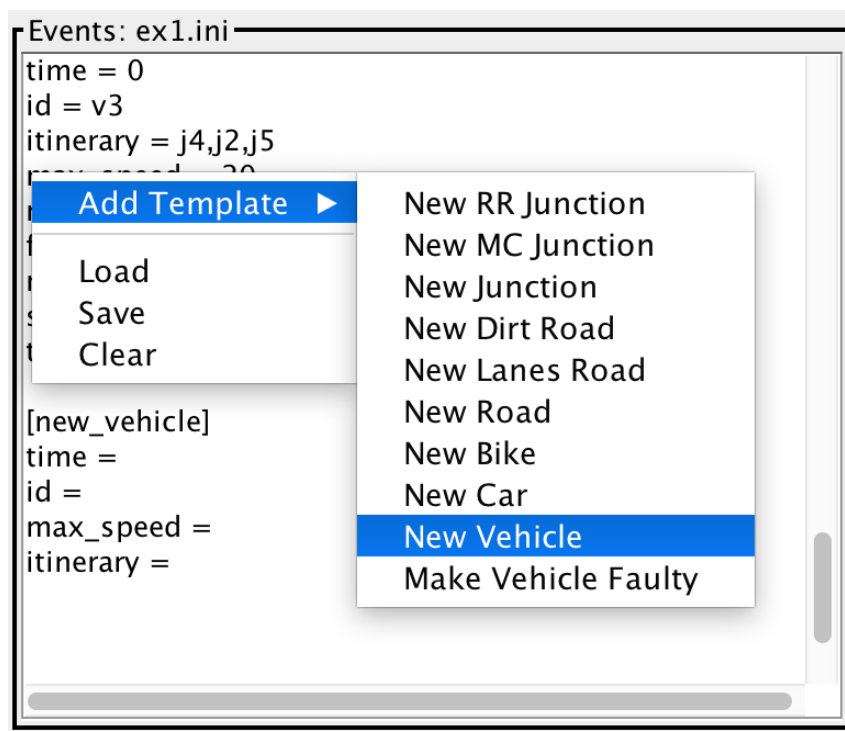


Figura 4: Un menú desplegable para añadir plantillas de eventos.