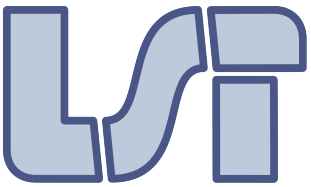


UNIVERSIDAD DE GRANADA

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS

Escuela Técnica Superior de Ingenierías Informática y Telecomunicaciones

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

Traffic Flow Simulator

Juan Antonio Fajardo Serrano

Director del proyecto: Domingo Martín Perandrés

Curso 2014/2015

Convocatoria: Julio

Índice

Capítulo 1

Introducción

La presente memoria documenta el desarrollo de un simulador de tráfico, denominado *Traffic Flow Simulator*, que tendrá como base una interfaz gráfica desde la que el usuario podrá introducir cambios en las variables involucradas en la simulación y ver los resultados que ocasionan.

La simulación está definida como una representación de cierta parte del mundo real conseguida mediante la construcción de un modelo de ordenador que va evolucionando a lo largo del tiempo [1]. Según *Simulation of traffic systems* [2], el desarrollo de simuladores de tráfico mediante ordenador comenzó en 1955 con la tesis de Daniel L. Gerlough en la Universidad de California. En la actualidad, gracias al poder de cálculo de los computadores, se han logrado grandes avances en este campo, como algunos simuladores a escala macroscópica.

Este proyecto nace del problema actual de tráfico en las grandes ciudades y la necesidad de mejorar el flujo de vehículos en puntos concretos de las zonas urbanas, que es actualmente un problema diario.

Traffic Flow Simulator tiene como objetivo principal ser una herramienta útil que facilite la visualización de la simulación de tráfico en base a un modelo lógico, de forma que permita evaluar de manera clara las diferentes alternativas a situaciones concretas, permitiendo al usuario recrear dichas situaciones mediante la interacción con la herramienta teniendo en cuenta puntos importantes como son la congestión de las vías o los comportamientos de conducción, de forma que se pueda mejorar el flujo de tráfico en las vías urbanas.

El flujo de tráfico de vehículos es un sistema complejo y difícil de modelar y medir, ya que en este se ven involucradas multitud de variables dependientes unas de otras, entre las que destacan los problemas de movilidad, los cambios de sentido, las rotondas, los tiempos de semaforización, las obras en la calzada, los accidentes de tráfico y otras variables sin contar con el factor humano y las condiciones atmosféricas.

Este simulador posibilitará realizar simulaciones de tráfico sobre distintas distribuciones de redes viarias, teniendo en cuenta los diferentes perfiles de conductores que podemos encontrarnos día a día en la carretera. Así mismo, el usuario podrá personalizar las distintas variables que influirán en la simulación de forma que pueda recrear los escenarios que desee.

La parte visual estará compuesta por un entorno pasivo que no afectará al tráfico y otro activo que afectará directamente al tráfico como son semáforos y vehículos.

Capítulo 2

Objetivos

En este capítulo se mostrarán los objetivos que han sido marcados para la realización de este proyecto.

2.1 Objetivo principal del proyecto

El objetivo principal de este proyecto es la obtención de una aplicación multiplataforma de código abierto que permita observar, de forma gráfica en tres dimensiones, el flujo de tráfico de vehículos en distintas configuraciones de red viaria, que el usuario podrá cambiar mediante la interacción con la aplicación.

2.2 Red viaria

La red viaria con la que trabajará el simulador será una red sencilla, en la cual, habrá puntos por los que entrarán y saldrán los vehículos hacia y desde la porción de red viaria que se esté simulando, habrá intersecciones de mínimo tres vías y máximo cuatro vías y algún mecanismo que nos permita definir curvas sin realizar intersecciones.

2.3 Señales de tráfico

Como señales para controlar el flujo de tráfico, la red viaria contará con semáforos en las intersecciones que se encargarán de permitir el paso alternativo de vehículos entre las distintas vías que lleguen a cada una de las intersecciones.

2.4 Vehículos

Los vehículos que intervendrán en la simulación serán de al menos tres tipos, coches, autobuses, y camiones no articulados.

Los modelos de estos vehículos serán de poco nivel de detalle con el fin de poder incluir en la simulación tantos como se pueda sin que se vea afectado el rendimiento del sistema.

El usuario podrá variar el número de vehículos involucrados en la simulación así como la proporción del tipo de vehículos (públicos o privados) mediante la interacción con la interfaz gráfica.

2.5 Conductores

Para simular el flujo de tráfico de una forma más realista, el sistema conductista contará con al menos tres tipos de conductores, aquellos que cumplen las normas de circulación siempre, aquellos que las cumplen normalmente y por último, aquellos que casi nunca las cumplen, en menor proporción respecto a los otros dos tipos.

El usuario podrá cambiar la proporción de los tres tipos de conductores mediante la interacción con la interfaz gráfica.

2.6 Entorno

El entorno estará formado por los elementos que conforman las vías de la red (plataforma de la vía, líneas de arcenes, líneas de mediana, líneas de carril taxi bus, líneas discontinuas para los carriles normales, y líneas de detención), señales horizontales para indicar carriles normales y carriles taxi bus, túneles para indicar los puntos por los que los vehículos entrarán y saldrán de la porción de red viaria simulada, y semáforos de tres estados para la regulación del tráfico.

Capítulo 3

Estado del arte

En el presente capítulo se abordará el estado del arte de los distintos elementos de los que se hará uso en el desarrollo de este proyecto.

En el apartado *Simuladores de tráfico* se presentará la esencia del proyecto; en el apartado *Representación de redes viarias*, se tratará una manera de trabajar con los mapas de redes viarias de forma que podamos generar el entorno en el que se desenvolverán los vehículos en nuestro simulador; en el apartado *Sistemas de partículas conductistas* se presentará el modelo de interacción utilizado por los vehículos mencionados anteriormente; finalmente, en el apartado *Herramientas gráficas* veremos las herramientas que se vienen utilizando para la creación de simuladores de tráfico.

3.1 Simuladores de tráfico

En general, la simulación está definida como una representación de cierta parte del mundo real conseguida mediante la construcción de un modelo de ordenador que va evolucionando a lo largo del tiempo [1].

