

# IberOgre y Sion Tower

## Memoria para el V CUSL

3 de abril de 2011

Revisión 1

David Saltares Márquez



# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Motivaciones y objetivos</b>	<b>4</b>
2.1. IberOgre . . . . .	4
2.2. Sion Tower . . . . .	4
<b>3. Enlaces</b>	<b>5</b>
<b>4. Requisitos y dependencias</b>	<b>6</b>
4.1. Instalación en GNU/Linux . . . . .	6
4.2. Instalación en Windows . . . . .	7
4.3. Compilación en GNU/Linux . . . . .	7
4.4. Compilación en Windows . . . . .	8
<b>5. Desarrollo</b>	<b>8</b>
5.1. IberOgre . . . . .	8
5.1.1. Estructura de los artículos . . . . .	8
5.1.2. Artículos publicados . . . . .	9
5.2. Sion Tower . . . . .	10
5.2.1. Documento de diseño . . . . .	10
5.2.2. Música y efectos de sonido 3D . . . . .	11
5.2.3. Detección de colisiones . . . . .	11
5.2.4. Carga de escenarios en formato DotScene . . . . .	12
5.2.5. Demo técnica . . . . .	13
<b>6. Colaboraciones y difusión</b>	<b>13</b>
6.1. Aportaciones a IberOgre . . . . .	13
6.2. Difusión del proyecto . . . . .	14
6.3. Colaboraciones en Sion Tower . . . . .	15

## 1. Introducción

En este texto se proporciona información a modo de resumen sobre los objetivos, motivaciones, desarrollo y logros del proyecto *IberOgre* y *Sion Tower*. Para detalles adicionales es recomendable acudir a las fuentes que se citan a lo largo del texto como el blog de desarrollo o la forja en Red Iris.

**IberOgre** es una wiki en español sobre desarrollo de videojuegos en 3D utilizando el motor de renderizado Ogre. Surge principalmente ante la inexistencia de documentación al respecto en castellano. No solo está dirigida a cubrir el uso del mencionado motor sino que también trata los conceptos matemáticos y físicos mínimos para desarrollar juegos tridimensionales.

*IberOgre* se aloja en los servidores de la Oficina de Software Libre de la Universidad de Cádiz (OSLUCA):

<http://osl2.uca.es/iberogre>



Figura 1: Logo del proyecto *IberOgre*

**Sion Tower** es un videojuego de estrategia y acción multiplataforma (GNU/Linux y Windows) con elementos de fantasía en 3D cuyo objetivo principal es servir de ejemplo final a *IberOgre* aunque tiene un claro componente lúdico para los usuarios de escritorio. Controlamos a un joven mago que debe defender la Torre Sagrada de una invasión mientras sus compañeros están celebrando un rito. Cada nivel corresponde a un piso de la torre y el objetivo consiste en evitar que los enemigos lleguen al centro del mismo empleando hechizos y colocando trampas.

Está desarrollado en C++ utilizando bibliotecas libres como Ogre, OIS, SDL, SDL mixer y pugixml.



Figura 2: Logo del proyecto *Sion Tower*

## 2. Motivaciones y objetivos

En esta sección se exponen las motivaciones que me llevaron a embarcarme en este proyecto así como los objetivos que me he propuesto cumplir de cara al V Concurso Universitario de Software Libre y más allá.

### 2.1. IberOgre

- Proporcionar documentación en castellano sobre desarrollo de videojuegos en 3D en un panorama de vacío al respecto.
- Especial atención y soporte a herramientas de desarrollo libres ya que la mayoría de documentación en inglés se centra en software privativo.
- Aunar los conceptos de geometría del espacio y uso de un motor 3D para que el lector no se vea obligado a acudir a decenas de fuentes diferentes.
- Ayudar al usuario a dar el salto de las 2D a las 3D. *IberOgre* podría considerarse como una extensión de *Wikijuegos*<sup>1</sup>, una wiki sobre desarrollo de videojuegos en 2D utilizando libSDL alojada también en la OSLUCA.
- Ofrecer artículos con ejemplos y ejercicios prácticos para que el usuario comprenda el funcionamiento de Ogre y amplíe sus conocimientos por sí mismo.
- Seguir buenas prácticas para desarrollar software multiplataforma.
- Crear una comunidad que aprenda y colabore activamente en la creación de nuevo contenido.

