



**UNIVERSIDADE DA CORUÑA**  
**Escuela Politécnica Superior. Ferrol**



**Ingeniería Industrial**

Título: **DISEÑO DE UN ENTORNO 3D PARA UN  
SIMULADOR DE CONDUCCIÓN DE  
AUTOMÓVILES**

Autor: **LUIS MONZÓN GÓMEZ**

Fecha: **SEPTIEMBRE 2003**

# MARCO DEL PROYECTO

- Se encuadra dentro de las líneas de investigación que lleva a cabo el Laboratorio de Ingeniería Mecánica de la E.P.S. del Campus de Ferrol.
- Proyecto: “Un entorno unificado de simulación para la dinámica en tiempo real de sistemas multicuerpo con análisis tensional y control”, financiado por la CICYT por el trienio 2001-03.

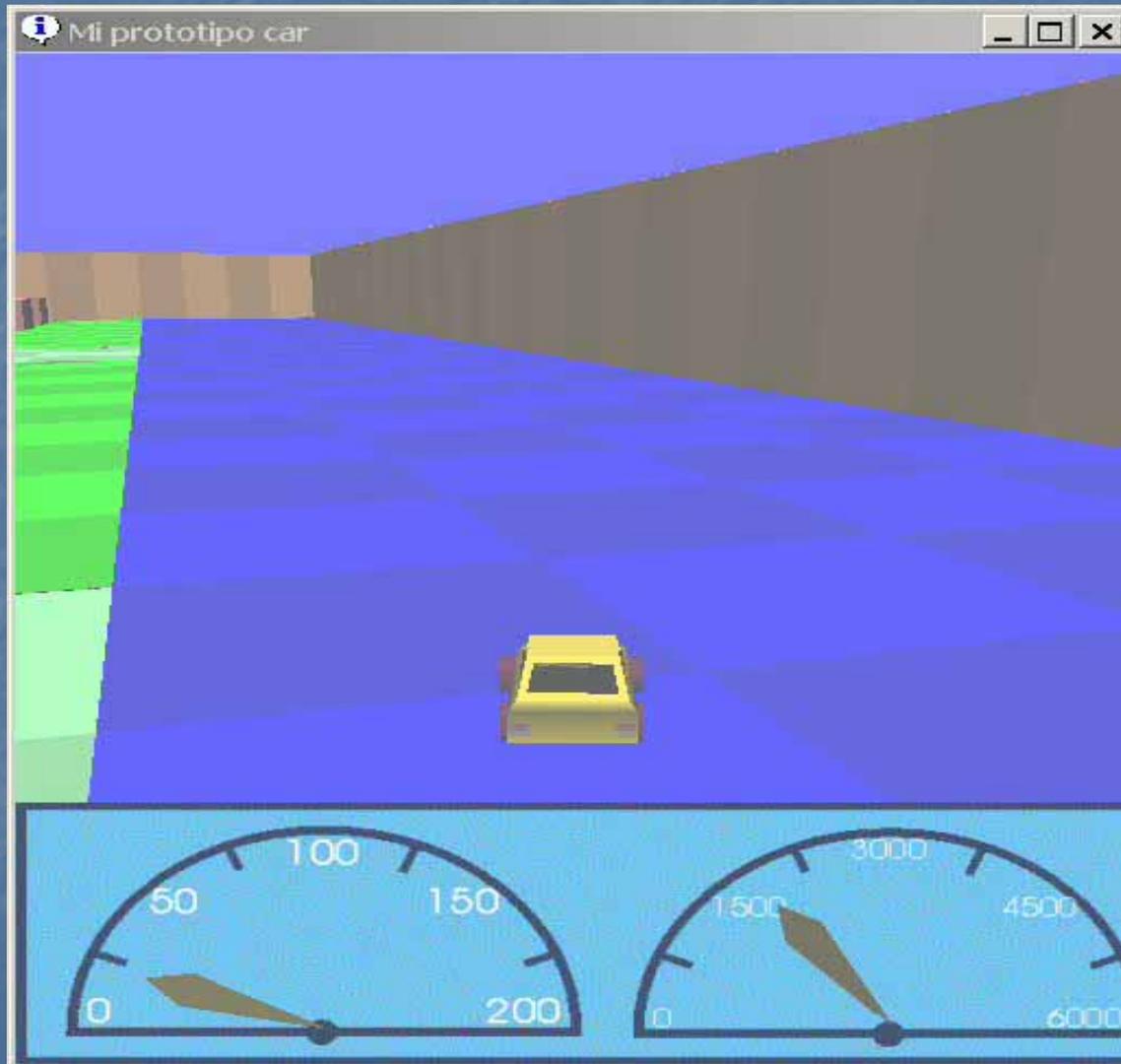
# PUNTO DE PARTIDA DEL PROYECTO

- Proyecto: “Simulación del control de un vehículo automóvil” de Abraham Prieto García (2003).