Un simulador de tráfico es un modelo de ordenador de alguna parte de los sistemas de transporte del mundo real, cuya principal finalidad es la de ayudar a planear, diseñar y operar mejor dichos sistemas de transporte.

Se estima que la simulación nació en 1777 con el planteamiento del problema conocido como *La aguja de Buffon*. Este es un problema clásico de probabilidad geométrica, de realización práctica y cuyo interés radica en que es un método difícil para ir aproximando el valor del número π a partir de sucesivos intentos. Fue planteado por el naturalista francés Buffon en 1733 y reproducido por él mismo ya resuelto en 1757. Se trata de lanzar una aguja sobre un papel en el que se han trazado rectas paralelas distanciadas entre sí de manera uniforme. Se puede demostrar que si la distancia entre

las rectas es igual a la longitud de la aguja, la probabilidad de que la aguja cruce alguna de las líneas es $2/\pi$.

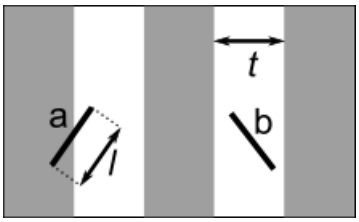


Figura 3.1: La aguja de Buffon

Los sistemas de simulación de transporte hicieron su aparición hace cuarenta años y la simulación por ordenador comenzó cuando D.L. Gerlough publicó su discurso: *Simulation of freeway traffic on a general-purpose discrete variable computer* (Simulación del tráfico de la autopista en un ordenador de propósito general de variables discretas.) en la Universidad de California, Los Ángeles, en 1955 [3].

Desde entonces, la simulación por ordenador se ha convertido en una herramienta ampliamente utilizada en la ingeniería del transporte con muchas aplicaciones que van desde la investigación científica hasta la planificación y el entrenamiento.

Actualmente, las líneas de investigación que se están siguiendo en este campo son mayoritariamente simulaciones microscópicas aunque también hay novedades muy interesantes en los modelos teóricos macroscópicos.

Como ejemplo de simulador de tráfico podemos nombrar *CORSIM*, el cual permite representar en dos dimensiones una red viaria y simular el flujo del tráfico a lo largo del tiempo indicando una serie de parámetros.

En el proyecto que nos ocupa, se va a elaborar un simulador de tráfico microscópico en tres dimensiones que permitirá analizar distintos tipos de configuraciones en una porción limitada de una red viaria.

3.2 Representación de redes viarias

Como se explica en el libro *Fundamentals of Traffic Simulation* [4], la representación de mapas o de las redes viarias dependerá del tipo de análisis que se quiera llevar a cabo sobre el sistema de transporte.

Inicialmente, la estructura más sencilla para representar lógicamente un mapa será un grafo dirigido, cuyos nodos serán las intersecciones y cuyos arcos serán las secciones viarias que conectan dichas intersecciones, indicando la dirección de las mismas.

Para completar este modelo, se caracterizan los arcos del grafo con atributos como, por ejemplo, *capacidad del enlace, número de carriles, modos de transporte que pueden utilizar cada carril (bus, coche, taxi, etc), densidad del tráfico, etc.*

Este modelo sencillo puede ser detallado añadiendo nodos para indicar los posibles giros permitidos en una intersección, tal y como se muestra en la figura 3.2.

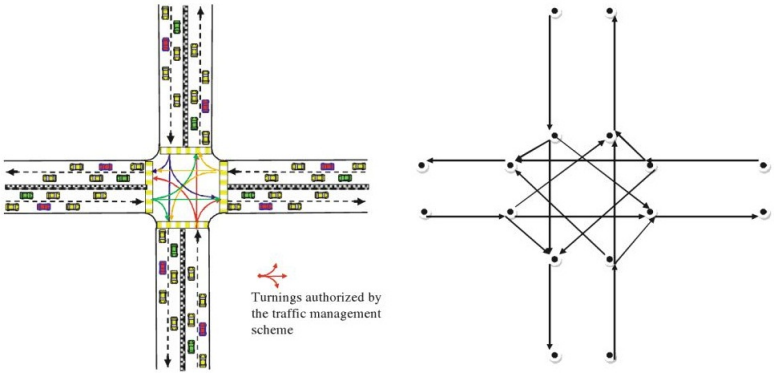


Figura 3.2: Representación de una intersección (Barceló 2010) [4]

Una vez que tenemos un modelo sencillo, al cual se le pueden añadir tantos detalles como deseemos en forma de atributos en los nodos y en los arcos, se nos presenta el problema de la búsqueda de caminos en una red viaria.

Como solución a dicho problema aparece el estudio *A new data structure to represent road networks* [5], en el cual se propone utilizar tres niveles de detalle para representar el mapa. Estos niveles pueden observarse en la figura 3.3 y en la tabla 3.1

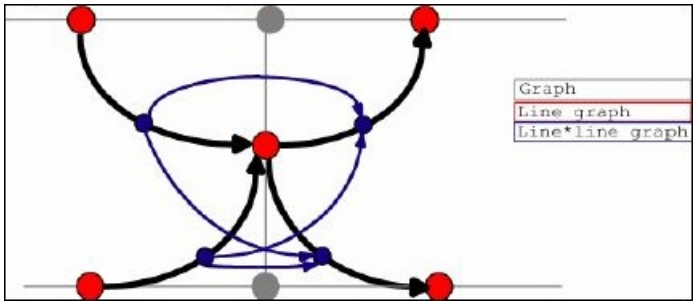


Figura 3.3: Representación de una intersección (Bogaert) [5]

Nivel	Nodos	Arcos
0 (un grafo)	Intersecciones	Segmentos de vía
1 (grafo de líneas)	Segmentos de vía	Giros
2 (grafo de líneas de líneas)	Giros	Conexiones entre giros

Tabla 3.1: Niveles representados en la figura 3.3

Dado que en los objetivos de este proyecto no se encuentra el que los vehículos elijan y sigan un camino concreto a través de la porción de red viaria simulada, no profundizaremos más en el problema de la búsqueda de caminos en una red viaria.