### 2.2. Sion Tower

- Deseo personal de aprender a desarrollar juegos en 3D.
- Proporcionar una aplicación lo más realista posible de los conocimientos expuestos en *IberOgre* .

---

<sup>1</sup>Wikijuegos: <http://osl.uca.es/wikijuegos>

- Ofrecer información extensa sobre el proceso de desarrollo completo de un videojuego.
- Liberar los elementos del motor (debidamente documentados) más susceptibles a ser reutilizados en otros proyectos.
- Formar un pequeño equipo multidisciplinar que colabore aportando contenido multimedia al proyecto: arte 2D/3D, música y efectos de sonido.
- Motor extensible orientado a la creación de contenidos como niveles adicionales.

### 3. Enlaces

Se adjunta la lista de enlaces en los que encontrar información adicional del proyecto *IberOgre y Sion Tower* .



Figura 3: Medios con información sobre *IberOgre y Sion Tower*

- Blog de desarrollo: noticias, anuncios, experiencias, anécdotas y documentación sobre el proceso de desarrollo.

<http://siondream.com/blog/category/proyectos/pfc>

- Forja: repositorio Subversion, noticias, sistema de gestión de tareas y ficheros publicados.

<https://forja.rediris.es/projects/cus15-iberogre>

- Web estática de la forja: información básica y enlaces a todos estos medios aprovechando el buen posicionamiento en buscadores de Red Iris.

<http://cus15-iberogre.forja.rediris.es>

- Twitter: noticias, detalles del desarrollo y contacto cercano con los seguidores del proyecto.

<http://twitter.com/iberogre>

- Canal de Youtube: vídeos sobre las distintas fases del desarrollo de *Sion Tower* .

<http://youtube.com/user/davidsaltares>

## 4. Requisitos y dependencias

Para poder compilar y ejecutar tanto los ejemplos de *IberOgre* como las pruebas (directorio branches) o versión estable (trunk) de *Sion Tower* es necesario instalar las versiones de desarrollo de Ogre y de las bibliotecas que se utilizan de forma auxiliar como OIS, SDL y SDL mixer.

Es posible encontrar detalles adicionales en los artículos correspondientes de *IberOgre* :

- Instalación de Ogre3D 1.7 en GNU/Linux
- Instalación de Ogre3D 1.7 en Windows



Figura 4: Todo el código de *IberOgre* y *Sion Tower* es multiplataforma

### 4.1. Instalación en GNU/Linux

La instalación de los paquetes supone una distribución basada en Debian como Ubuntu, en caso de utilizar otra distribución el proceso no debe diferir en exceso del que se expone a continuación.

1. Instalar las dependencias de Ogre:

```
sudo apt-get install g++ make libfreetype6-dev libboost-date-time-dev \  
libboost-thread-dev nvidia-cg-toolkit libfreeimage-dev zlib1g-dev \  
libzip-dev libois-dev libcppunit-dev doxygen libxt-dev libxaw7-dev \  
libxxf86vm-dev libxrandr-dev libglu-dev cmake
```

2. Descargar el código fuente de la última versión de Ogre desde la web oficial:

<http://www.ogre3d.org/download/source>

3. Descomprimos el paquete y accedemos desde la terminal.
4. Creamos un makefile dependiente de nuestra plataforma con cmake, compilamos e instalamos con make:

```
cmake .  
make  
sudo make install
```

5. Copiamos las bibliotecas de Ogre para no tener que distribuirlas con cada prueba:

```
sudo cp lib/* /usr/lib/
```

6. Instalamos OIS (Object Oriented Input System), SDL y SDL mixer:

```
sudo apt-get install libois-dev libsdl1.2-dev libsdl-mixer1.2-dev
```