## ➤ APORTACIONES

- Modelo de simulación del control de un vehículo automóvil, correspondiente a un vehículo experimental del Laboratorio de Ingeniería Mecánica.
- Pauta para la unión del entorno y el modelo de control.

# APLICACIÓN INICIAL



# OBJETIVOS

- Creación de terrenos 3D con un mayor realismo.
- Independencia del terreno empleado en la aplicación.
- Modelado de un vehículo.
- Obtención de datos a partir de un terreno dado: altitud y normal.

# HERRAMIENTAS

- Aplicación 3DStudio Max 4.2.

- Modelado de vehículos.
- Modelado de terrenos.

- Librería gráfica OpenGL Performer.

- Desarrollo de la aplicación.

- Aplicación 3DStudio Max 4.2.

- Modelado de vehículos.

- Aplicación VtBuilder y Enviro.

- Modelado de terrenos.

- Herramientas VtTerrain sobre la librería gráfica OpenSceneGraph.

- Desarrollo de la aplicación.

# MODELADO DEL VEHÍCULO

- Datos:
  - Formato 3Ds. Formato legible por numerosas librerías gráficas.
  - Se han conservado las medidas del vehículo experimental.



- Aplicaciones complementarias:
  - Rhinoceros: como depuración de formatos.

# MODELADO DEL TERRENO

## comparativa de aplicaciones

### 3DStudio

- Requiere un mayor esfuerzo de elaboración.
- Nuevo terreno, necesita su respectivo proceso de modelado.
- Necesidad de elaborar los terrenos con poco detalle para no cargar el proceso de renderizado del simulador

### VTBUILDER

- Entornos ya creados a partir de datos geoespaciales, o bien creados en los más conocidos generadores de terrenos.
- Garantiza sencillez y rapidez en la carga de los distintos terrenos.
- Introduce entornos de cierta complejidad.