3.3 Sistemas de partículas conductistas

Los sistemas de partículas son colecciones de objetos independientes, a menudo representados por una sola forma o punto. Todas las partículas tienen una serie de atributos los cuales permiten tener distintos tipos de partícula, tanto en aspecto como en comportamiento.

Generalmente, la implementación de sistemas de partículas conductictas se lleva a cabo mediante la utilización de máquinas de estados finitos, las cuales sirven para modelar el comportamiento de las partículas, de manera que éstas actuarán de una forma u otra en base a sus atributos.

En el caso particular de este proyecto, cada vehículo será una partícula con una serie de características o atributos, lo que nos permitirá modelar distintos tipos de vehículo y distintos tipos de comportamiento de los conductores, con el fin de representar la variedad de los mismos que podemos observar en la realidad.

3.4 Herramientas gráficas

En el apartado gráfico vamos a contar con dos herramientas que se están utilizando mucho en estos tiempos en el área de la informática gráfica. Por un lado podremos usar Unity3D [6], que es un motor gráfico de videojuegos que cuenta con su propio editor y permite programación en C#, ya que sus características nos permiten simular un sistema de partículas conductista en un entorno 3D, así como realizar una interacción con la aplicación de forma sencilla e intuitiva. Por otro lado podremos contar con la herramienta Blender [7], que se usa en modelado, y nos permitirá incluir los modelos necesarios en la aplicación. Además, esta última se integra perfectamente con la primera.

Capítulo 4

Propuesta

En este capítulo expondremos el trabajo que se pretende realizar para conseguir cada uno de los objetivos mostrados en el capítulo anterior.

4.1 Objetivo principal del proyecto

Se desarrollará una aplicación gráfica multiplataforma, mediante la herramienta *Unity3D* utilizando el motor gráfico que incluye, para alojar el simulador de tráfico.

Debido a problemas con Unity3D en la generación de la aplicación para sistemas Linux y a la imposibilidad de realizar pruebas sobre el sistema operativo de Apple, la aplicación será finalmente construida para el sistema operativo Windows en sus versiones de 32 y 64 bits.

4.2 Red viaria

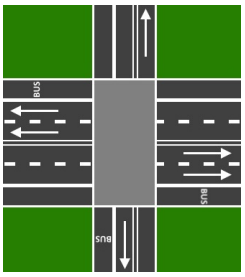
La red viaria que utilizará el simulador vendrá especificada en un fichero con formato GraphML [8]. Dicha red estará determinada por dos grafos, un grafo no dirigido, de topología, en el que los cruces, las curvas y los nodos límite (los puntos de entrada/salida de vehículos desde y hacia la porción de red viaria simulada) serán los nodos, y las vías serán los arcos; y un grafo dirigido, para indicar los giros, en el que los carrilles de las vías serán los nodos y los giros serán los arcos.

Debido a las restricciones que impone la definición de GraphML sobre los identificadores, cada grafo deberá estar en un fichero distinto. Para ello usaremos dos ficheros por cada red viaria, siendo los ficheros *.topology.graphml para el primer tipo de grafo y los ficheros *.turns.graphml para el segundo. El identificador del grafo en cada uno de los ficheros de la pareja de cada grafo será el mismo.

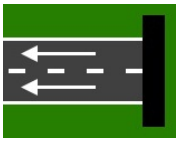
Nodos del grafo de topología

Debemos distinguir tres tipos:

- Nodos de cruce: Representan un cruce de vías típico. El grado de estos nodos es tres o cuatro.



- Nodos de límite de vía: Representan el límite del área simulada en esa vía. Será el lugar por el que los vehículos salgan y entren desde y hacia la porción de red viaria simulada. El grado de este tipo de nodo es igual a uno. En el entorno 3D tendrán apariencia de tunel.



- Nodos de continuación: Representan ángulos en las vías de tal forma que en ellos no se producirán intersecciones de vías. El grado de este tipo de nodo es igual a dos.

Por tanto, cada nodo estará definido por un identificador alfanumérico, el tipo de nodo (0: nodo de cruce, 1: nodo de límite de vía, 2: nodo de continuación), sus coordenadas en el plano bidimensional x,y, y para los nodos de tipo cruce, el tipo de cruce, cruce normal (0) o rotonda (1). Las rotondas se han quedado como punto de ampliación de la aplicación.

Arcos del grafo de topología

Cada arco estará definido por un identificador alfanumérico, el nodo de origen, el nodo de destino, una cadena de texto para poder dar nombre a la vía, y dos cadenas de texto que servirán para indicar los tipos de carril en cada sentido.

Cada una de las cadenas de tipo de carril especificará el tipo de carril o carriles de ese sentido que haya desde el exterior de la vía hacia el interior de la misma utilizando los códigos: P: para indicar un carril de transporte público, N: un carril normal, o la cadena '0' para indicar que no hay carriles en ese sentido.

Nodos del grafo de giros

Cada nodo estará definido por un identificador alfanumérico que se corresponderá con uno de los identificadores de los carriles de los arcos del grafo de topología. Los identificadores de los carriles tendrán la forma siguiente: identificador del arco al que pertenece, guión bajo, la cadena "src_des" o la cadena "des_src" en función de la dirección en la que va el carril, (desde el origen al destino o viceversa), guión bajo, número de carril desde el exterior del arco hacia el interior, siendo 0 el primer carril.

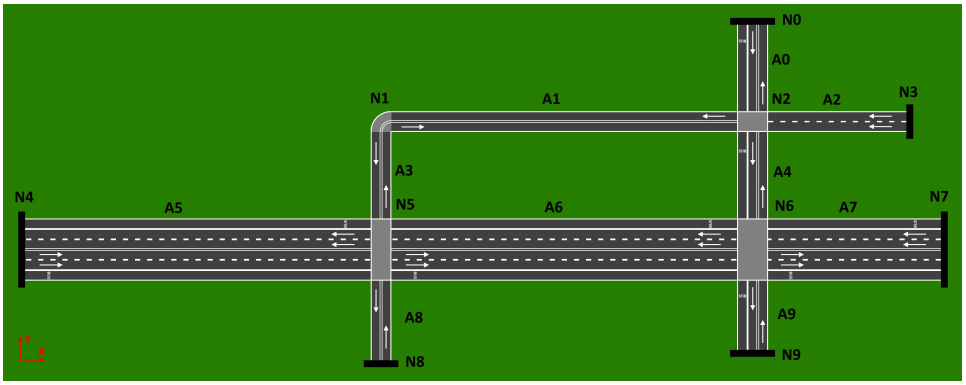
Por ejemplo, el segundo carril del arco "a2" en dirección origen destino será:

```
<node id="a2_src_des_1"/>
```

Arcos del grafo de giros

Cada arco estará definido por un identificador alfanumérico, el nodo de origen y el nodo de destino. Si el arco está definido significa que el giro está permitido. En el caso de carriles que lleguen a nodos de tipo continuación o límite, no será necesario definir los arcos.