## 4.2. Instalación en Windows

1. Descargar una versión de MinGW que contenga GCC 4.5 o superior.
2. Instalar el paquete en *C:/mingw* y añadir *C:/mingw/bin* a la variable de entorno *PATH*.
3. Instalar el editor de flujo sed, utilizado para generar las dependencias en los makefiles. Añadir su ruta a la variable de entorno *PATH*.

```
http://downloads.sourceforge.net/gnuwin32/sed-4.2-1-setup.exe
```

4. Descargar e instalar la versión para MinGW de Ogre desde la web oficial:

```
http://www.ogre3d.org/download/sdk
```

5. Establecer la variable de entorno *OGRE\_HOME* que apunte al directorio de instalación de Ogre.
6. Establecer la variable de entorno *BOOST\_ROOT* que apunte al directorio boost dentro de la instalación de Ogre.
7. Descargar e instalar SDL y SDL mixer para MinGW.

## 4.3. Compilación en GNU/Linux

Para compilar cualquier ejemplo de *IberOgre* o el juego *Sion Tower* en GNU/Linux es necesario acceder al directorio desde una terminal y ejecutar:

```
# Modo debug  
make  
  
# Modo release  
make modo=release
```

## 4.4. Compilación en Windows

Para compilar cualquier ejemplo de *IberOgre* o el juego *Sion Tower* en Windows es necesario acceder al directorio desde una terminal y ejecutar:

```
# Modo debug
make -f makefile-windows

# Modo release
make -f makefile-windows modo=release
```

## 5. Desarrollo

Haremos un pequeño recorrido a modo de resumen sobre el trabajo realizado durante el periodo del V CUSL. Me gustaría hacer hincapié en que para obtener detalles adicionales lo mejor es acudir a las fuentes referenciadas.

### 5.1. IberOgre

#### 5.1.1. Estructura de los artículos

Cada artículo de *IberOgre* cuenta con una estructura similar para que el lector se habitue a un flujo de trabajo uniforme. En primer lugar se hace una pequeña introducción sobre los contenidos a tratar en el texto. Posteriormente se pasa a una enumeración de conocimientos previos recomendados para que el lector pueda acudir a los correspondientes artículos y evitar sentirse perdido. A continuación da comienzo el grueso del artículo en el que se desgrana el tema en cuestión empleando pequeños ejemplos y diagramas. Finalmente se incluye un ejemplo de mayores dimensiones debidamente documentado que el usuario puede descargar, probar, estudiar y modificar. Finalmente se ofrece una pequeña conclusión a modo de resumen indicando los siguientes pasos a dar a partir de ese momento.

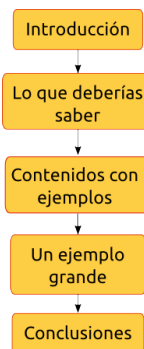


Figura 5: Flujo de lectura en *IberOgre*



A continuación se adjunta una lista de los principales artículos publicados en *IberOgre* hasta la fecha:

### 5.1.2. Artículos publicados

#### ■ Primeros pasos

- Comenzando en IberOgre: qué es IberOgre, su filosofía, estructura y objetivos.

#### ■ Programación de videojuegos en 3D: por el momento no se ha redactado ningún artículo sobre conocimientos generales de desarrollo de videojuegos en 3D.