# LIBRERÍAS GRÁFICAS

## OPENSCENEGRAPH vs OPENGL PERFORMER

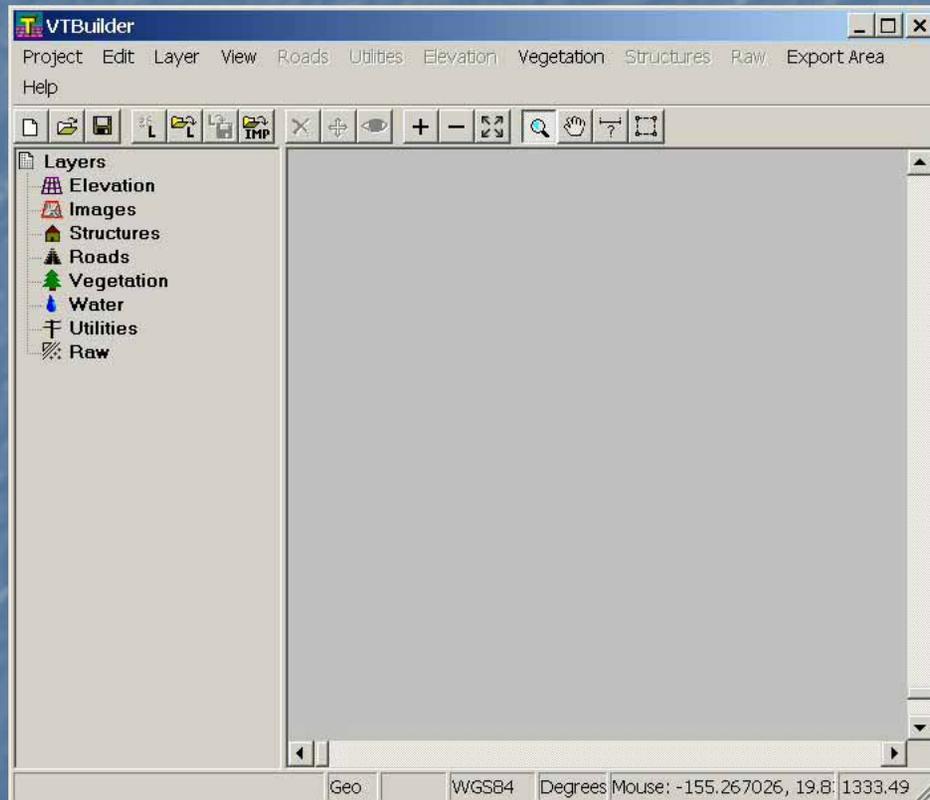
### CARACTERÍSTICAS COMUNES

- Ambas son librerías gráficas de alto nivel empleadas en el volcado de imágenes 3D en "Real Time".
- Se apoyan en la librería gráfica OpenGL para renderizar.
- Ambas trabajan en entornos de C++.

### OPENSCENEGRAPH

- Integrada en el VTP software.

# VTBUILDER

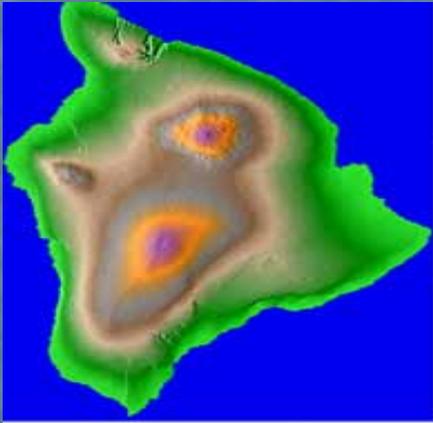


# GENERACIÓN PASO A PASO DE UN ENTORNO DE SIMULACIÓN

Geometría

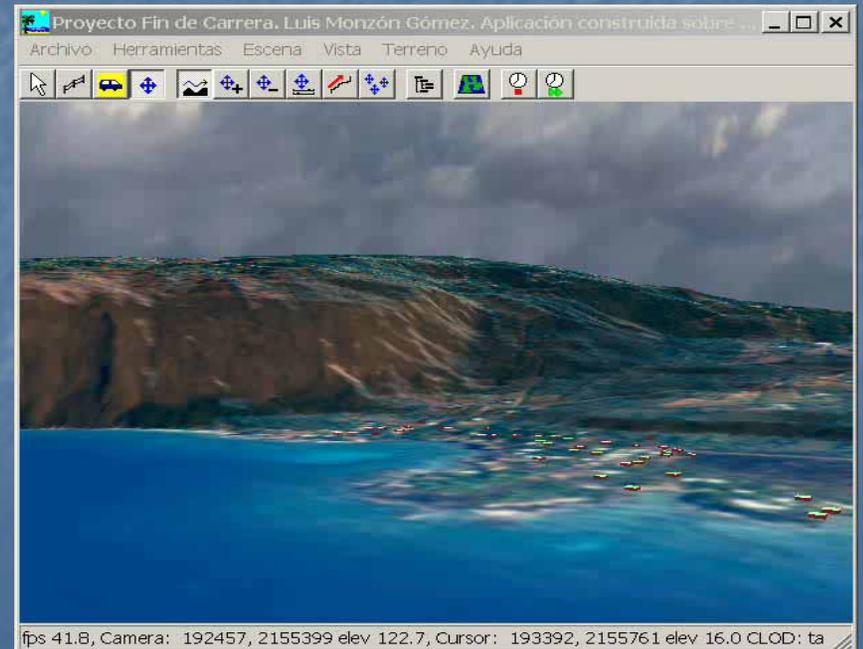
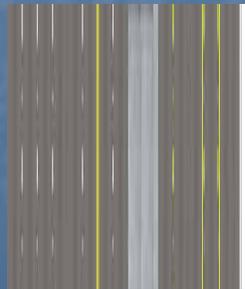
textura

cielo



Ejemplo de producto final

accesorios



# SOFTWARE VIRTUAL TERRAIN PROJECT.

## Estructura de directorios

- **VTP** (Virtual Terrain Project)
  - ▢ **Terrain Apps** (Terrain Applications)
    - ▢ **Simulador** (Programa de Simulación y Ejecutable de la aplicación).
    - ▢ **Datos** (Carpeta de datos de Terrenos).
  - ▢ **Terrain SDK** (Carpeta de estructura de librerías gráficas utilizadas)
    - ▢ **Vtdata** (librería que incluye operaciones 2D y datos geoespaciales).
    - ▢ **Vtlib** (Librería de construcción de terrenos y renderizado en 3D).
    - ▢ **Xmlhelper** (Librería para la lectura de archivos xml).
  