Ejemplo Para ilustrar esta especificación vamos a ver un ejemplo de red viaria y cuál sería su representación en GraphML.



Como se puede apreciar en la imagen, el grafo de topología asociado consta de diez nodos, tres nodos de tipo cruce, un nodo de tipo continuación y seis nodos de tipo límite de vía. Además, el grafo cuenta con diez arcos que representan las vías de la imagen.

Para este ejemplo se ha supuesto que en cada cruce se puede ir en todas las direcciones que admite esta red, aunque se aprovecha la multiplicidad de carriles para hacer los giros a izquierda desde el carril o carriles situado/s más a la izquierda, y análogamente con los giros a la derecha.

El grafo de giros está compuesto por diez nodos, los arcos del grafo de topología, que en este ejemplo están todos presentes en el grafo debido a que todos están unidos a algún nodo de tipo cruce; y treinta y tres arcos que representan los giros.

A continuación se mostrará la representación de la red viaria con sintaxis de GraphML.

Grafo de topología:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
        <key id="node_type" for="node" attr.name="Node_type" attr.type="int">
                <default>1</default>
        </key>
        <key id="pos_x" for="node" attr.name="Coordinate_X" attr.type="double"/>
        <key id="pos_y" for="node" attr.name="Coordinate_Y" attr.type="double"/>
        <key id="crossing_type" for="node" attr.name="Crossing_type" attr.type="int">
                <default>0</default>
        </key>
        <key id="road_name" for="edge" attr.name="Road_name" attr.type="string">
                <default></default>
        </key>
        <key id="src_des" for="edge" attr.name="Lanes_on_source-destination_sense" attr.type="string">
                <default>PNN</default>
        </key>
        <key id="des_src" for="edge" attr.name="Lanes_on_destination-source_sense" attr.type="string">
                <default>N</default>
        </key>
        <graph id="example_1" edgedefault="undirected">
                <node id="n0">
```