#### ■ Ogre3D

- Conociendo Ogre3D: qué es Ogre, sus capacidades y limitaciones. Repaso sobre sus características principales.
- Conceptos generales: arquitectura y descripción de cada subsistema que compone el motor.
- Instalación de Ogre3D 1.7 en GNU/Linux.
- Instalación de Ogre3D 1.7 en Windows.
- Creación de un entorno de trabajo multiplataforma: explicación para establecer un entorno de trabajo (jerarquía de directorios, compilación...) para trabajar con proyectos que usen Ogre tanto en Windows como en GNU/Linux.
- Inicialización y cierre de Ogre: detalles sobre el proceso de inicialización, configuración y cierre de Ogre. El primer artículo en el que comenzamos a crear aplicaciones.
- Gestión de recursos: conocer la forma en la que Ogre gestiona recursos como modelos, texturas y esqueletos. Aprender a controlar el ciclo de vida de los recursos.
- Creación básica de escenas: ventanas, cámaras y carga de modelos para construir escenas sencillas.
- Manipulación de nodos: gestión de los nodos de la escena, uso de transformaciones (movimiento, escala y rotación).
- Luces, sombras y entorno: trabajar con efectos de iluminación, sombras, efectos de niebla y fondos.

#### ■ Otras tecnologías

- Manejo básico de OIS: conocimientos sobre la biblioteca de gestión de eventos de entrada que se utiliza a lo largo de la wiki.
- Exportar modelos desde Blender: proceso de instalación, configuración y uso del plugin para exportar modelos desde Blender a Ogre. Consejos para conseguir exportaciones óptimas.

- Colisiones y físicas con OGREBullet: instalación y uso de esta biblioteca de simulación física con cuerpos rígidos para Ogre.



Figura 6: Imagen de uno de los ejemplos de 'Luces, sombras y entorno'

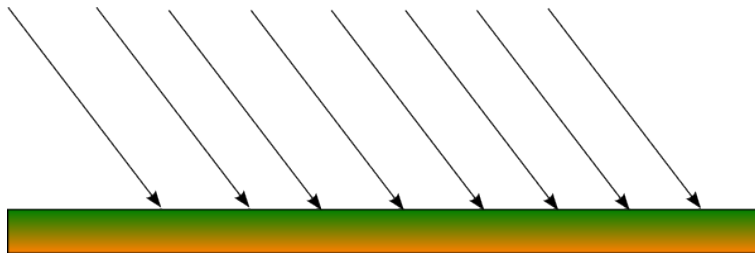


Figura 7: Diagrama explicativo de la luz direccional en Ogre

## 5.2. *Sion Tower*

### 5.2.1. Documento de diseño

Uno de los primeros pasos en el desarrollo de *Sion Tower* fue la elaboración de un documento de diseño. Este escrito recoge una descripción del juego, mecánicas de juego, menús, controles, personajes, enemigos y habilidades. Finaliza con una lista de recursos artísticos necesarios para el proyecto. Puede descargarse desde:

<http://forja.rediris.es/frs/download.php/2019/gdd.pdf>

Esto me ha permitido conseguir colaboraciones de gran importancia. Los interesados en el proyecto pueden informarse sobre el mismo, además, pueden

conocer cuáles son las tareas pendientes acudiendo a la lista de recursos. Todos los artistas han compartido la misma idea de *Sion Tower* en términos de estilo gracias a las descripciones.

### 5.2.2. Música y efectos de sonido 3D

Ogre no incorpora reproducción de música ni efectos de sonido por lo que fue necesario desarrollar un sistema propio. Cuenta con las siguientes funcionalidades:

1. Reproducción de música en formato OGG.
2. Reproducción de efectos de sonido.
3. Integración total con el sistema de gestión de recursos de Ogre.
4. Independiente del resto del motor.
5. Audio 3D: inserción de efectos en nodos de forma que el volumen de cada altavoz varía en función de la distancia y el ángulo con respecto a la fuente.