- **APIs** (Application Program Interface)
  - ▢ **gdal118** (Librería para la lectura de datos geoespaciales).
  - ▢ **libpng-1.2.2** (Librería para crear y manipular archivos gráficos en formato png).
  - ▢ **libjpeg-6b** (Librería para leer y escribir archivos en formato jpg).
  - ▢ **libwww** (Interface de propósito general).
  - ▢ **mini-4.7** (Librería que se ocupa de la triangularización del terreno).
  - ▢ **netcdf-3.5.0.win32bin** (interface para acceso a datos).
  - ▢ **osg-0.9.4-bin OSG** (Open Scene Graphic (Open GL)).
  - ▢ **wxWindows-2.4.0** (Librería gráfica de interface de diálogo).
  - ▢ **STLport-4.5.3** (Librería que aporta seguridad y mejora en el rendimiento).

# DIRECTORIO DE APLICACIONES

## Terrain Apps

### Simulador

- Simulador.ini;
- Carpeta wx de archivos relativos al GUI;
- Carpeta de archivos de encabezado .h y de código fuente .cpp;
- Simulador.dsw;

### Datos

- Información relacionada con el terreno.
- Se clasifican por tipos de datos en diferentes carpetas.

- **Terrains**: Contiene los archivos de iniciación .ini.
- **Elevation**: Contiene archivos .bt y .prj de elevación.
- **GeoSpecific**: Contiene archivos .jpg y .png con texturas.
- **Datos del coche**: Contiene archivos .3ds con los componentes del automóvil.
- **Building Data**: Contiene archivos .vtst de edificios.
- **Road Data** : Contiene archivos .rmf de carreteras.
- **Locations** : Contiene archivos .loc con emplazamientos de terreno predeterminados.
- **PlantModels** : Contiene archivos .png de plantas.
- **Sky** : Contiene archivos jpg y png de cielos.

# SOFTWARE VIRTUAL TERRAIN PROJECT.

## Estructura de directorios

- VTP (Virtual Terrain Project)
  - ▢ **Terrain Apps** (Terrain Applications)
    - ▢ **Simulador** (Programa de Simulación y Ejecutable de la aplicación).
    - ▢ **Datos** (Carpeta de datos de Terrenos).
  - ▢ **Terrain SDK** (Carpeta de estructura de librerías gráficas utilizadas)
    - ▢ **Vtdata** (librería que incluye operaciones 2D y datos geoespaciales).
    - ▢ **Vtlib** (Librería de construcción de terrenos y renderizado en 3D).
    - ▢ **Xmlhelper** (Librería para la lectura de archivos xml).
  
- APIs (Application Program Interface)
  - ▢ gdal118 (Librería para la lectura de datos geoespaciales).
  - ▢ libpng-1.2.2 (Librería para crear y manipular archivos gráficos en formato png).
  - ▢ libjpeg-6b (Librería para leer y escribir archivos en formato jpeg).
  - ▢ libwww (Interface de propósito general).
  - ▢ mini-4.7 (Librería que se ocupa de la triangularización del terreno).
  - ▢ netcdf-3.5.0.win32bin (interface para acceso a datos).
  - ▢ osg-0.9.4-bin OSG (Open Scene Graphic (Open GL)).
  - ▢ wxWindows-2.4.0 (Librería gráfica de interface de diálogo).
  - ▢ STLport-4.5.3 (Librería que aporta seguridad y mejora en el rendimiento).

## VTDATA

- Librería base de programación, junto a vtlib.
- Librería desarrollada en C++.
- Proporciona clases, operaciones 2D, y datos geoespaciales

## VTLIB

- Librería construida sobre:
  - la librería vtdata.
  - La librería gráfica **OSG**.
- construye y renderiza terrenos en 3D a partir de fuentes de datos geoespaciales.
- Formatos que soporta:
  - .flt (Open Flight)
  - .3ds (3D Studio)
  - .obj (Alias Wavefront)
  - .lwo/lw/geo (Light Wave)
  - .pfb (Performer)

## XMLHELPER

- Librería empleada para leer archivos XML.
- Encapsula los métodos más comunes de trabajo en xml.
- En este proyecto los archivos xml se usan para hacer listas de datos.