```
                <data key="pos_x">2168</data>
                <data key="pos_y">1044</data>
        </node>
        <node id="n1">
                <data key="node_type">2</data>
                <data key="pos_x">1095</data>
                <data key="pos_y">751</data>
        </node>
        <node id="n2">
                <data key="node_type">0</data>
                <data key="pos_x">2168</data>
                <data key="pos_y">751</data>
        </node>
        <node id="n3">
                <data key="pos_x">2623</data>
                <data key="pos_y">751</data>
        </node>
        <node id="n4">
                <data key="pos_x">53</data>
                <data key="pos_y">381</data>
        </node>
        <node id="n5">
                <data key="node_type">0</data>
                <data key="pos_x">1095</data>
                <data key="pos_y">381</data>
        </node>
        <node id="n6">
                <data key="node_type">0</data>
                <data key="pos_x">2168</data>
                <data key="pos_y">381</data>
        </node>
        <node id="n7">
                <data key="pos_x">2723</data>
                <data key="pos_y">381</data>
        </node>
        <node id="n8">
                <data key="pos_x">1095</data>
                <data key="pos_y">7</data>
        </node>
        <node id="n9">
                <data key="pos_x">2168</data>
                <data key="pos_y">83</data>
        </node>
        <edge id="a0" source="n0" target="n2">
                <data key="src_des">PN</data>
        </edge>
        <edge id="a1" source="n1" target="n2">
                <data key="src_des">N</data>
        </edge>
        <edge id="a2" source="n2" target="n3">
                <data key="src_des">0</data>
                <data key="des_src">NN</data>
        </edge>
        <edge id="a3" source="n1" target="n5">
                <data key="src_des">N</data>
        </edge>
        <edge id="a4" source="n2" target="n6">
                <data key="src_des">PN</data>
        </edge>
        <edge id="a5" source="n4" target="n5">
                <data key="des_src">PNN</data>
        </edge>
        <edge id="a6" source="n5" target="n6">
                <data key="des_src">PNN</data>
        </edge>
        <edge id="a7" source="n6" target="n7">
                <data key="des_src">PNN</data>
        </edge>
        <edge id="a8" source="n5" target="n8">
                <data key="src_des">N</data>
        </edge>
        <edge id="a9" source="n6" target="n9">
                <data key="src_des">PN</data>
        </edge>
    </graph>
</graphml>
```

Grafo de giros:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
        <graph id="example_1" edgedefault="directed">
                <node id="a0_src_des_0"/>
                <node id="a0_src_des_1"/>
                <node id="a0_des_src_0"/>
                <node id="a1_src_des_0"/>
                <node id="a1_des_src_0"/>
                <node id="a2_des_src_0"/>
                <node id="a2_des_src_1"/>
                <node id="a3_src_des_0"/>
                <node id="a3_des_src_0"/>
                <node id="a4_src_des_0"/>
                <node id="a4_src_des_1"/>
                <node id="a4_des_src_0"/>
                <node id="a5_src_des_0"/>
                <node id="a5_src_des_1"/>
                <node id="a5_src_des_2"/>
                <node id="a5_des_src_0"/>
                <node id="a5_des_src_1"/>
                <node id="a5_des_src_2"/>
                <node id="a6_src_des_0"/>
                <node id="a6_src_des_1"/>
                <node id="a6_src_des_2"/>
                <node id="a6_des_src_0"/>
                <node id="a6_des_src_1"/>
                <node id="a6_des_src_2"/>
                <node id="a7_src_des_0"/>
                <node id="a7_src_des_1"/>
                <node id="a7_src_des_2"/>
                <node id="a7_des_src_0"/>
                <node id="a7_des_src_1"/>
                <node id="a7_des_src_2"/>
                <node id="a8_src_des_0"/>
                <node id="a8_des_src_0"/>
                <node id="a9_src_des_0"/>
                <node id="a9_src_des_1"/>
                <node id="a9_des_src_0"/>

                <edge id="e00" source="a0_src_des_0" target="a1_des_src_0"/>
                <edge id="e01" source="a0_src_des_0" target="a4_src_des_0"/>

                <edge id="e02" source="a0_src_des_1" target="a1_des_src_0"/>
                <edge id="e03" source="a0_src_des_1" target="a4_src_des_1"/>

                <edge id="e04" source="a1_src_des_0" target="a0_des_src_0"/>
                <edge id="e05" source="a1_src_des_0" target="a4_src_des_0"/>
                <edge id="e06" source="a1_src_des_0" target="a4_src_des_1"/>

                <edge id="e07" source="a2_des_src_0" target="a0_des_src_0"/>
                <edge id="e08" source="a2_des_src_0" target="a1_des_src_0"/>

                <edge id="e09" source="a2_des_src_1" target="a4_src_des_0"/>
                <edge id="e10" source="a2_des_src_1" target="a4_src_des_1"/>

                <edge id="e11" source="a3_src_des_0" target="a5_des_src_0"/>
                <edge id="e12" source="a3_src_des_0" target="a5_des_src_1"/>
                <edge id="e13" source="a3_src_des_0" target="a6_src_des_0"/>
                <edge id="e14" source="a3_src_des_0" target="a6_src_des_2"/>
                <edge id="e15" source="a3_src_des_0" target="a8_src_des_0"/>

                <edge id="e16" source="a4_src_des_0" target="a6_des_src_0"/>
                <edge id="e17" source="a4_src_des_0" target="a7_src_des_0"/>
                <edge id="e18" source="a4_src_des_0" target="a9_src_des_0"/>

                <edge id="e19" source="a4_src_des_1" target="a6_des_src_1"/>
                <edge id="e20" source="a4_src_des_1" target="a7_src_des_2"/>
                <edge id="e21" source="a4_src_des_1" target="a9_src_des_1"/>

                <edge id="e22" source="a4_des_src_0" target="a1_des_src_0"/>
                <edge id="e23" source="a4_des_src_0" target="a0_des_src_0"/>

                <edge id="e24" source="a5_src_des_0" target="a8_src_des_0"/>
                <edge id="e25" source="a5_src_des_0" target="a3_des_src_0"/>
                <edge id="e26" source="a5_src_des_0" target="a6_src_des_0"/>

                <edge id="e27" source="a5_src_des_1" target="a8_src_des_0"/>
                <edge id="e28" source="a5_src_des_1" target="a6_src_des_1"/>
                <edge id="e29" source="a5_src_des_1" target="a6_src_des_2"/>

                <edge id="e30" source="a5_src_des_2" target="a3_des_src_0"/>
                <edge id="e31" source="a5_src_des_2" target="a6_src_des_1"/>
                <edge id="e32" source="a5_src_des_2" target="a6_src_des_2"/>
```

```
            <edge id="e33" source="a6_src_des_0" target="a9_src_des_0"/>
            <edge id="e34" source="a6_src_des_0" target="a4_des_src_0"/>
            <edge id="e35" source="a6_src_des_0" target="a7_src_des_0"/>

            <edge id="e36" source="a6_src_des_1" target="a9_src_des_1"/>
            <edge id="e38" source="a6_src_des_1" target="a7_src_des_1"/>
            <edge id="e39" source="a6_src_des_1" target="a7_src_des_2"/>

            <edge id="e40" source="a6_src_des_2" target="a4_des_src_0"/>
            <edge id="e41" source="a6_src_des_2" target="a7_src_des_1"/>
            <edge id="e42" source="a6_src_des_2" target="a7_src_des_2"/>

            <edge id="e43" source="a6_des_src_0" target="a3_des_src_0"/>
            <edge id="e44" source="a6_des_src_0" target="a5_des_src_0"/>
            <edge id="e45" source="a6_des_src_0" target="a8_src_des_0"/>

            <edge id="e46" source="a6_des_src_1" target="a3_des_src_0"/>
            <edge id="e47" source="a6_des_src_1" target="a5_des_src_1"/>
            <edge id="e48" source="a6_des_src_1" target="a5_des_src_2"/>

            <edge id="e49" source="a6_des_src_2" target="a8_src_des_0"/>
            <edge id="e50" source="a6_des_src_2" target="a5_des_src_1"/>
            <edge id="e51" source="a6_des_src_2" target="a5_des_src_2"/>

            <edge id="e52" source="a7_des_src_0" target="a4_des_src_0"/>
            <edge id="e53" source="a7_des_src_0" target="a6_des_src_0"/>
            <edge id="e54" source="a7_des_src_0" target="a9_src_des_0"/>

            <edge id="e55" source="a7_des_src_1" target="a4_des_src_0"/>
            <edge id="e56" source="a7_des_src_1" target="a6_des_src_1"/>
            <edge id="e57" source="a7_des_src_1" target="a6_des_src_2"/>

            <edge id="e58" source="a7_des_src_2" target="a9_src_des_1"/>
            <edge id="e59" source="a7_des_src_2" target="a6_des_src_1"/>
            <edge id="e60" source="a7_des_src_2" target="a6_des_src_2"/>

            <edge id="e61" source="a8_des_src_0" target="a6_src_des_0"/>
            <edge id="e62" source="a8_des_src_0" target="a6_src_des_1"/>
            <edge id="e63" source="a8_des_src_0" target="a3_des_src_0"/>
            <edge id="e64" source="a8_des_src_0" target="a5_des_src_0"/>
            <edge id="e65" source="a8_des_src_0" target="a5_des_src_2"/>

            <edge id="e66" source="a9_des_src_0" target="a7_src_des_0"/>
            <edge id="e67" source="a9_des_src_0" target="a7_src_des_1"/>
            <edge id="e68" source="a9_des_src_0" target="a4_des_src_0"/>
            <edge id="e69" source="a9_des_src_0" target="a6_des_src_0"/>
            <edge id="e70" source="a9_des_src_0" target="a6_des_src_2"/>
        </graph>
</graphml>
```

4.3 Señales de tráfico

Como se mencionó anteriormente, las señales con las que cuenta el simulador son semáforos de tres estados. Estos semáforos han sido modelados por mi en Blender [7] e importados en Unity3D.