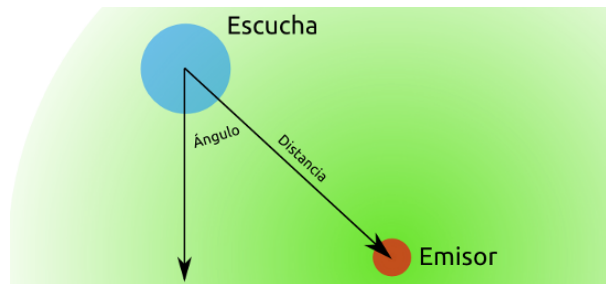


Figura 8: Esquema ilustrando el concepto del audio 3D

El código fuente está disponible junto a su documentación en:

<http://forja.rediris.es/frs/download.php/2075/siontower-3dsound-v0.1.tar.gz>

### 5.2.3. Detección de colisiones

Ogre tampoco incorpora detección de colisiones por lo que se ha implementado un sistema con las características que siguen:

1. Basado en formas: esferas, cajas (AABB y OBB) y planos.
2. Cuerpos colisionables compuestos por varias formas básicas.
3. Encapsulación de la parte física y visual de los elementos de juego.
4. Filtrado, detección y respuesta de colisiones mediante callbacks.

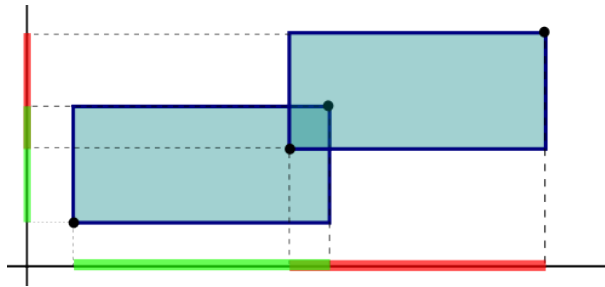


Figura 9: Test de colisión entre dos AABBs

El código fuente está disponible junto a su documentación en:

<http://forja.rediris.es/frs/download.php/2091/siontower-collisions-v0.2.tar.gz>

En el blog del proyecto se ha publicado una serie compuesta por cinco artículos documentando a fondo las generalidades y detalles del sistema de detección de colisiones.

#### 5.2.4. Carga de escenarios en formato DotScene

El motor de *Sion Tower* está orientado a la creación de contenidos. Es posible que los diseñadores o usuarios creen sus propios niveles en Blender. Colocan objetos del escenario, definen las zonas transitables por los enemigos y especifican información adicional. Los niveles se exportan a un fichero XML en formato DotScene que es cargado por el motor. Cuando *Sion Tower* avance en su desarrollo se proporcionará un manual al respecto.

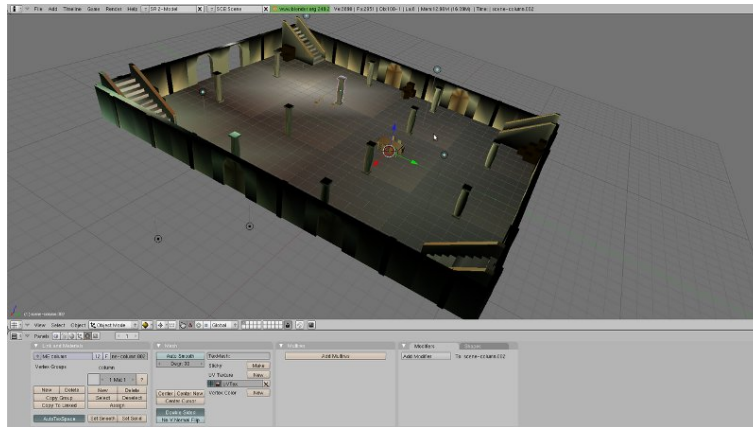


Figura 10: Creación de niveles desde Blender

### 5.2.5. Demo técnica

Con motivo de la fase local del V Concurso Universitario de Software Libre se publicó una demo técnica de *Sion Tower*. En la demo se muestra al prototipo del protagonista moviéndose por el escenario con la detección de colisiones activada y la música compuesta por los colaboradores. Puede descargarse a través de los siguientes enlaces:

<http://forja.rediris.es/frs/download.php/2151/siontower-0.1-demo-src.tar.gz>

<http://forja.rediris.es/frs/download.php/2152/siontower-0.1-demo-win.zip>



Figura 11: Sion Tower 0.1 demo técnica

La versión actual de *Sion Tower* incorpora más características como la búsqueda de caminos que solo se han mostrado en vídeos en el blog.