# SOFTWARE VIRTUAL TERRAIN PROJECT.

## Estructura de directorios

- VTP (Virtual Terrain Project)
  - ▢ **Terrain Apps** (Terrain Applications)
    - ▢ **Simulador** (Programa de Simulación y Ejecutable de la aplicación).
    - ▢ **Datos** (Carpeta de datos de Terrenos).
  - ▢ **Terrain SDK** (Carpeta de estructura de librerías gráficas utilizadas)
    - ▢ **Vtdata** (librería que incluye operaciones 2D y datos geoespaciales).
    - ▢ **Vtlib** (Librería de construcción de terrenos y renderizado en 3D).
    - ▢ **Xmlhelper** (Librería para la lectura de archivos xml).
  
- APIs (Application Program Interface)
  - ▢ **gdal118** (Librería para la lectura de datos geoespaciales).
  - ▢ **libpng-1.2.2** (Librería para crear y manipular archivos gráficos en formato png).
  - ▢ **libjpeg-6b** (Librería para leer y escribir archivos en formato jpg).
  - ▢ **libwww** (Interface de propósito general).
  - ▢ **mini-4.7** (Librería que se ocupa de la triangularización del terreno).
  - ▢ **netcdf-3.5.0.win32bin** (interface para acceso a datos).
  - ▢ **osg-0.9.4-bin OSG** (Open Scene Graphic (Open GL)).
  - ▢ **wxWindows-2.4.0** (Librería gráfica de interface de diálogo).
  - ▢ **STLport-4.5.3** (Librería que aporta seguridad y mejora en el rendimiento).

- **Gdal118:** Librería para resolver proyecciones.
- **libpng-1.2.2:** Crea y manipula archivos gráficos en formato PNG.
- **libjpeg-6b:** Lee y escribe archivos de compresión de imagen .jpg.
- **Libwww:** Sirve de text de protocolo.
- **mini-4.7:** Se ocupa de la triangularización del terreno.
- **netcdf-3.5.0.win32bin:** Soporta la creación, acceso y tratamiento de los datos.
- **STLport-4.5.3:** Proporciona seguridad, y una mejora en la utilización de memoria y en la eficiencia en el tiempo de ejecución.

# OSG-0.9.4-BIN OSG

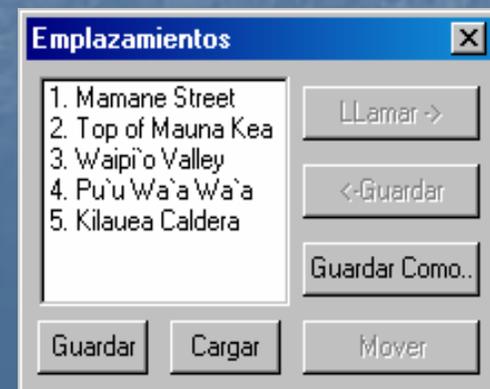
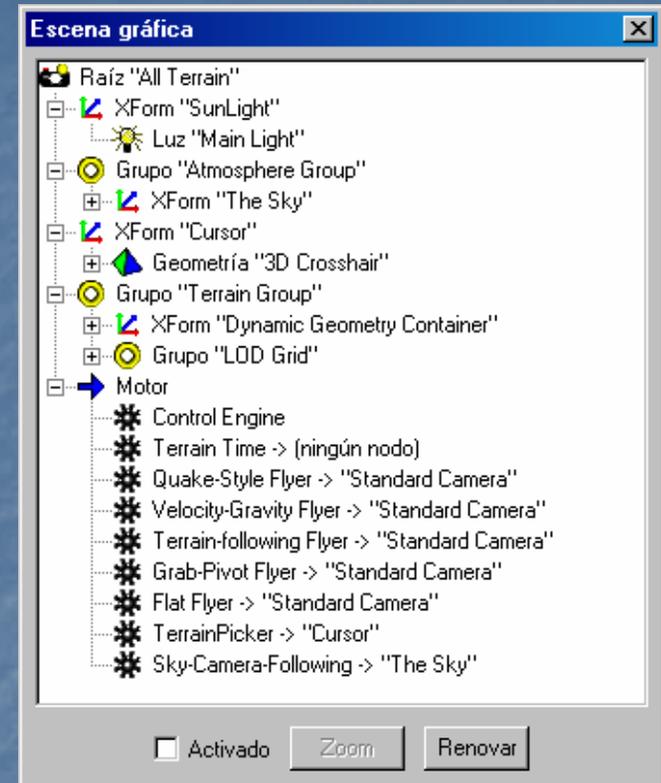
- Herramienta de volcado de imágenes en tres dimensiones en Real Time.
- Se emplea en simulación visual, juegos, realidad virtual, visualización científica y modelado.