Su funcionamiento consiste en realizar un ciclo contínuo desde el estado rojo, pasando después a verde, luego a naranja y, finalmente, a rojo de nuevo.

El tiempo que tardan en ponerse en verde por primera vez vendrá determinado por el número de semáforos que haya en un cruce. Se decide aleatoriamente el orden en el que se pondrán en verde y a partir de ahí se calcula el tiempo de retardo para comenzar el ciclo de tal forma que cada semáforo del cruce permanecerá en rojo el tiempo que los demás semáforos utilizan para sus estados verde y naranja.

4.4 Vehículos

Se importarán los siguientes modelos utilizando la herramienta Blender [7] o el propio importador de modelos obj de Unity3D:

- Bus
- Taxi Checker Marathon
- Chevrolet Camaro
- Todoterreno verde
- Todoterreno naranja
- Cabeza tractora camión

Estos modelos no han sido realizados por mi, han sido tomados de lugares con las licencias adecuadas.

Una vez cargados en Unity3D se han ajustado sus escalas en el editor de forma que respetasen en la medida de lo posible las proporciones en las medidas entre unos vehículos y otros.

4.5 Conductores

Para simular la lógica de los conductores se ha utilizado un modelo muy básico fundamentándose en tres tipos de conductor:

- Buenos
- Regulares
- Malos

Los buenos conductores respetan la señalización de los semáforos en todo momento, los conductores regulares la respetan con una probabilidad del 50%, mientras que los malos conductores no respetarán dicha regulación en ningún momento.

4.6 Entorno

El entorno está formado por un suelo con textura de hierba y un skybox para formar el cielo. Los nodos límite son falsos túneles modelados por mi en Blender [7] e importados en Unity3D.

Los distintos elementos que conforman las vías de la red han sido modelados en base a cubos escalados dada la complicación para dibujar texturas dentro de otras texturas y así obtener un único plano con la vía representada.

4.7 Funcionamiento de la aplicación

4.7.1 Menú principal

Una vez iniciada la aplicación y pasada la splash screen de Unity3D (estamos usando la versión gratuita de la herramienta) llegamos al menú principal:

Desde aquí podemos cargar algún mapa (Start), ver información de la aplicación (Credits) o salir (Exit).

Cuando vamos a cargar un mapa, la aplicación lee los nombres de los ficheros, que se encuentran en la carpeta

```
"Traffic-Flow-Simulator__x86_Data\StreamingAssets\Maps"
```

en el caso de la aplicación para Windows 32 bits o

```
"Traffic-Flow-Simulator__x86_64_Data\StreamingAssets\Maps"
```

en el caso de la aplicación para Windows 64 bits, y muestra tantos botones como mapas haya, con sus respectivos nombres. Una vez que se pulsa alguno de ellos comienza el proceso de carga y creación de la pantalla de simulación.

4.7.2 Pantalla de simulación

Al cargar la pantalla de simulación, tanto el skybox como los controles de la Interfaz de Usuario ya están preparados, a continuación se creará el suelo con las dimensiones del mapa cargado más un margen, se representará la red viaria, se posicionará la cámara y dará comienzo la simulación con la creación de vehículos.

4.7.3 Representación de los arcos

La representación de la red viaria comienza con el dibujado de los arcos del grafo (los tramos rectos de red viaria) que se dibujan de la siguiente forma:

Se comienza con una plataforma viaria escalada en el eje Z de forma que el nodo origen del arco se encuentre hacia la parte negativa del eje Z y el nodo destino hacia la parte positiva del eje Z. A continuación se dibujan las líneas de los arcenes. Seguidamente, si el arco tiene carriles en ambas direcciones se dibujan las líneas de mediana (líneas centrales de división de sentidos), atendiendo a que éstas verán modificada su posición desde el centro del arco en función de la diferencia del número de carriles que haya en cada sentido.

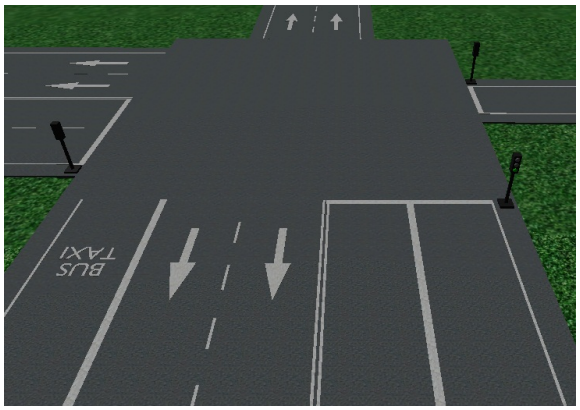
A continuación, para cada uno de los sentidos en el que haya carriles, se dibujarán tantas líneas de carril como carriles haya en dicho sentido menos una, desde el exterior del arco hacia el centro, atendiendo al tipo de carril que le haya sido asignado, así, un carril taxi bus será delimitado por la izquierda por una línea contínua ancha mientras que los carriles normales serán delimitados por la izquierda por líneas discontínuas estrechas.

Para finalizar el dibujado de líneas, si los carriles del sentido que se está dibujando llegan a un nodo de intersección, se dibujará una línea de detención perpendicular a las anteriores.

Además, al comienzo de cada carril se dibujarán marcas horizontales que indicarán el tipo de carril: siendo la flecha blanca para los carriles normales y las letras TAXI BUS para los carriles de tipo público.