## 6. Colaboraciones y difusión

Uno de los mayores logros de *IberOgre* y *Sion Tower* está siendo el apoyo que recibe de la comunidad. El apoyo se manifiesta de varias formas: desde sugerencias o informes de errores hasta artículos nuevos, difusión, colaboraciones artísticas, etc. A continuación se proporciona una lista de las más relevantes.

### 6.1. Aportaciones a IberOgre

- **Artículo *Conceptos generales***: Alberto Cejas Sánchez, compañero que también participa en el V CUSL con *Fútbol es Así* redactó este texto en su totalidad.
- **Artículo *Colisiones con OgreBullet***: el compañero Alberto Cejas Sánchez también ha creado al completo un artículo sobre la detección de colisiones con la biblioteca open source OgreBullet.

- **Edición en Wikimedia:** Noelia Sales Montes y Emilio José Rodríguez Posada confeccionaron una guía liberada bajo Creative Commons 3.0 by-sa con los conocimientos básicos de edición de artículos en sistemas Wikimedia. Es accesible desde la portada de *IberOgre* lo que ayudará a conseguir colaboradores adicionales.
- **Traducción del manual oficial:** Mario Velázquez Muñoz, alumno de la Universidad Carlos III de Madrid, aportó una traducción completa del manual de referencia oficial de Ogre. La traducción formaba parte de la documentación de su Proyecto Fin de Carrera y está disponible en la portada de *IberOgre* bajo Creative Commons 3.0 by-nc-sa.
- **Otras aportaciones:** otros compañeros como Jesús González Rodríguez (*Balloon Breakers*) han aportado artículos nuevos y ediciones de los ya existentes.



Figura 12: Protagonista de Balloon Breakers, colaborador de IberOgre

## 6.2. Difusión del proyecto

- **Comunidades de desarrollo:** varias comunidades de desarrollo de videojuegos en español han publicado artículos hablando de *IberOgre* y *Sion Tower* .

<http://www.creagames.es/iberogre-un-proyecto-espanol-de-ogre-engine>

<http://razonartificial.com/2011/03/iberogre-documentacion-de-ogre-en-espanol>

<http://programandoideas.com/2011/01/iberogre-tutoriales-de-ogre3d-en-espanol>

- **Web oficial de Ogre:** *IberOgre* apareció en la portada de la web oficial de Ogre dentro de la cuarta sección de noticias.

<http://www.ogre3d.org/2011/03/01/ogre-news-4>

- **Twitter:** 65 seguidores de los que se han recibido muchas sugerencias y opiniones. Steve Streeting, fundador de *Ogre*, recomendó *IberOgre* a través de este medio.
- **Blog:** más de 60 artículos con 46.000 visitas en el periodo del concurso.



Figura 13: Protagonista de *Sion Tower* por AJR

### 6.3. Colaboraciones en *Sion Tower*

- **Banda sonora original:** el Estudio Evergreen formado por Antonio Caro Oca y Daniel Pellicer García están trabajando en una banda sonora especialmente compuesta para *Sion Tower*. Actualmente han terminado varias piezas que pueden escucharse en el blog o:

<http://estudioevergreen.wordpress.com/canciones>

- **Arte 3D:** AJR (Antonio Jiménez Rodríguez) es diseñador gráfico de forma profesional y está trabajando en el arte 3D de los personajes (diseño, modelado, texturizado y animación). Su trabajo está en el blog de desarrollo y en su blog personal:

<http://ajr-portafolio.blogspot.com/>

- **Suavizado de movimientos:** Javier Santacruz López-Cepero ha colaborado con un parche para *Sion Tower* que suaviza las rutas en la búsqueda de caminos con la malla de navegación.



Figura 14: Enemigo Goblin de *Sion Tower* por AJR.