

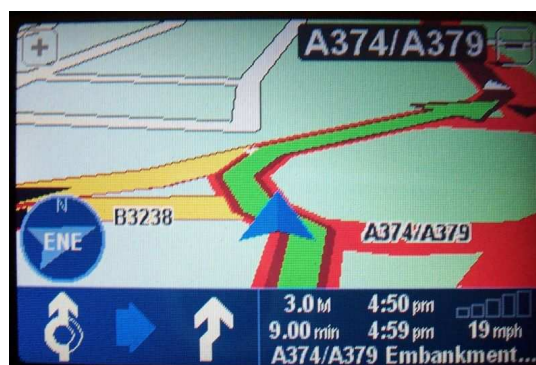
Laboratorio de Programación II. Práctica Curso 2007-2008.

Ingeniería Técnica en Informática de Gestión.

Francisco Chávez de la O.

Navegador CUM.

1.	Introducción.	2
2.	Descripción de la práctica.	2
3.	Comienzo de la ejecución.	3
4.	Descripción de los ficheros.	3
4.1.	Monumentos.txt.....	3
4.2.	Turistas.txt.....	4
4.3.	Nombre_calle.txt.....	4
5.	Descripción de clases más destacadas.....	6
5.1.	Clase elemento del grafo.	6
6.	Elementos del juego.	6
7.	Creación de las pantallas de trabajo.	8
8.	Detalle de funcionamiento del juego.....	8
8.1.	Comienza el juego.	8
8.2.	Selección de visita.	9
8.3.	Viaje entre monumentos.....	9
8.4.	Selección de nueva visita o fin del juego.	11
8.5.	Descarga de datos.	12
9.	Especificaciones del programa.....	12
10.	Documentación.	12



1. Introducción.

En este curso vamos a desarrollar un juego que simule un sistema de navegación para automóviles. El juego consiste en completar una serie de visitas turísticas en Mérida a lo largo de las cuales se deberán superar multitud de dificultades hasta llegar al monumento deseado. En cada una de las visitas hay que superar obstáculos y obtener el máximo de bonificaciones para conseguir que la ruta sea lo mejor posible.

La misión del conductor es llegar con el automóvil en perfecto estado al monumento deseado en un tiempo limitado. Se deberá usar el ratón y el teclado para seleccionar el monumento a visitar y para mover el automóvil.

Para la implementación del juego se debe utilizar programación orientada a objetos bajo el lenguaje de programación C++.

2. Descripción de la práctica.

El objetivo de este juego es poder llegar al monumento seleccionado con el máximo número de bonificaciones y el automóvil en perfecto estado.

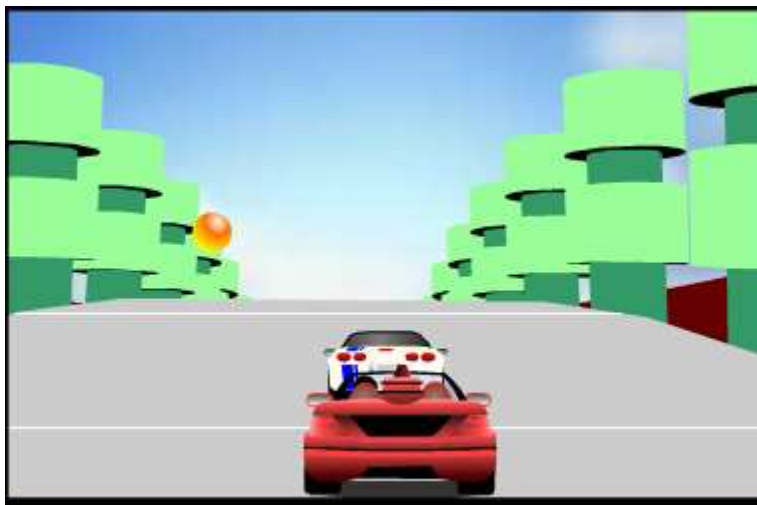


Figura 1. Pantalla durante el juego.

Durante la ruta tendrás que esquivar los obstáculos que vayan apareciendo y conseguir el máximo de bonificaciones para poder avanzar hasta el monumento seleccionado. Todo dependerá de tu habilidad y destreza para manejar el automóvil.

3. Comienzo de la ejecución.

Al comienzo de la ejecución debe cargarse la configuración del juego, dicha configuración será suministrada al alumno en ficheros de texto (que se deberán pasar **obligatoriamente** como argumentos al programa principal), los cuales deberán almacenarse en estructuras estáticas y dinámicas de datos indicadas a continuación.