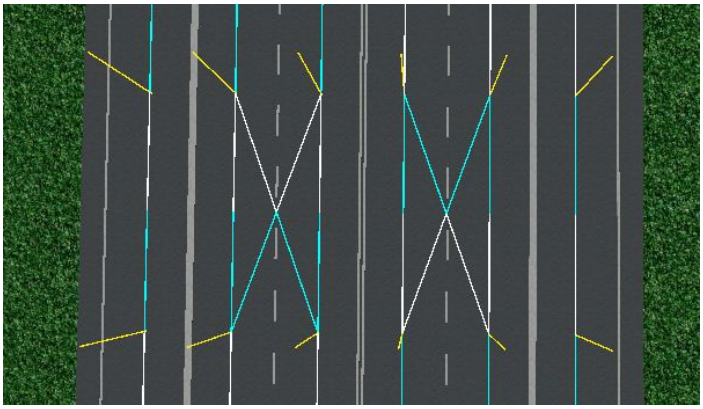
Por último, se añadirán los semáforos al final del arco en el arcén de cada sentido si se desemboca en un cruce, orientados adecuadamente según el sentido que se esté dibujando.

A continuación podemos ver una imagen que ilustra la explicación anterior:



Una vez que la representación del arco ha sido construida se coloca en su posición y se orienta para quedar alineada con los nodos origen y destino. La posición no será el punto medio de las posiciones de los nodos ya que según el tipo de nodo que se encuentre en el extremo habrá que realizar un pequeño ajuste de posición. Además la longitud del arco tampoco será la distancia entre los nodos ya que entonces se solaparían las representaciones, esta ha sido calculada de forma que no solape en ninguno de los dos extremos con las representaciones de los nodos.

Además, para el guiado de los vehículos, los arcos contarán con nodos guía (invisibles en la aplicación) al principio y final de cada carril, así como dos nodos guía en el centro que servirán para hacer cambios de carril si los carriles contiguos son del mismo tipo. Veamos un ejemplo:



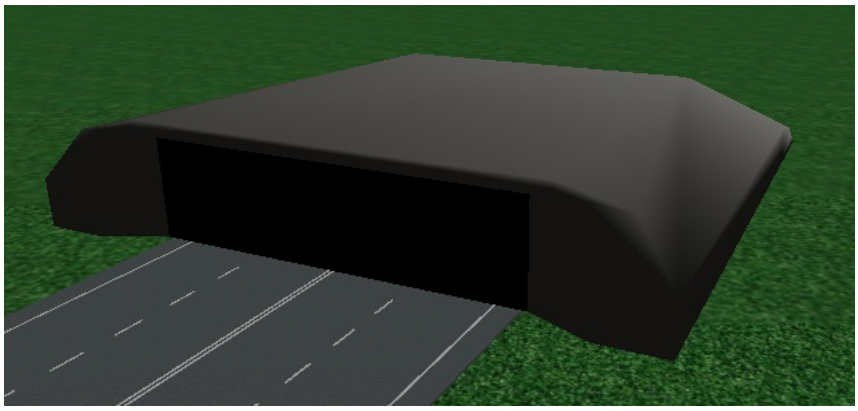
Como se puede apreciar en la imagen, los carriles más exteriores no permiten cambio de carril dado que son de tipo público y no hay más carriles de tipo público contiguos a ellos, mientras que los demás si permiten el cambio ya que son contiguos de tipo normal.

4.7.4 Representación de los nodos

A continuación se dibujan los nodos. Según su tipo, serán dibujados de la siguiente forma:

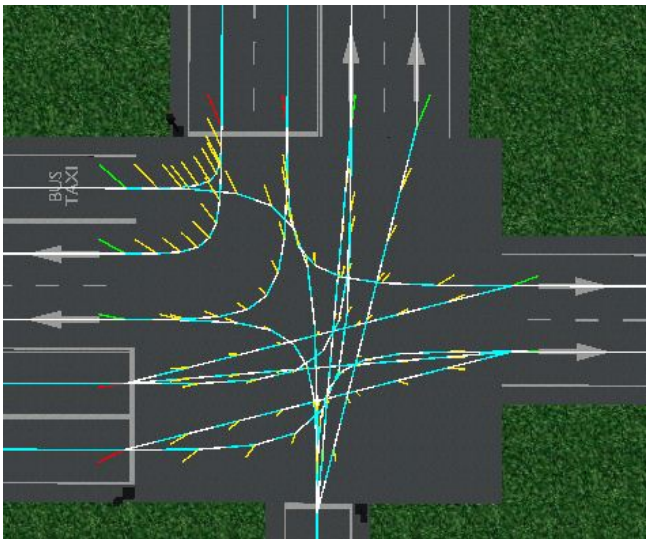
- Nodos límite:
  La representación de los nodos límite es la más sencilla que podemos encontrar, consiste en instanciar un tunel 3D en la posición del nodo límite, escalarlo a lo ancho para abarcar todo el ancho del arco que llega hasta él y rotarlo de manera que quede alineado con dicho arco.

  Este tipo de nodo no afecta en manera alguna la circulación de los vehículos, por tanto, los vehículos se introducirán en el tunel y desaparecerán al llegar al nodo guía de final de carril por no tener ningún nodo guía siguiente.



- Nodos de intersección:
  Para representar un nodo de intersección se toma una plataforma viaria vacía, es decir sin líneas ni marcas, se instancia en la posición correspondiente y se escala de tal forma que quede una forma completamente cuadrada cuyos laterales tienen la misma dimensión que la anchura del arco más ancho de los que inciden en dicho nodo.

  Además, para el guiado de los vehículos, el nodo de intersección une los nodos guía de final de carril con los de inicio de carril según los giros permitidos indicados en el fichero de giros explicado en la sección *4.2 Red viaria*. Para ello, traza una curva de Bézier [9] con nodos guía intermedios por cada uno de los giros permitidos.