## PROPIEDADES

- Representación; Dos subprocesos, el culling y el sorting.
- Productividad; Emplea OpenGL, controla la representación y emplea bases de datos y archivos de diferentes formatos.
- Portabilidad; Windows 95/98/ME2000/XP, Linux, Mac OSX, FreeBSD, IRIX, Solaris, HP-UX, Playstation 2 :
- Interoperabilidad; Similar a \_OpenGL. Se puede emplear con cualquier GUI que sea capaz de crear un contexto de OpenGL
- Extensibilidad; Encapsula datos de primitivas de OpenGL. Se realizan operaciones de lectura y escritura sin modificar sus aplicaciones centrales.
- Fuente; Emplea las metodologías de software más modernas, lo que le permite desarrollar solo lo que se necesita en el momento que se necesita.

# WXWINDOWS

- Generador de marcos en lenguaje C++.
- Proporciona avanzados GUI (Graphic User Interfaces).
- Plataformas que soportan wxWindows:
  - Windows 95/98/ME, Windows NT, Windows 2000, Windows XP.
  - La mayoría de las variantes Unix con GTK+, X11 y Motif/Lesstif.
  - MacOS 9.x y 10.x.
  - OS/2 (alpha).
- Soporta la mayoría de los compiladores, incluido el Visual C++ 6.0.



# APLICACIÓN DEL SIMULADOR

## ARCHIVOS DE INICIACIÓN

### ■ Simulador.ini

### Terreno.ini

DataPath	../Datos del coche	Name	Generic Terrain
InitialTerrain	Generic Terrain	Filename	PRUEBA.bt
FullScreen	0	Locations_File	
Gravity	0	Vertical_Exag	1.3
HTMLPane	0	Min_Height	10
FloatingToolBar	0	Nav_Speed	100
Sound	1		
VirtualCursor	0	LOD_Method	2
SpeedTest	0	Pixel_Error	2
QuakeNavigation	0	Tri_Count	10000
PlantSize	1	Tristrips	1
PlantShadows	0	Detail_Texture	1

# APLICACIÓN DEL SIMULADOR

## VENTANAS DE INICIACIÓN

**Ventana de Inicio de la Aplicación Enviro**

Opciones de Terreno:

Vista de la Tierra

Imagen:

Terreno

Opciones Globales:

Pantalla Completa  Velocidad y Gravedad

Barra de herramientas flotante  Sonido

HTML  Cursor Virtual

Tamaño de Exageración para Plantas y Vallas:

Sombras de Plantas

**Parámetros de Creación de Terreno**

Elevación | Textura | Ilustración | Navegación

Malla de Elevación:

Grid Filename:

Método del nivel de detalle del terreno:

Límite de píxeles de error:

Número de Triángulos:

Uso de strips/fans  Detalle de Textura

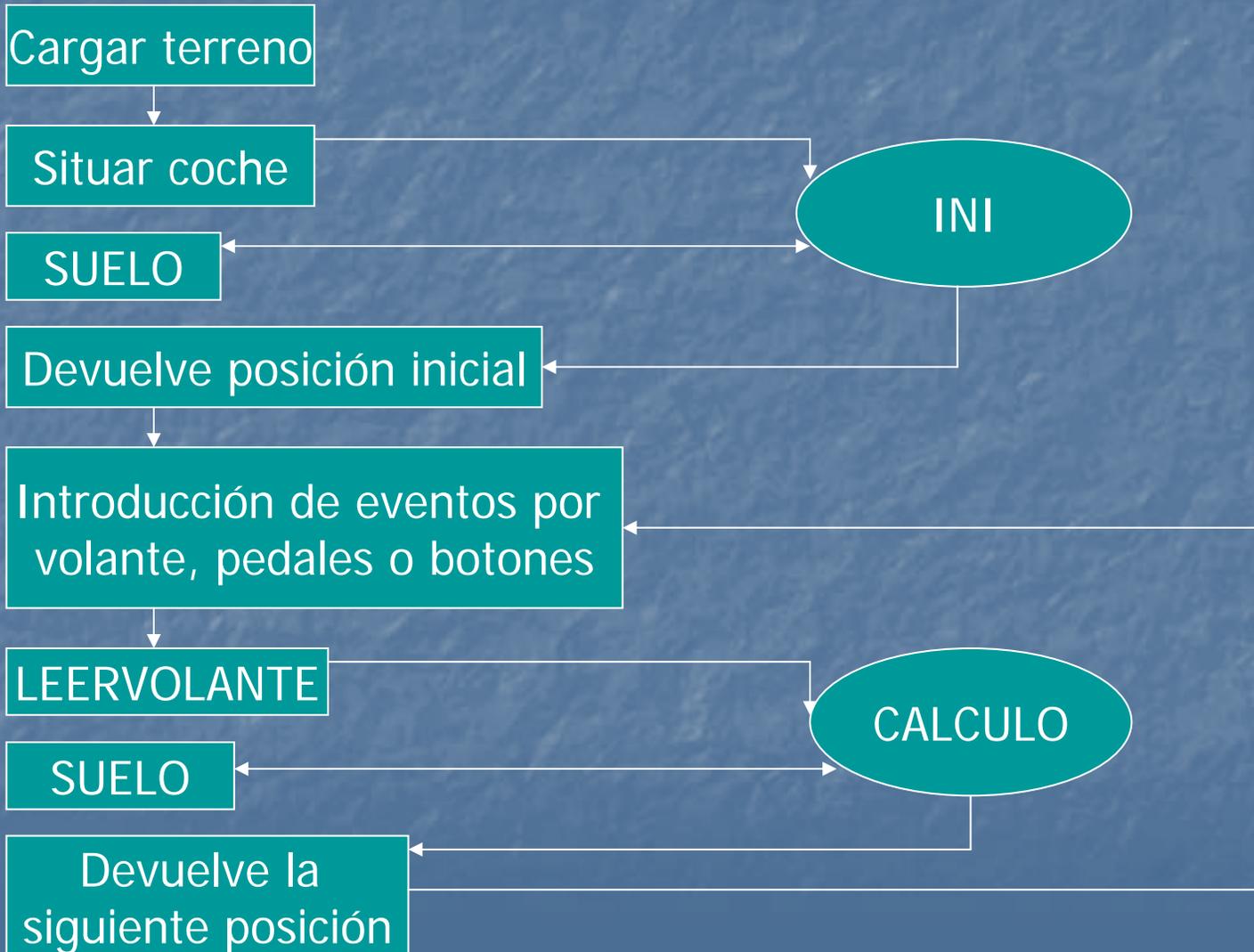
Elevación TIN:

Nombre de Archivo TIN:

Exageración vertical:  x

# APLICACIÓN DEL SIMULADOR

## DIAGRAMA DEL PROCESO DE SIMULACIÓN



# APLICACIÓN DEL SIMULADOR

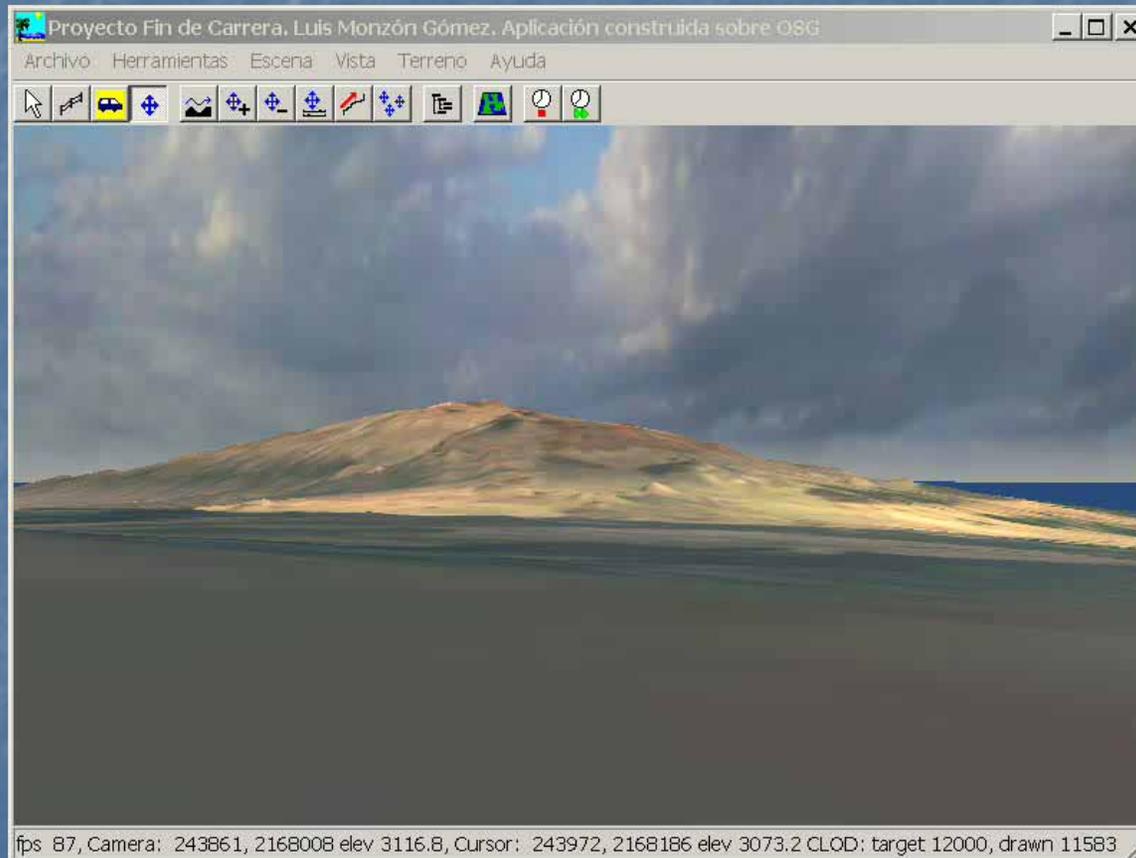
## UNIÓN DE ENTORNO DE SIMULACIÓN Y MODELO DINÁMICO

- Programación referente al entorno->C++.
- Programación del modelo dinámico->Fortran.
- Lenguaje mixto de programación:
  - Llamada desde C++.Ejemplo:
    - ```
extern "C" void __stdcall CALCULO(double *, double *, double *, int*);  
double dmatrizarray[496], mod;  
int sense;  
CALCULO(&t, dmatrizarray, &mod, &sense);
```

- Llamada desde Fortran. Ejemplo:

- ```
extern "C" void __stdcall SUELO(double *x, double *y, double *a,double n[3]){  
    FPoint3 vector;  
    vector.x = *x;  
    vector.z = *y;  
    FPoint3 carNormal;  
    vtTerrain *pTerr = GetCurrentTerrain();  
    vtHeightField3d *pHF = pTerr->GetHeightField();  
    float f=*a;  
    pHF->FindAltitudeAtPoint(vector, f, &carNormal);  
    *a=f;  
    n[ ]=carNormal;  
}
```

# INTERIOR DE LA APLICACIÓN DEL SIMULADOR



# CONCLUSIONES

- Objetivos iniciales cumplidos íntegramente:
  - Creación de terrenos 3D con un mayor realismo.
  - Independencia del terreno empleado en la aplicación.
  - Obtención de datos a partir de un terreno dado: altitud y normal.
  - Modelado de un vehículo.
- Logros adicionales:
  - Carga de distintos entornos y modelos de vehículos, sin necesidad de tener que volver a reprogramar el código y adecuarlo a los nuevos datos.
  - Situación de distintos elementos(carreteras, vegetación, edificios, ...) sobre el terreno.
  - Visión de vuelo sobre el terreno.
  - Desarrollo de un sencillo y completo interface de usuario.
  - Completo desarrollo del simulador de conducción, que en un principio se había pensado como un desarrollo posterior a este proyecto.
  - Elección por ratón de la posición inicial del automóvil
  - Control de cámara en la simulación.

# OTRAS CAPACIDADES

- Solución de estados que no resuelve el modelo dinámico.
- Optimización de la visualización.
- Creación de entornos más familiares.
- Cálculo de colisiones.
- Almacenamiento directo de imágenes en formato de película.