Nombre de fichero.	Estructura de almacenamiento
Monument.txt	Grafo
Turistas.txt	Árbol
Segmento.txt	Cola de segmentos de cada calle

Una vez mostrada la información del juego, se debe presentar la configuración de los diferentes monumentos de Mérida (dicha información está disponible en el grafo). Junto a la configuración de los diferentes monumentos se debe mostrar los caminos existentes entre cada uno de ellos, el nombre de la calle a tomar y la distancia que hay entre un monumento y otro.

4. Descripción de los ficheros.

4.1. Monumentos.txt

```
int velocidad
int gasolina
Nombre_monumento
Nombre_monumento
Nombre_monumento
Nombre_monumento
Nombre_monumento
char *#char *#char *#char *#char *
char *#char *#char *#char *#char *
char *#char *#char *#char *#char *
char *#char *#char *#char *#char *
char *#char *#char *#char *#char *
```

int velocidad: Campo que indicará la velocidad inicial del automóvil. Cada vez que el jugador consiga una determinada bonificación podrá aumentar la velocidad del automóvil, consiguiendo llegar al monumento en menor tiempo.

int gasolina: Campo que indicará la cantidad de gasolina inicial del automóvil.

Nombre_monum: Campo de tipo cadena, que contendrá el nombre de cada monumento.

*char *#char *#char *#char *#char **: Nombre de la carretera que existe entre para ir de un monumento a otro.
Si el *contenido de un campo es NULL*, significará que **no hay conexión** entre los monumentos.

Ejemplo:

Arco_trajano
Puente_romano
...
NULL#0#Santa_Eulalia#1200#Rambla#500#NULL#0#NULL#0
...

El “Arco_trajano” está conectado directamente con el “Puente_romano” a través de la calle Santa Eulalia que tiene una longitud de 1200 metros.

4.2. Turistas.txt

Nombre_turista#fecha_visita#Nacionalidad#Monumento1#Monumento2#...

Cuando se inicie una partida, se deberá cargar este fichero en un árbol binario ordenado por nombre del turista, además se almacenará la fecha en que visitó Mérida, su nacionalidad y una lista de los monumentos visitados, una vez que se finalice el juego, el turista se almacenará en el árbol binario ordenado y se volcarán los datos del árbol, en el fichero Turista.txt. Si el turista repite visita, se deberá modificar su registro, actualizando la fecha de última visita y añadiendo los monumentos nuevos que haya visto en esta segunda, tercera, ...visita.

4.3. Nombre_calle.txt

Habrán tantos ficheros de este tipo como enlaces entre los monumentos a visitar. En cada uno se almacenarán las características de cada calle, con los obstáculos y bonificaciones que existan.

- Elementos obstáculos:
 - o Bache.
 - o Accidente.
- Elementos bonificadores:
 - o Estatua.
 - o Vasija.

El formato del fichero será el descrito a continuación:

color

X_superior

Y_superior

X_inferior

Y_inferior

Nombre_calle#1#o#bache#5#7
char *#int#char# char *# int#int

-----> Para elementos tipo Bache.

Nombre_calle #1#o#accidente#5#5#15
char *#int#char# char *# int#int#int

-----> Para elementos tipo accidente.

Nombre_calle #1#b#estatua#5#5#100
char *#int#char#char *#int#int#int

-----> Para elementos tipo estatua.

Nombre_calle #2#b#vasija#8#7#150#10
char *#int#char#char *#int#int#int#int

-----> Para elementos tipo vasija.

Cada línea descrita anteriormente corresponde a un tipo de elemento que nos podemos encontrar en cada uno de los segmentos que formarán la calle y que se cargarán en una cola de objetos (obstáculos y bonificaciones).

Ejemplo:

0

100

100

200

300

Trajano#1#o#bache#5#7

0:Color de la calle negro

100: coordenada x superior izquierda

100: coordenada y superior izquierda

200: coordenada x inferior derecha

300: coordenada y inferior derecha

Ela última línea pertenece a la formación de la calle *Trajano*, en concreto es el segmento *número 1*, que está formado por un obstáculo del tipo bache.

5. Descripción de clases más destacadas.

5.1. Clase elemento del grafo.

El grafo de monumentos a visitar estará formado por un vector que contiene los nombres de cada uno de los monumentos que pueden ser visitados por el turista, junto con una matriz que describe las diferentes conexiones entre estos monumentos, es decir, una matriz de punteros a una clase elementos del grafo como la descrita a continuación:

Elementos del grafo:

Nombre_calle	char*
Cola_segmentos	cola *

Donde:

Nombre_calle: indica la carretera entre los nodos.