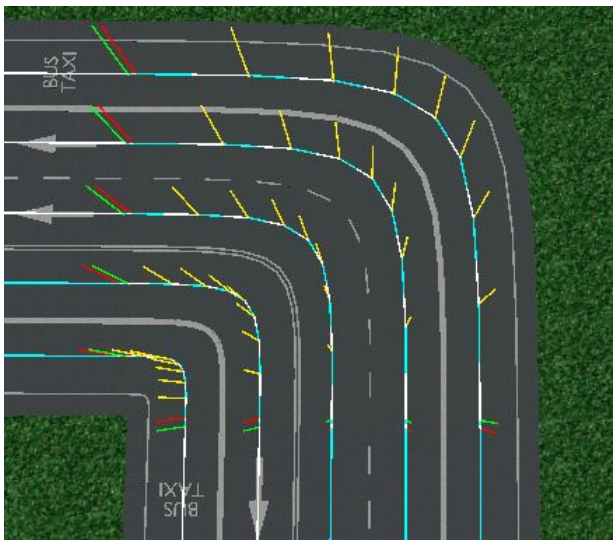


- Nodos de continuación:

  Los nodos de continuación (tramos curvos) se dibujan de forma análoga a los arcos, con un par de excepciones: en estos nodos no se permite el cambio de carril y en vez de ser una plataforma recta son diez tramos de plataforma con sus líneas correspondientes que conforman la forma curva.

  Para conseguir la forma curva se ha utilizado una curva de Bézier [9] con tres puntos, el punto central de la curva en el extremo del arco de referencia, el punto central de la curva en el extremo del otro arco que incide en el nodo y el punto central en el que se cruzarían los arcos. Este último punto se utiliza como los dos puntos de control propios de una curva de Bézier [9].

Además, se incluyen nodos de guiado intermedios para que los vehículos simulen el giro a lo largo de la sección curva. Finalmente se coloca en la posición adecuada y se alinea con los arcos que inciden en el nodo, como se puede apreciar en la siguiente imagen:



4.7.5 Finalizando la representación

Como paso final, se unen los nodos guía de final de carril de los arcos con los nodos correspondientes de los nodos de continuación e intersección.

Una vez representada toda la red viaria se sincronizan los semáforos tal y como se explicó en el apartado *4.3 Señales de tráfico*.

Así concluye la representación de la red viaria, a continuación, da comienzo la simulación generando los vehículos atendiendo a los parámetros de la interfaz de usuario.

Dicha interfaz pondrá a disposición del usuario unos controles deslizantes (sliders) para controlar las siguientes variables:

- El número de vehículos dentro de la simulación, con un mínimo de 0 y un máximo de 200.
- La proporción de cada tipo de conductor, bueno, regular y malo, con un mínimo de 0% y un máximo de 100%, respetándose un total entre los tres tipos del 100%.
- La proporción de cada tipo de vehículo, público y privado, con un mínimo de 0% y un máximo de 100%, respetándose un total entre los dos tipos del 100%.

- El tiempo que pasan los semáforos en verde, con un mínimo de 5 segundos y un máximo de 60 segundos.
- El tiempo que tardan en eliminarse de la simulación los vehículos accidentados, con un mínimo de 0 segundos y un máximo de 180 segundos.

Así mismo, la interfaz también dispone de un contador de vehículos accidentados, un botón para pausar/reanudar la simulación o volver al menú principal, un botón para mostrar el panel de ayuda con la información relativa a los controles de la cámara y un botón para mostrar/ocultar los nombres de las calles.

4.7.6 Comportamiento de los vehículos

Cada vehículo es instanciado en un nodo guía, dentro de un túnel, y se le asigna como objetivo que se desplace al siguiente nodo guía, cuando llega a este, comprueba a cuántos nodos guía puede avanzar atendiendo al tipo de vehículo que puede circular por cada nodo y elige uno de forma aleatoria, repitiendo este ciclo de forma indefinida hasta que llega a un nodo el cual no tiene nodos guía siguientes, lo cual significa que el vehículo se encuentra dentro de un tunel y que debe eliminarse de la simulación. Esto provocará una actualización automática de los contadores de cantidad de vehículos, tipo de vehículos y tipos de conductor, de cara a la instanciación de nuevos vehículos.

Cada vehículo va comprobando constantemente que no haya vehículos en su camino a una distancia inferior a diez metros, en cuyo caso moderará su velocidad para no chocar con el vehículo que tenga inmediatamente delante. Para ello comprueba si existe alguno entre su posición y su nodo guía objetivo, si no hay ninguno y aún no ha comprobado los diez metros, comprobará de forma recursiva desde el nodo guía objetivo hasta cualquiera de los siguientes, y así sucesivamente.

Además, cada vehículo cuenta con dos rayos, para raycasting, de metro y medio en la parte frontal orientados hacia delante y hacia los lados para comprobar que no hay vehículos en las inmediaciones por delante; así como un rayo más en la parte frontal de 10 metros para la detección de los semáforos.

Si un vehículo detecta un semáforo en verde no hará ningún cambio, mientras que si lo detecta en naranja o rojo disminuirá la velocidad hasta detenerse cerca de la línea de detención. Esto ocurrirá para los buenos conductores, ya que, como se mencionó en el apartado *4.5 Conductores*, los

malos conductores no lo harán y los conductores regulares se lo pensarán constantemente mientras se acercan al semáforo, ya que en cada frame realizan una evaluación de la situación, pudiendose casi detener y luego reanudar la marcha con el semáforo en un estado distinto al verde.

Bibliografía

[1] D. Drew, *Traffic flow theory and control*. New York: McGraw-Hill, 1968.

[2] M. Pursula, "Simulation of traffic systems - an overview," *Journal of Geographic Information and Decision Analysis*, vol. 3, no. 1, pp. 1–8, 1999.

[3] H. Kallberg, "Traffic simulation," Master's thesis, University of Technology, Transportation Engineering, Espoo, 1971.

[4] J. Barceló, ed., *Fundamentals of Traffic Simulation*, vol. 145 of *International Series in Operations Research & Management Science*. Springer New York, 2010.

[5] P. Bogaert, V. Geografie, N. V. D. Weghe, V. Geografie, R. Maddens, V. Geografie, L. D. Temmerman, V. Geografie, and V. Geografie, "A new data structure to represent road networks."

[6] "Unity web page." `unity3d.com/es`.

[7] "Blender web page." `www.blender.org`.

[8] GraphML Team, *The GraphML File Format*. graphml.graphdrawing.org, 2001-2007.

[9] "Curva de bézier." `es.wikipedia.org/wiki/Curva_de_Bézier`.