Cola_segmentos: Una cola con los trozos que contiene la carretera.

6. Elementos del juego.

Los turistas que visiten la ciudad realizarán dicha visita por medio de dos tipos diferentes de coches. Por una parte comenzarán la visita con un coche más antiguo y lento y a medida que recopilen puntos podrán cambiar a un coche más moderno y rápido.

Los dos tipos de coches estarán formados por la carrocería, *ruedas y faros* y cada uno de ellos tendrá una forma diferente.

El primer coche lo llamaremos coche normal y tendrá las siguientes características:

- Carrocería cuadrada.
- Formado por 6 ruedas.
- 2 faros.
- **Únicamente movimientos horizontales.**
- Movimientos lentos, cada pulsación de las teclas que lo muevan de izquierda a derecha **avanzará una única posición** en la pantalla.

El segundo tipo de coche lo llamaremos bólido y tendrá las siguientes características:

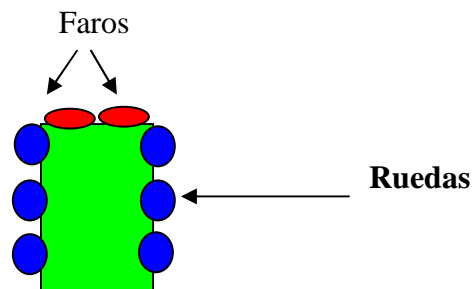
- Carrocería elíptica.
- Formado por 4 ruedas.
- 2 faros.
- **Movimientos horizontales y verticales.**
- Movimientos rápidos, cada pulsación de la teclas se traducirá en un avance de 4 posiciones en la pantalla tanto en vertical como en horizontal.

Los coches disponen de un depósito de gasolina de 50 litros como máximo, la cantidad de gasolina inicial se dará en el fichero Monument.txt. Este depósito se irá consumiendo a lo largo del avance del juego, por lo que, si el coche se queda sin gasolina, se parará y se terminará el juego. A lo largo de las diferentes visitas, el turista podrá ir acumulando gasolina cuando obtenga los tipos de bonificación necesarios para ello.

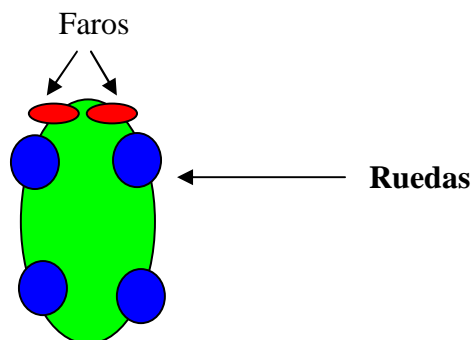
Por otra parte, las ruedas de los coches son muy frágiles, por lo que si toman un bache, se destrozará una de las ruedas que tiene el coche si no tiene recambios para colocar una nueva, el juego finalizará. Estas nuevas ruedas podrán ir siendo acumuladas a lo largo de las diferentes visitas, tomando elementos bonificadores.

El formato de cada tipo de coche es el siguiente:

Coche normal.



Bólico.



La carrocería de los diferentes coches estará formada por círculos o elipses. Las diferentes figuras necesarias serán las siguientes:

- Figura base, compuesta por:

X int
Y int
Color int

- Círculo, compuesto por:

Figura base
Radio int

- Rectángulo, compuesto por:

Figura base
Xf int
Yf int

- Elipse, compuesta por:

Figura base
Radio_mayor int
Radio_menor int

De esta forma las carrocerías de los diferentes coches serán rectángulos o elipses, las ruedas serán elementos del tipo círculo y los faros del coche serán elementos del tipo elipse.

7. Creación de las pantallas de trabajo.

La práctica se debe desarrollar utilizando el modo gráfico del lenguaje C++. Las diferentes pantallas del juego se deben crear de forma similar a la de la figura 1, mostrando los puntos conseguidos, el camino que está realizando el turista en cada momento y el tiempo transcurrido.

8. Detalle de funcionamiento del juego.

8.1. Comienza el juego.

Al comienzo de la ejecución del juego, una vez cargada la configuración del mismo desde los ficheros, deberá mostrarse por pantalla la información del juego, teclas que pueden usarse junto con una breve descripción del juego. En cualquier momento el jugador podrá abandonar el juego, indicándose la acción en la pantalla inicial del juego.

El siguiente paso será la presentación de los diferentes monumentos que pueden ser visitados junto con sus conexiones. Se partirá de la Oficina de Turismo de Mérida situada en la calle Santa Eulalia. Todos los monumentos se mostrarán del mismo color, la información correspondiente estará almacenada en el fichero denominado

Monument.txt. Este fichero contiene el nombre de todos los monumentos, el nombre de las calles donde se encuentran, así como las conexiones entre cada uno de ellos indicando la distancia. En el transcurso de la ejecución estos datos se almacenarán en la estructura de datos estática "*grafo*".

Por último debe solicitarse el nombre del turista que va a realizar la visita, su nacionalidad y guardar la fecha de su visita (fecha del sistema, ddmmaa).

8.2. Selección de visita.

Una vez que el grafo de visitas ha sido presentado en pantalla se debe seleccionar con el ratón el monumento que se desea visitar. Con ello el programa debe analizar si es posible realizar esa visita, es decir, si existe camino entre los dos nodos del grafo. En caso de que pueda realizarse la visita comenzará el juego, si esta visita no se puede realizar se visualizará un mensaje de error y el turista podrá seleccionar otro monumento o abandonar el juego.

8.3. Viaje entre monumentos.

A lo largo del viaje entre los monumentos, el turista irá encontrándose con diferentes obstáculos y bonificaciones.

El jugador debe mover el coche horizontal y verticalmente para poder salvar los obstáculos y para tomar las bonificaciones.

La acción de cada uno de los diferentes objetos que puede encontrarse en la carretera es la siguiente:

Estatua.

Cada vez que se encuentre una estatua en el camino, si el jugador la toma, podrá acumular una rueda de repuesto, para futuros imprevistos a lo largo de la visita.

Vasija.

Las vasijas contienen gasolina que se acumulará en la lista de extras del coche.

Bache.

Este obstáculo es muy peligroso para el viaje, ya que cuando el coche toma uno perderá una rueda, que tendrá que ser sustituida por alguna de las ruedas de repuesto acumuladas a lo largo del camino.

Accidente.

Este obstáculo hará que el coche se inmovilice debido al accidente, por lo que no podrá moverse durante un tiempo indicado en el mismo.

Estos obstáculos y bonificaciones serán presentados en la pantalla utilizando **el tipo de figura seleccionada por el alumno.**

La mecánica del juego es la siguiente. Una vez que el turista elija la visita que desea realizar se comenzará a presentar en pantalla la calle que debe recorrer para llegar al monumento deseado. Esta calle está formada por segmentos dentro de los cuales se encuentran diferentes tipos de objetos, obstáculos y bonificaciones. Estos segmentos irán avanzando a lo largo de la pantalla de juego, desde la posición superior de la misma hasta la posición final. Junto con el avance de cada uno de los segmentos irán moviéndose los elementos que se encuentran dentro de los mismos. Los segmentos estarán almacenados en la cola de cada elemento del grafo correspondiente al camino entre los dos monumentos que el jugador desea visitar. Dentro de cada segmento de calle se encuentra una lista con los objetos que lo forman.

Los segmentos de calle se tomarán de la cola del elemento del grafo y pasarán a una lista de segmentos activos en pantalla, la cual permitirá recorrerla y modificar las coordenadas para permitir moverlos a lo largo de la pantalla.

A medida que avanza la calle por la pantalla, el coche debe ir sorteando obstáculos y cogiendo bonificaciones. Las bonificaciones que el jugador vaya tomando se almacenarán en la lista extra del coche. Esta lista será recorrida cada vez que el coche necesite algún elemento de la misma.

Tratamiento de los obstáculos.

Cada vez que el coche toma un **bache** éste le hace perder una de las ruedas. La rueda que perderá siempre será la última del vector donde se encuentran almacenadas las ruedas dentro del coche. En este momento se debe **sustituir la rueda** por una nueva. Para ello, se deberá recorrer la lista de extras del coche y buscar una rueda de repuesto, si la tiene, sustituirá a la rueda perdida en el bache. La nueva rueda tomará los valores de la rueda que se debe cambiar.

Cuando el turista se encuentre en su camino con un **accidente**, se **bloqueará el coche durante el tiempo indicado** por el obstáculo, pero la carretera seguirá avanzando por lo que es posible que choque con un nuevo obstáculo o con alguna bonificación que se encuentre en su camino.

Por otro lado, cada vez que el coche avance 3 segmentos de calle (desaparezcan por la parte final de la pantalla), se debe calcular la gasolina restante. La gasolina se decrementará a razón de 10 litros. El coche llegará a la reserva cuando le queden 20 litros en el depósito. En este momento, se debe recorrer la lista de extras del coche para poder rellenar el depósito con la gasolina que haya almacenada en las vasijas. Se podrá rellenar la gasolina de tantas vasijas hasta llenar el depósito.

Por último, si el coche toma un bache y no tiene una rueda de sustitución o se queda sin gasolina y no hay ninguna vasija para rellenar el depósito, el juego finalizará.

Tratamiento de las bonificaciones.

A medida que el coche y la calle avancen, se podrán ir tomando bonificaciones. Estas bonificaciones **serán almacenadas en una lista extra del coche**, para poder realizar los cambios necesarios al ir tomando los diferentes obstáculos.

Las bonificaciones tienen una puntuación que se irá acumulando y permitirán al usuario poder cambiar de coche.

Llegada a un monumento.

Una vez que se ha acabado la calle deben calcularse los puntos que se han acumulado durante el camino para ver si es posible cambiar de tipo de coche.

El tipo de coche cambiará si el turista ha obtenido 1000 puntos. En este momento el tipo de coche con el que jugará será con el bólido, pudiendo avanzar horizontal y verticalmente. Además la lista de bonificaciones que el turista ha acumulado pasará al nuevo coche.

8.4. Selección de nueva visita o fin del juego.

Una vez que el turista ha llegado al monumento deseado, se presentará en la pantalla de nuevo el grafo de monumentos. En este momento se podrá elegir visitar un nuevo monumento o finalizar el juego. Si el usuario selecciona un nuevo monumento, se calculará si es posible avanzar hasta este nuevo monumento desde que el se encuentra ahora o no, indicando esta información en la pantalla. Si es posible avanzar al nuevo monumento, se leerá la calle del monumento que se quiere visitar y se continuará de la forma descrita en la sección anterior.

Si por el contrario el usuario decide abandonar el juego se debe insertar el turista en el árbol de turistas con toda la información del mismo. Si el turista existe en el árbol se modificará la fecha de la visita y la lista de monumentos, añadiendo los que haya visitado en esta ocasión.

Como final del juego se mostrarán por pantalla los turistas cuya fecha de visita coincida con el mes y año que se introduzca por teclado.

8.5. Descarga de datos.

Una vez finalizado el juego se deben descargar los datos correspondientes a los ficheros de turistas, almacenados en el árbol.

9. Especificaciones del programa.

El programa debe implementarse utilizando el lenguaje de programación C++, los compiladores que pueden utilizarse son: Borland C++ 5.0 o Dev-C++. Para la creación de esta práctica se debe utilizar la programación orientada a objetos haciendo uso de las clases y objetos de C++. Debe estar estructurado en un proyecto junto con sus ficheros fuentes y cabeceras.

El alumno podrá introducir tantas modificaciones como estime oportunas aumentando con ello las funcionalidades del juego. Todas las modificaciones extras serán evaluadas de forma positiva.

Nota:

No se recogerá ninguna práctica que no cumpla como mínimo con los siguientes requisitos:

- Prohibido variables globales.
- Los ficheros se deben pasar como argumentos al programa principal.
- Todas las clases deberán tener implementados correctamente sus constructores y destructores.
- No se admitirán métodos virtuales innecesarios.
- Documentación paginada.

10. Documentación.

- Se debe entregar un disquete con los **ficheros fuentes de la práctica, sin el fichero ejecutable**, este será compilado y ejecutado por el profesor.
- La codificación de los diferentes programas y compilación de los mismos se podrá realizar en Borland C++ o Dev C++.
- **La práctica debe funcionar en el laboratorio de prácticas para que pueda ser evaluada y calificada por los profesores.**
- Se debe entregar una **memoria explicativa** de la práctica donde se recojan los siguientes puntos.
 - Índice de todo lo que contiene la memoria incluyendo la paginación del código fuente.
 - Enunciado de la práctica.
 - Objetivos generales de la práctica.



- Objetivos específicos, incluyendo la **jerarquía de clases utilizada**.
- Manual del programador.
- Manual del usuario.
- **No se recogerá ninguna documentación que no se adapte al esquema anterior.**

Nota: Esta práctica puede estar sujeta a cambios que se comunicarán en las diferentes clases de teoría y práctica. Los Profesores.